

TESIS DOCTORAL

2016



**CODIGOALFABETIZACIÓN Y PENSAMIENTO
COMPUTACIONAL EN EDUCACIÓN PRIMARIA Y
SECUNDARIA: VALIDACIÓN DE UN
INSTRUMENTO Y EVALUACIÓN DE PROGRAMAS**

MARCOS ROMÁN GONZÁLEZ

**MÁSTER UNIVERSITARIO EN INNOVACIÓN E
INVESTIGACIÓN EN EDUCACIÓN**

PROGRAMA DE DOCTORADO EN EDUCACIÓN

Directora:

DRA. CARMEN JIMÉNEZ FERNÁNDEZ

Codirector:

DR. JUAN CARLOS PÉREZ GONZÁLEZ

TESIS DOCTORAL

2016



**CODIGOALFABETIZACIÓN Y PENSAMIENTO
COMPUTACIONAL EN EDUCACIÓN PRIMARIA Y
SECUNDARIA: VALIDACIÓN DE UN
INSTRUMENTO Y EVALUACIÓN DE PROGRAMAS**

MARCOS ROMÁN GONZÁLEZ

**MÁSTER UNIVERSITARIO EN INNOVACIÓN E
INVESTIGACIÓN EN EDUCACIÓN**

PROGRAMA DE DOCTORADO EN EDUCACIÓN

Directora:

DRA. CARMEN JIMÉNEZ FERNÁNDEZ

Codirector:

DR. JUAN CARLOS PÉREZ GONZÁLEZ

AGRADECIMIENTOS

Esta tesis doctoral ha sido posible gracias a la contribución y estímulo de múltiples personas e instituciones. A todas ellas, mi más profundo agradecimiento.

Gracias a mi directora, Dra. Carmen Jiménez Fernández, y a mi codirector, Dr. Juan Carlos Pérez González, por no haber puesto *techo* a esta tesis, y por haber tenido el arrojo de adentrarse en las nuevas líneas de investigación que pretende abrir.

Gracias a mi departamento, MIDE I, por su apoyo incondicional durante el proceso de realización de esta tesis: apoyo económico para la adquisición de las pruebas psicométricas utilizadas, y para mi estancia internacional de investigación en Reino Unido; apoyo académico a través de la confianza continua depositada en mi persona y proyectos a lo largo de estos últimos cuatro años.

Gracias a mi universidad, la UNED, y a su Facultad de Educación, que ya es la mía, por haberme formado como licenciado primero, como máster después, y como doctor finalmente. Ha sido un largo recorrido lleno de aprendizaje y satisfacción que ahora se ve colmado.

Gracias a la Universidad de Sussex (Reino Unido), y a sus profesoras Dra. Kate Howland y Dra. Judith Good, por acogerme durante el verano de 2015 y demostrarme que el código no tiene fronteras, ni entre territorios ni entre géneros.

Gracias a los colegas de la Universidad Rey Juan Carlos, Dr. Gregorio Robles y futuro Dr. Jesús Moreno, por enseñarme el valor de la investigación interdisciplinar y en equipo.

Gracias a todos los centros educativos, profesores y alumnos, que han colaborado en los estudios empíricos de esta tesis doctoral. En especial a los institutos públicos de la Comunidad Valenciana pertenecientes a la red de innovación '*IT Teaching*' por su profundo y continuado compromiso con las investigaciones que se relatan a continuación

Gracias a mis amigos, por seguir ahí, por recordarme que la cumbre de la montaña permanecería intacta, esperándonos hasta cuando yo finalizara las páginas que siguen. Por ser mis más duros y leales críticos.

Gracias a mis padres, Juan de Dios y Elvira, y a mi hermana, Elvira. Son la materia prima del esfuerzo, la constancia, la lucha perseverante, que me han permitido llegar a la meta

Y, sobre todo, gracias a mi mujer, Nuria, por ocuparse de *todo aquello* mientras yo me ocupaba de *todo esto*; en especial de lo más preciado que hay en mi vida, mis hijos Irene y Pablo.

Marcos Román González, 20 de febrero de 2016.

ÍNDICE

Pág.

INTRODUCCIÓN.....	13
-------------------	----

I. PARTE TEÓRICA	16
------------------	----

CAPÍTULO 1:

APROXIMACIÓN INICIAL AL PROBLEMA DE INVESTIGACIÓN DESDE LAS ALTAS CAPACIDADES.....	17
--	----

1.1. Criterio 1: Adecuación.....	19
----------------------------------	----

1.2. Criterio 2: Viabilidad.....	24
----------------------------------	----

1.2.1. Un ejemplo de plataforma en línea para los más capaces: Khan Academy.....	28
--	----

1.2.2. Una metodología emergente para los más capaces: la <i>'flipped classroom'</i>	37
--	----

1.3. Criterio 3: Relevancia.....	48
----------------------------------	----

1.4. Enunciación inicial de hipótesis directivas.....	56
---	----

1.5. Revisión sistemática.....	59
--------------------------------	----

1.5.1. Proceso de fijación de los parámetros de búsqueda.....	59
---	----

1.5.2. Criterios de inclusión y de exclusión.....	62
---	----

1.5.3. Procedimiento de tabulación.....	63
---	----

1.5.4. Resultados.....	64
------------------------	----

1.5.5. Discusión.....	69
-----------------------	----

1.5.5.1. ¿Altas capacidades digitales?.....	72
---	----

1.6. Conclusiones parciales.....	75
----------------------------------	----

CAPÍTULO 2:

CODIGOALFABETIZACIÓN	76
2.1. Algunas definiciones previas.....	76
2.2. Ampliación del foco de investigación.....	78
2.2.1. Hacia una inteligencia digital.....	78
2.2.1.1. Implicaciones para la generación ‘app’.....	84
2.2.2. Código y nuevos alfabetismos en la era digital: hacia la sociedad 3.0.....	86
2.2.3. El movimiento ‘code literacy’.....	89
2.2.3.1. Un evento a escala mundial: ‘The Hour of Code’.....	93
2.2.4. Delimitación final del problema de investigación.....	98
2.3. Adecuación: el <i>porqué</i> de la codigoalfabetización.....	99
2.3.1. Estudios clásicos sobre efectos del aprendizaje del ‘coding’.....	99
2.3.2. Estudios recientes sobre efectos del aprendizaje del ‘coding’.....	102
2.4. Viabilidad: el <i>cómo</i> de la codigoalfabetización.....	106
2.4.1. Taxonomía de recursos para la enseñanza-aprendizaje del ‘coding’.....	107
2.4.2. Currículos educativos para la enseñanza-aprendizaje del ‘coding’.....	116
2.4.2.1. Un ejemplo internacional: el currículo de Reino Unido.....	120
2.4.2.2. Un ejemplo nacional: el currículo de la Comunidad de Madrid.....	122
2.5. Relevancia: el <i>para qué</i> de la codigoalfabetización.....	124
2.5.1. Productos digitales derivados del ‘coding’.....	125
2.5.2. Análisis crítico del sujeto codigoalfabetizado: ‘hackers’ y ‘makers’.....	129
2.5.2.1. Breve aproximación desde la perspectiva de género.....	136
2.6. Enunciación final de hipótesis directivas.....	140
2.7. Síntesis: del ‘Learn to code’ al ‘Code to learn’.....	140

CAPÍTULO 3:

PENSAMIENTO COMPUTACIONAL	143
3.1. Concepto	144
3.1.1. Definiciones genéricas	144
3.1.2. Definiciones operativas	146
3.1.3. Definiciones psicológico-cognitivas	147
3.1.4. Definiciones educativo-curriculares	153
3.1.5. Una definición propia del dominio muestral del constructo	163
3.2. Desarrollo	164
3.2.1. Modelos de desarrollo del pensamiento computacional	164
3.2.2. Experiencias y recursos para el desarrollo del pensamiento computacional	182
3.2.2.1. Un ejemplo de integración ‘online-unplugged’: el curso ‘K-8 Intro to Computer Science’ de Code.org	192
3.3. Evaluación	205
3.3.1. Modelos de evaluación del pensamiento computacional	205
3.3.2. Instrumentos de medida del pensamiento computacional	209
3.3.2.1. El Concurso Internacional Bebras	218
3.3.2.2. Un caso particular: Dr. Scratch	222
3.4. Síntesis: hacia una consolidación del pensamiento computacional como constructo psicopedagógico	225

II. PARTE EMPÍRICA 226

CAPÍTULO 4

OBJETIVOS Y METODOLOGÍA	227
4.1. Objetivos	227

4.1.1. Generales.....	227
4.1.2. Específicos.....	227
4.2. Metodología.....	228
4.2.1. Plan de análisis de datos.....	230

CAPÍTULO 5

PRIMER ESTUDIO: LA HORA DEL CÓDIGO EN ESPAÑA.....	232
5.1. Antecedentes.....	232
5.2. Objetivos específicos.....	232
5.3. Método.....	234
5.3.1. Muestra.....	235
5.3.1.1. Muestra de centros.....	235
5.3.1.2. Muestra de profesores.....	242
5.3.1.3. Muestra de estudiantes.....	243
5.3.2. Instrumentos.....	248
5.3.2.1. Cuestionario <i>pre</i> para profesores.....	248
5.3.2.2. Cuestionario <i>post</i> para profesores.....	249
5.3.2.3. Cuestionario para estudiantes.....	249
5.3.3. Procedimiento.....	250
5.4. Resultados.....	251
5.4.1. Relativos a las características de los centros.....	251
5.4.2. Relativos al cuestionario <i>pre</i> para profesores.....	255
5.4.3. Relativos al cuestionario <i>post</i> para profesores.....	267
5.4.4. Relativos al cuestionario para estudiantes.....	271
5.5. Discusión y conclusiones parciales.....	334

CAPÍTULO 6

SEGUNDO ESTUDIO: EL TEST DE PENSAMIENTO COMPUTACIONAL (TPC)....	350
6.1. Antecedentes.....	350
6.2. Objetivos específicos.....	350
6.3. Fase I: Diseño y validación de contenido.....	352
6.3.1. Método.....	358
6.3.1.1. Muestra.....	359
6.3.1.2. Instrumentos.....	360
6.3.1.3. Procedimiento.....	361
6.3.2. Resultados.....	362
6.4. Fase II: Virtualización y validación aparente.....	374
6.5. Fase III: Puesta en servicio y aplicación.....	376
6.5.1. Método.....	376
6.5.1.1. Muestra.....	376
6.5.1.2. Instrumentos.....	380
6.5.1.3. Procedimiento.....	386
6.5.2. Resultados.....	387
6.5.2.1. Descriptivos.....	388
6.5.2.1.1. Análisis de ítems.....	397
6.5.2.2. Fiabilidad.....	401
6.5.2.2.1. Consistencia interna.....	401
6.5.2.2.2. Estabilidad.....	403
6.5.2.3. Validez.....	404
6.5.2.3.1. Validez criterial.....	404
6.5.2.3.1.1. Concurrente.....	404

6.5.2.3.1.2.	Predictiva.....	413
6.5.2.3.2.	Validez discriminante.....	420
6.5.2.3.3.	Validez convergente.....	422
6.5.2.3.3.1.	Con tareas Bebras.....	423
6.5.2.3.3.2.	Con Dr. Scratch.....	425
6.5.2.3.4.	Validez factorial.....	427
6.6.	Discusión y conclusiones parciales.....	436

CAPÍTULO 7

TERCER ESTUDIO: EVALUACIÓN DE PROGRAMAS DE CODIGOALFABETIZACIÓN Y DESARROLLO DEL PENSAMIENTO COMPUTACIONAL.....	455
7.1. Evaluación principal: el curso ‘ <i>K-8 Intro to Computer Science</i> ’ de Code.org.....	455
7.1.1. Antecedentes.....	455
7.1.2. Hipótesis específicas.....	456
7.1.3. Método.....	457
7.1.3.1. Muestra.....	458
7.1.3.1.1. Muestra de estudiantes.....	458
7.1.3.1.2. Muestra de profesores.....	459
7.1.3.2. Instrumentos.....	460
7.1.3.3. Procedimiento.....	461
7.1.4. Resultados.....	464
7.1.5. Discusión y conclusiones parciales.....	488
7.2. Evaluación complementaria: el curso ‘ <i>K-5 Computer Science Fundamentals (Course 2)</i> ’ de Code.org.....	495
7.2.1. Antecedentes.....	495

7.2.2. Hipótesis específicas.....	497
7.2.3. Método.....	497
7.2.3.1. Muestra.....	498
7.2.3.2. Instrumentos.....	498
7.2.3.3. Procedimiento.....	498
7.2.4. Resultados.....	499
7.2.5. Discusión y conclusiones parciales.....	500
7.3. Estudio de casos de alta capacidad computacional en el curso ‘ <i>Computer Programming</i> ’ de Khan Academy.....	502
7.3.1. Antecedentes.....	502
7.3.2. Objetivos específicos.....	503
7.3.3. Método.....	504
7.3.3.1. Muestra.....	504
7.3.3.2. Instrumentos.....	504
7.3.3.3. Procedimiento.....	504
7.3.4. Resultados.....	505
7.3.5. Discusión y conclusiones parciales.....	508

CAPÍTULO 8

CONCLUSIONES, IMPLICACIONES, LIMITACIONES Y SUGERENCIAS.....	511
8.1. Conclusiones.....	511
8.1.1. Con respecto al primer objetivo general.....	511
8.1.2. Con respecto al segundo objetivo general.....	512
8.1.3. Con respecto al tercer objetivo general.....	515
8.1.4. Con respecto al cuarto objetivo general.....	518

8.1.5. Con respecto al quinto objetivo general.....	519
8.1.6. Con respecto al sexto objetivo general.....	520
8.2. Respuesta al problema general de investigación.....	520
8.3. Implicaciones.....	521
8.3.1. En el marco de la directiva de <i>adecuación</i>	521
8.3.2. En el marco de la directiva de <i>viabilidad</i>	522
8.3.3. En el marco de la directiva de <i>relevancia</i>	522
8.4. Limitaciones y sugerencias.....	523
REFERENCIAS	526
<hr/>	
ÍNDICE DE FIGURAS Y TABLAS	559
<hr/>	
LISTA DE SÍMBOLOS, ABREVIATURAS Y SIGLAS	582
<hr/>	
RESUMEN / ABSTRACT	586
<hr/>	
ANEXOS	588
<hr/>	
A. CSEW: Cuestionario para profesores/as (previo).....	588
B. CSEW: Cuestionario para profesores/as (post).....	592
C. CSEW: Cuestionario para estudiantes.....	594
D. Figuras y Tablas complementarias al Capítulo 5.....	599
E. Test de Pensamiento Computacional (<i>versión 1.0</i>).....	629
F. Test de Pensamiento Computacional (<i>versión 2.0</i>).....	670

G. Instrucciones, ejemplos y autoevaluación añadidos al TPC durante su virtualización...	699
H. Tareas Bebras utilizadas para la validación convergente del TPC.....	702
I. Correo para el profesor con las instrucciones de aplicación del TPC.....	712
J. Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org.....	713
K. Cuestionario de satisfacción para estudiantes sobre el Curso K-8 de Code.org.....	716
L. Testimonios gráficos del Curso K-8 de Code.org.....	719

INTRODUCCIÓN

Motivación y finalidad del trabajo

Es evidente que vivimos inmersos en un ecosistema digital repleto de objetos programables controlados por software (Manovich, 2013). Del mismo modo que la aparición de la electricidad o del motor de combustión hizo posible el advenimiento de la sociedad industrial, la actual sociedad digital (aspirante a convertirse en una auténtica ‘sociedad del conocimiento’) descansa en el software como uno de sus principales fundamentos. Hoy cada vez más aspectos de la vida y de la cultura se hayan mediados por el software, esto es, por el conjunto de programas y aplicaciones informáticas que condicionan nuestro comportamiento e interacción con los innumerables objetos digitales que nos rodean: consultar nuestro saldo bancario, pagar impuestos, votar en las primarias de un partido político, reservar una habitación de hotel, revisar nuestro perfil en Facebook o hacer una declaración de amor por WhatsApp son sólo algunos de los innumerables ejemplos de acciones cotidianas que son realizadas software mediante.

“El software se ha convertido en nuestra interfaz con el mundo, con los demás, con nuestra memoria y nuestra imaginación: un lenguaje universal que el mundo emplea para hablar y un motor universal que propulsa el mundo. Lo que representaron la electricidad y el motor de combustión para los primeros años del siglo XX es hoy el software para los primeros años del siglo XXI” (Manovich, 2013, p. 16)

Así en nuestra sociedad digital, el ordenador¹ (ya sea como un dispositivo fijo en forma del habitual ‘ordenador de mesa-escritorio’, ya sea como uno de los emergentes dispositivos móviles: *smartphones* o *tablets*) se ha convertido en una especie de ‘metamedio’, un medio supremo a través del cual el resto de medios clásicos se vehiculan y finalmente se manifiestan: prensa, radio, televisión, etc. En ese sentido, el software, es decir, el lenguaje de los ordenadores, se convierte en un ‘metalenguaje’ en el cual se codifica el resto: el texto, el hipertexto, la imagen, el vídeo, la música, las animaciones, las aplicaciones. Toda la comunicación digital es mediada por el código informático que le da soporte.

En este contexto, ser capaz de manejar el lenguaje de las computadoras emerge como una habilidad indispensable, un nuevo alfabetismo, que nos permite participar de manera plena y efectiva en la realidad digital que nos rodea: se trata de ‘programar o ser programado’ (Rushkoff, 2010); de ser ‘app capacitado o app dependiente’ (Gardner & Davis, 2014). El término ‘codigoalfabetización’ (del inglés *‘code-literacy’*) ha sido acuñado recientemente para referirse al proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación. Así, se considera que una persona está codigoalfabetizada cuando es capaz de leer y escribir en el lenguaje de los ordenadores y otras máquinas, y de pensar computacionalmente (Román-González, 2014a). Si la codigoalfabetización alude en última instancia a una nueva práctica de lectoescritura, el pensamiento computacional (PC) se refiere al proceso cognitivo subyacente de resolución de problemas que le da soporte.

¹ A lo largo de la tesis doctoral se utilizan alternativamente como sinónimos los términos ‘ordenador’ (de mayor uso en el ámbito español) y ‘computadora’ (de mayor uso en el ámbito sudamericano).

Así las cosas, no es de extrañar que varios países estén impulsando el aprendizaje de la programación informática en sus sistemas educativos, y que instituciones como la Comisión Europea ([European Commission, 2014](#)) o la Fundación Nacional para la Ciencia de los Estados Unidos ([National Science Foundation, 2014](#)) hayan incluido las prácticas de programación (*'coding'*) como una prioridad en la formación de las nuevas generaciones de ciudadanos digitales. Si la primera brecha digital, hoy ya superada en el mundo desarrollado, se abrió entre los que tenían acceso a Internet y los que carecían del mismo; una segunda brecha digital asoma ahora entre aquéllos que se limiten a consumir objetos digitales (aplicaciones, juegos, webs, software diverso...) y aquéllos que sean capaces de expresarse, crear y compartir los suyos propios.

En este marco conceptual, la pregunta general de nuestra tesis doctoral se enuncia así: ¿es posible y deseable incorporar la codigoalfabetización en el sistema educativo español?

Organización y estructura del trabajo

Para tratar de dar respuesta a dicha pregunta, de tan amplio alcance, ésta se irá enfocando y reenfocando a través de sucesivas aproximaciones, para derivar finalmente en una serie de objetivos e hipótesis específicas que podrán ser sometidas a contraste. Con esa intención, la presente tesis doctoral se estructura en dos grandes bloques: una parte teórica y una parte empírica.

La parte teórica se subdivide a su vez en tres apartados principales: a) en primer lugar, se hace una aproximación tentativa a nuestro objeto de estudio, la codigoalfabetización, desde el ámbito de las altas capacidades y circunscrito a la programación de aplicaciones para dispositivos móviles (*'apps'*); b) tras justificar lo insuficiente del acercamiento anterior, en segundo lugar se amplía el foco de la codigoalfabetización hacia una práctica emergente de lectoescritura, un 'nuevo alfabetismo', extensible a la población escolar general y a la creación de todo tipo de objetos digitales (no sólo *'apps'*) como animaciones, juegos, webs, etc...; c) en tercer lugar, y para finalizar la parte teórica, se revisa el constructo del 'pensamiento computacional', correlato cognitivo de la codigoalfabetización, que intentaremos someter posteriormente a las leyes y exigencias de la psicometría.

Con respecto a la parte empírica, ésta se compone de tres estudios: a) en primer lugar, se desarrolla un estudio descriptivo-exploratorio sobre el evento 'La Hora del Código'² (*The Hour of Code*), promovido por la fundación Code.org³ y considerado como la mayor iniciativa a nivel mundial para la promoción y desmitificación del aprendizaje de código informático en las escuelas, y su impacto en España durante las ediciones de diciembre 2013 y diciembre 2014; b) en segundo lugar, se procede a la validación de un instrumento, el Test de Pensamiento Computacional (TPC), específicamente diseñado para población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO), pero con aplicaciones complementarias en el último ciclo de Educación Primaria (5º Primaria – 6º Primaria) y en el segundo ciclo de la Educación Secundaria Obligatoria (3º ESO y 4º ESO); c) en tercer lugar, presentamos la evaluación en profundidad del programa-curso trimestral '*K-8 Intro to Computer Science*'⁴ ([Code.org, 2015a](#)) de la fundación

² <https://hourofcode.com/es>

³ <https://code.org/>

⁴ <https://studio.code.org/s/20-hour>

INTRODUCCIÓN

Code.org implantado en aulas de primer y segundo ciclo de la ESO. Esta evaluación principal se complementa con una evaluación adicional, de menor rango y profundidad, sobre el programa-curso '*K-5 Computer Science Fundamentals (Course 2)*'⁵ (Code.org, 2015b) también de la fundación Code.org y que supone una versión simplificada del curso *K-8* para ser aplicada en el último ciclo de Primaria.

Para cerrar el círculo, y en un juego de espejos simétricos, se concluye la parte empírica aportando un breve estudio de casos de alumnos con altas capacidades computacionales de primer ciclo de la ESO que, adicionalmente al programa-curso '*K-8 Intro to Computer Science*' (Code.org, 2015a) seguido por sus compañeros de aula ordinaria, siguieron un programa-curso complementario de mayor dificultad a través de la plataforma Khan Academy: '*Computer Programming*'⁶ (Khan Academy, 2015); mediante una metodología de 'clase invertida' (*flipped classroom*) (Tourón & Santiago, 2015)

La tesis doctoral se completa con las correspondientes conclusiones, referencias y anexos.

⁵ <https://studio.code.org/s/course2>

⁶ <https://www.khanacademy.org/computing/computer-programming>

I. PARTE TEÓRICA

La recopilación de fuentes e informaciones pertinentes que nutren la parte teórica de la tesis doctoral, se ha realizado principalmente a través de los siguientes motores de búsqueda:

- El metabuscador *Linceo+*⁷ de la UNED que nos ha proporcionado acceso a las bases de datos académicas especializadas ERIC (Educación), PsycINFO (Psicología), ACM Digital Library (Informática) y Econlit (Economía)
- El buscador académico especializado *Google Scholar*⁸, a partir de un sistema de alertas vinculado a las siguientes palabras clave en lengua inglesa: ‘*code literacy*’ (codigoalfabetización), ‘*computer science education*’ (educación en ciencias de la computación), ‘*computational thinking*’ (pensamiento computacional), ‘*computational thinking assessment*’ (evaluación del pensamiento computacional), y ‘*computational thinking test*’ (test de pensamiento computacional)
- Para las referencias no académicas y ‘literatura gris’ (prensa en línea, blogosfera, mensajes en redes sociales, etc...) se han utilizado dos herramientas de curación de contenidos: un tablero digital en *Pinterest*⁹, que en el momento de escribir estas líneas acumula más de 900 *pins* y 250 seguidores; y un tablero digital en *Scoop.it*¹⁰, con más 2.600 visitas acumuladas actualmente. En la Figura 0.1 se presentan sendos códigos QR que permiten ser escaneados por el lector¹¹ para acceder a los respectivos tableros desde su dispositivo móvil con conexión a Internet.



Figura 0.1. Códigos QR que enlazan a los tableros sobre codigoalfabetización curados por el doctorando en Pinterest y Scoop.it

⁷ <http://uned.summon.serialssolutions.com/>

⁸ <http://scholar.google.es/>

⁹ <https://es.pinterest.com/marcosromangonz/codigoalfabetizacion/>

¹⁰ <http://www.scoop.it/t/codigoalfabetizacion>

¹¹ Por razones de economía lingüística, a lo largo de la tesis doctoral se utilizan los sustantivos en su forma genérica, que ha de entenderse aplicable a ambos sexos. Así, por ejemplo, ‘lector’ se refiere tanto a un lector como a una lectora.

CAPÍTULO 1

APROXIMACIÓN INICIAL AL PROBLEMA DE INVESTIGACIÓN DESDE LAS ALTAS CAPACIDADES

Comenzamos con una primera aproximación y delimitación de nuestro problema de investigación, formulada desde el ámbito de las altas capacidades y restringida al aprendizaje de la programación de aplicaciones para dispositivos móviles (*'apps'*). Este apartado ha sido ya parcialmente presentado a la comunidad científica a través de diversas publicaciones y comunicaciones (Román-González, [2013a](#), [2013b](#), [2014a](#), [2014b](#))

Nos situamos inicialmente en el marco general de la atención a la diversidad de intereses, capacidades y necesidades del alumnado, establecido inicialmente por la *Ley Orgánica 1/1990 de Ordenación del Sistema Educativo* (LOGSE, 1990), con especial acento en los principios de comprensividad e inclusión; y desarrollado posteriormente por la *Ley Orgánica 10/2002 de Calidad de la Educación* (LOCE, 2002), la *Ley Orgánica 2/2006 de Educación* (LOE, 2006), y la reciente *Ley Orgánica 8/2013 para la Mejora de la Calidad Educativa* (LOMCE, 2013) que profundizan en el principio de equidad.

Se ha ido avanzando pues desde posiciones que apuestan por el desarrollo y adaptación de un mismo programa de enseñanza que gira en torno a unos mínimos culturales para todo ciudadano, hacia posiciones que enfatizan la equidad como idea clave para garantizar un sistema educativo que ofrece a cada estudiante aquello que necesita. Recientemente se añade la idea de excelencia como referencia última de la calidad del sistema, en el sentido de que éste debe promover que cada estudiante llegue tan alto y lejos como le sea posible.

Diríamos entonces que una educación es excelente si logra de cada alumno un rendimiento satisfactorio o acorde con las capacidades y recursos que posee; y si esas adquisiciones implican para todos los intervinientes en el proceso educativo un compromiso con los valores, aspiraciones y problemas del entorno próximo y remoto (Jiménez Fernández, 2010). Dicho de otra manera, promover la excelencia equivale a facilitar los recursos necesarios que permitan a cada alumno llegar tan lejos, tan rápido, con tanta amplitud y con tanta profundidad como su capacidad y competencia le permitan. Es decir, equiparar potencial y rendimiento en *cada* alumno.

Si bien la LOGSE (1990) introdujo el concepto-término de 'Necesidades Educativas Especiales' (NEE), en su texto se suprimió la consideración explícita de los alumnos de alta capacidad; perdiendo entonces la oportunidad de incorporarse a las tendencias al alza de los países desarrollados en el sentido de apostar por la excelencia de todos y cada uno de los alumnos respondiendo de manera específica a sus necesidades (Jiménez Fernández, 1997). Este vacío vino a subsanarse posteriormente con la LOCE (2002), la LOE (2006) y la LOMCE (2013).

En cualquier caso, ya desde la LOGSE (1990) hasta el momento actual, la atención a la diversidad del alumnado en España se ha confiado fundamentalmente a dos tipos de medidas: medidas de tipo curricular (diversificación curricular, opcionalidad u optatividad, repetición de curso, y aceleración

de curso), y medidas de tipo organizativo. En la práctica, viene siendo la aceleración de curso (o flexibilización), la medida más comúnmente adoptada para el caso de los alumnos y alumnas de alta capacidad (R.D. 943/2003; o más recientemente, y en el ámbito de la Comunidad de Madrid, ORDEN 1493/2015). La aceleración es frecuentemente aplicada de manera reduccionista como ‘salto de curso’ puro y duro, sin ninguna modificación curricular asociada; y así entendida no es la forma ideal de atender las necesidades educativas específicas de estos alumnos (Jiménez Fernández, 2010). La aceleración es un modelo económico y sencillo de implantar para la administración educativa, pero insuficiente: es de dudosa aplicación en estudiantes con disincronía entre su superior desarrollo intelectual y su nivel de madurez social y afectiva, y es un modelo desestimado en ocasiones diferencialmente por las chicas de alta capacidad frente a sus homólogos masculinos, pues ellas parecen preferir permanecer con sus compañeros, evitar la competitividad, y pueden ser más proclives a aburrirse en la clase ordinaria con tal de mantener intacto su círculo social (Crombie, Bouffard-Bouchard, & Schneider, 1992); tendencia que va declinando.

Con respecto a la diversificación curricular como medida de atención para los más capaces, en la realidad de nuestras aulas se ha traducido casi exclusivamente en ‘adaptaciones curriculares no significativas’, es decir, en adaptaciones que no modifican los objetivos nucleares del currículo que deben ser alcanzados por el estudiante a final de curso; cristalizando en programas de enriquecimiento curricular parcial, sólo focalizados en la mera ampliación de contenidos (Jiménez Fernández & García Perales, 2013).

Así pues, no se han venido contemplando ‘adaptaciones curriculares significativas’ para alumnos de alta capacidad. Sin embargo, para estos sujetos:

“(...) las adaptaciones curriculares significativas habrían tenido pleno sentido pero no para rebajar los mínimos establecidos, sino para superarlos generosamente. En términos práctico vendrían a significar la triple modificación del currículo o en su contenido, proceso y producto” (Jiménez Fernández, 2010, p. 198).

Por tanto creemos justificada la necesidad de buscar, encontrar y diseñar nuevas propuestas de enriquecimiento curricular completo y significativo, que impliquen de manera integrada la triple dimensión apuntada: contenido, proceso y producto; defendida tradicionalmente por diversos autores (Parke, 1989; Pendarvis, Howley & Howley, 1990; Renzulli & Reis, 1991). En este sentido, destacan las formulaciones clásicas del ‘Modelo de Enriquecimiento Triádico’¹² (*‘Enrichment Triad Model’*) (Reis & Renzulli, 2003; Renzulli, 1977, 1988); o más recientemente del ‘Modelo de Currículo Integrado’ (*‘ICM: Integrated Curriculum Model’*) definido por VanTassel-Baska (2015), que *“ha demostrado ser una ayuda a la hora de realizar el proceso de diferenciación (...) integrando las dimensiones de contenido, proceso y producto para hacerlas más equilibradas en el proceso de aprendizaje que experimentan los alumnos (de alta capacidad)” (VanTassel-Baska, 2015, p. 233)*

Por otro lado, en la sociedad actual la programación informática se está constituyendo como una competencia básica que permite participar eficazmente en un mundo repleto de objetos digitales

¹² Apoyado a su vez en una concepción de la alta capacidad como intersección de 3 conjuntos de rasgos o ‘Modelo de los Tres Anillos’ (Renzulli, 1978, 2005): a) Capacidad por encima de la media (vinculada a la dimensión *contenido*); b) Compromiso con la tarea (vinculada a la dimensión *proceso*); y c) Creatividad (vinculada a la dimensión *producto*).

(Kafai & Burke, 2014; Resnick, 2013a). Estar codigoalfabetizado (*'code-literate'*), esto es, ser capaz de leer y escribir con los lenguajes de programación y pensar computacionalmente, emerge como un requerimiento de las sociedades avanzadas del siglo XXI (Dans, 2013; Prensky, 2008; Rushkoff, 2010, 2012).

En este contexto, nos planteamos la siguiente pregunta de investigación, primera tentativa de esta tesis doctoral: ¿es posible y deseable aprender a programar aplicaciones (*'apps'*) para dispositivos móviles (*smartphones* y *tablets*) como enriquecimiento curricular indicado especialmente para alumnos y alumnas de alta capacidad? Una delimitación inicial del problema se ilustra en la Figura 1.1, en la que quedan representadas nuestras tres hipótesis generales o directivas: **adecuación**, **viabilidad**, y **relevancia**. A continuación, exploraremos y trataremos de fundamentar dichas hipótesis.

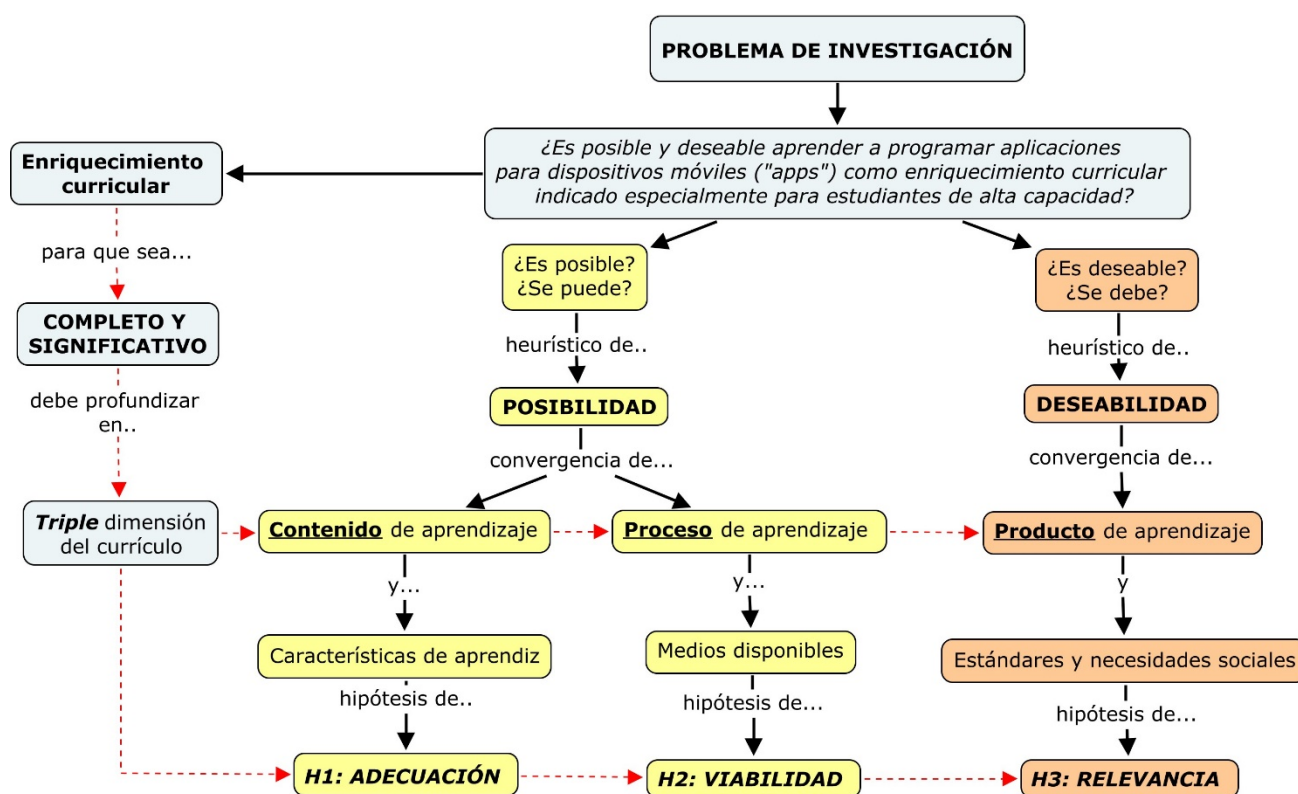


Figura 1.1. Delimitación inicial del problema de investigación.

1.1. Criterio 1: Adecuación

En informática, una aplicación es un “*programa preparado para una utilización específica*” (RAE, 2015), es decir, un tipo de programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de trabajos. En los últimos años, con la aparición y generalización masiva de dispositivos móviles con acceso a internet como *smartphones* y *tablets*, se ha multiplicado el diseño, la descarga y utilización de aplicaciones específicas para dichos dispositivos, que permiten ser disfrutadas y aprovechadas por el usuario de manera portable e itinerante.

La programación de ‘apps’ es una tarea que exige, esencialmente, manejar de manera sistemática, sostenida y orientada un lenguaje de programación (p.ej.: Java, JavaScript, HTML5, C#, Ruby, Python, etc.), compuesto por diversos comandos y parámetros (Figuras 1.2 y 1.3); en orden a crear un aplicación funcional e integrada (Román-González, 2014c). En este sentido, se adecua al estilo cognitivo de los sujetos de alta capacidad, caracterizado por la habilidad para manejar abstracciones y sistemas de símbolos a niveles altos de formalización; de manera que una inferencia curricular habitual señalada para los más capaces haya sido la introducción temprana en su aprendizaje de lenguajes simbólicos y formales¹³, como la música, las matemáticas, las lenguas extranjeras o la informática (VanTassel-Baska, 1989).



Figura 1.2. Consola de programación en lenguaje JavaScript & ProcessingJS (a la izquierda), e interfaz de la aplicación resultante (a la derecha)¹⁴. Tomado del curso ‘Computer Programming’ (Khan Academy, 2015)

<p>Coloring</p> <p><code>background(red, green, blue)</code> Set the background color</p> <p><code>color(red, green, blue)</code> Store all three color components in one variable</p> <p><code>fill(red, green, blue)</code> Set the fill color for shapes</p>	<p>Math</p> <p><code>abs(num)</code> Take the absolute value of a number</p> <p><code>cos(deg)</code> Take the cosine of an angle</p> <p><code>log(num)</code> Take the logarithm of a number</p> <p><code>pow(num, exponent)</code> Take the power of a number</p>	<p><code>bezier(x1, y1, cx1, cy1, cx2, cy2, x2, y2)</code> Draw a bezier curve</p> <p><code>ellipse(x, y, width, height)</code> Draw an ellipse</p> <p><code>line(x1, y1, x2, y2)</code> Draw a line</p> <p><code>point(x, y)</code> Draw a point</p>	<p>Programming Basics</p> <p><code>debug(arg1, arg2, ...)</code> Print to your browser's developer console</p> <p><code>for (var i = 0; i < 8; i += 1) { }</code> Repeat code a fixed number of times</p> <p><code>if (x < 20) { ... }</code> Only run code if a certain condition is met</p>
--	--	---	--

Figura 1.3. Comandos y parámetros para la programación en lenguaje JavaScript & ProcessingJS¹⁵. Tomado del curso ‘Computer Programming’ (Khan Academy, 2015)

¹³ Se entiende que un ‘lenguaje formal’ es aquel cuyos símbolos primitivos y reglas para unir esos símbolos están formalmente especificados. Ejemplos de lenguajes formales son las matemáticas, la lógica, la música o los lenguajes informáticos de programación.

¹⁴ <https://www.khanacademy.org/computer-programming/yoshi/1034542702>

Los sujetos de alta capacidad se caracterizan también por altos niveles de comprensión y de generalización. Son capaces de ver rápidamente relaciones más allá de los hechos observados, es decir, poseen gran capacidad de inferencia derivada de un aprendizaje a nivel más profundo que el propio de su edad (Jiménez Fernández, 2010). Ello puede suponer que, a partir de un conjunto finito de comandos y parámetros de programación, sean capaces de generar gran cantidad de soluciones. En este sentido, se han señalado como indicadores de la conducta de estos sujetos su alta capacidad para solucionar problemas (son un reto) y para producir ideas, objetos, soluciones nuevas (Whitmore, 1988).

Desde otro punto de vista, las tareas de programación exigen establecer relaciones formales, analogías, y procesos de inducción y memoria; en la línea del llamado pensamiento fluido (Cattell, 1963) en el cual destacan los sujetos de alta capacidad independientemente del ambiente cultural de procedencia. Por ello, la programación de ‘apps’ puede suponer un aprendizaje adecuado para sujetos de alta capacidad provenientes de todo tipo de culturas y niveles sociales. En esta misma línea, la programación, al igual que el manejo sistemático de otros lenguajes formales como la matemática, es un campo sin ‘efecto techo’ en el sentido de que el sujeto puede ir adquiriendo casi ‘ad infinitum’ niveles sucesivos de dominio, cada uno de los cuales apoyados en el anterior; promoviendo así dinámicas de desarrollo y entrenamiento sistemático del talento, tal y como se defiende actualmente en el ámbito de la alta capacidad (Gagné, 1991, 2004, 2015). Parece pues que la programación de ‘apps’ es un aprendizaje sin ‘efecto techo’, y, por lo tanto, adecuado para mentes hambrientas (*‘hungry minds’*) y curiosas intelectualmente (Von Stumm, Hell, & Chamorro-Premuzic, 2011); como las de los sujetos de alta capacidad.

Igualmente, Webb (1993) ha señalado como características propia de los niños con alta capacidad, entre otras, su intensa concentración, su atención amplia y sostenida en áreas de interés, y la persistencia en su comportamiento dirigido hacia objetivos; lo que dibuja un perfil motivacional propicio para tareas de programación.

Además, la programación de ‘apps’ no es mera recreación en el manejo de símbolos que se agotan en sí mismos, sino que aspira a crear un producto original, novedoso y útil. En este sentido es una tarea que exige creatividad; rasgo que ha venido incorporándose progresivamente en las definiciones de alta capacidad (Guilford, 1967; Renzulli, 1978; Renzulli & Gaesser, 2015). Igualmente, la programación de ‘apps’ es una tarea acorde con modelos como el de Torrance (1986) sobre pedagogía de la creatividad, dado que demanda poner en juego habilidades en las cinco categorías de pensamiento creativo: *pensamiento fluido* (o generar gran cantidad de soluciones); *flexible* (o generar soluciones diferentes a partir de los mismos principios); *original* (o generar soluciones novedosas y de síntesis); *elaborativo* (o embellecer las soluciones obtenidas); y *evaluativo* (o escoger y decidir entre las soluciones en función de las apreciaciones propias y ajenas). Todos estos principios, proyectados sobre nuestro objeto de estudio, y sintetizados en uno solo, “*generar gran cantidad de soluciones diferentes y novedosas a partir de los mismos principios de programación, en sucesivas versiones más embellecidas y perfectas, y evaluadas por una comunidad*” se producen en las actuales plataformas de aprendizaje en línea de la programación informática, como Khan Academy (2015) que analizaremos en profundidad algo más adelante; cuando a partir de un objeto de

¹⁵ Acceso a documentación completa en <https://www.khanacademy.org/computing/computer-programming/docs/pjs>

programación original se van generando sucesivos derivados o *'spin-offs'* por los distintos miembros de la plataforma (Figura 1.4)

Frog Face
9 Votes ▲ ▼ · Flag
Share...
New Program

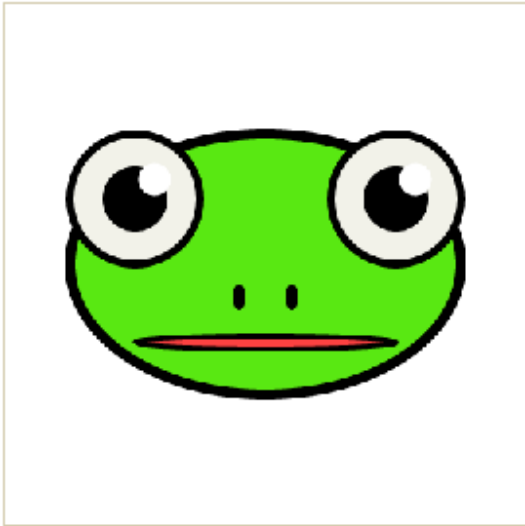
```

// Frog Face

// face
strokeWeight(8);
fill(89, 232, 18);
ellipse(200, 200, 300, 200);

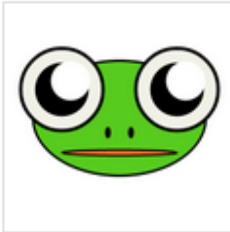
// eyes
strokeWeight(6);
fill(242, 242, 233);
ellipse(100, 150, 100, 100);
ellipse(300, 150, 100, 100);

// mouth
strokeWeight(4);
fill(252, 65, 65);
ellipse(200, 260, 200, 10);
                    
```



Joe Miller
(Updated 7 months ago)
Based on:
[Intro to Variables](#)
Save as a spin-off
Restart


Questions
Tips & Feedback
Spin-Offs
Documentation
Top Recent



Frog Face

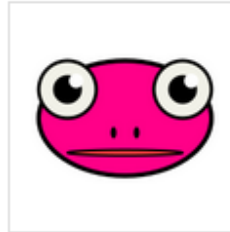
Jorge Bushby

1 Vote · 2 Spin-offs



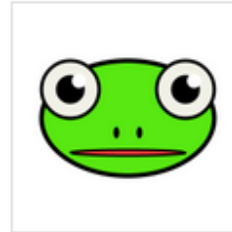
Frog Face

1 Vote · 1 Spin-off



Frog Face

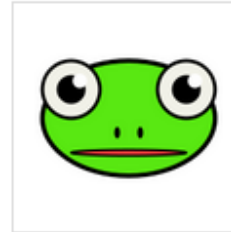
1 Vote



Frog Face

SABRINALAWHORN19


1 Vote



Frog Face


rodana2002

1 Vote



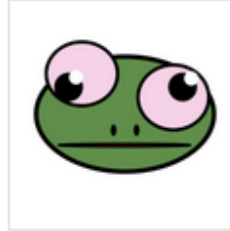
Frog Face

melissa bernabe-cruz



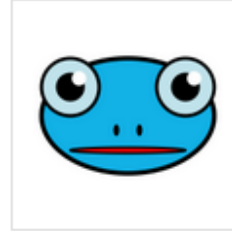
duck face

Melissa Chisholm




Spin-off of "Frog Face"

armand.likrama



Frog

Light Runner



Spin-off of "Frog Face"

AlanaFitzGerald97

Figura 1.4. Sucesivos derivados o versiones de un objeto original de programación¹⁶. Tomado del curso *'Computer Programming'* (Khan Academy, 2015)

Hay otras características de los sujetos de alta capacidad relacionadas con la creatividad, como su carácter juguetón, su desafío o reto ante lo convencional, su independencia de pensamiento, o su impulso natural a explorar ideas (Jiménez Fernández, 2010) que les convierten en excelentes

¹⁶ <https://www.khanacademy.org/computer-programming/frog-face/1292194550>

candidatos para aprender y experimentar con la programación de ‘apps’. Y muy especialmente la habilidad de estos sujetos para pensar en las cosas holísticamente, como un todo, para pasar después a comprender sus partes, que les facilita la generación de ‘apps’, que deben funcionar como un todo funcional e integrado.

Un último apunte desde la perspectiva de la creatividad: tal y como nos recuerda Amabile y Pillemer (2012, p. 7) “*el estado del sujeto intrínsecamente motivado es conducente a la creatividad, mientras que el extrínsecamente motivado va en detrimento*”. Es decir, si esperamos que un alumno de alta capacidad en última instancia genere ‘apps’ de manera creativa, no bastará con que aprenda y domine los principios de la programación, sino también que se encuentre motivado y comprometido internamente con la tarea. Por tanto, parece que debe ser un aprendizaje libremente escogido por estos estudiantes.

Además, la programación de ‘apps’ exige altas dosis de metacognición al implicar un continuo control y evaluación de múltiples procesos cognitivos de orden inferior que se van desplegando ‘en paralelo’: la programación demanda conjugar simultáneamente las tareas analíticas de escritura de código con la perspectiva sintética del producto final generado, la aplicación, que debe funcionar como un todo integrado. En este sentido, se ha señalado la superioridad de los sujetos de alta capacidad específicamente en características metacognitivas como los procesos de autorregulación consciente: prever el error y tratar de eliminarlo antes de que se produzca porque se trabajan mentalmente los conceptos (Cheng, 1993); y en los llamados metacomponentes (Sternberg, 1991), procesos ejecutivos de orden superior usados en la planificación, guía y toma de decisiones en la ejecución de las tareas.

Así pues, y siguiendo la ruta argumental que nos hemos fijado (Figura 1.1), parece que existe convergencia entre las características del **contenido** de aprendizaje que nos proponemos (‘programación de apps’) y las características del aprendiz destinatario, en este caso los alumnos de alta capacidad. Encontramos pues apoyo fundamentado para enunciar la hipótesis directiva de **adecuación**. Como síntesis final de este apartado, en la siguiente Tabla 1.1 tratamos de ilustrar como el aprendizaje de la programación de ‘apps’ daría respuesta a algunas de los principios curriculares propuestos tradicionalmente para los alumnos de alta capacidad (VanTassel-Baska, 1989).

Tabla 1.1. Convergencia entre principios curriculares para alumnos de AACC y el aprendizaje de programación de ‘apps’

<i>Principio curricular</i>	<i>La programación de apps...</i>
<i>Introducción de nuevos sistemas de símbolos en edades tempranas (ordenadores, lenguas extranjeras, estadística, música)...</i>	<i>...es un lenguaje formal y simbólico que combina comandos y parámetros en altos niveles de abstracción.</i>
<i>Currículo organizado de acuerdo a la estructura subyacente del contenido. Eliminación de la repetición.</i>	<i>...es un campo perfectamente organizado, jerarquizado y ‘sin efecto techo’, en el que se alcanzan sucesivos niveles de dominio al propio ritmo.</i>
<i>Oportunidades curriculares interdisciplinares. Uso de recursos y materiales de texto múltiples.</i>	<i>...puede proyectarse sobre cualquier ámbito de conocimiento: se pueden programar ‘apps’ sobre matemáticas, astronomía, informativas, lúdicas, etc. Es, en ese sentido, interdisciplinar. Además, los lenguajes de programación se vehiculan en lengua inglesa, con el aprendizaje transversal de idiomas que ello conlleva.</i>

1.2. Criterio 2: Viabilidad

Se ha señalado que, en general, el estilo de aprendizaje de los alumnos de alta capacidad puede caracterizarse de autónomo, centrado en la tarea, crítico, motivado, persistente y creativo (Jiménez Fernández, 2010); beneficiándose notablemente con las observaciones y correcciones del profesor o experto a su trabajo.

En este sentido, Benbow (1991) señala como metodologías especialmente adecuadas para el aprendizaje de alumnos de alta capacidad: las que contemplan el progreso continuo e individualizado (a través de currículos bien definidos y estructurados, o modulares, en el que el ritmo de avance es marcado por el propio alumno); los cursos a distancia, dado la gran capacidad de estos alumnos para el aprendizaje autónomo y autorregulado; y el uso de mentores, especialistas o expertos en campos determinados que se prestan a asesorar al alumno durante un tiempo fijado.

Por otro lado, Tourón (2008) ha señalado las posibilidades y ventajas de las metodologías de enseñanza-aprendizaje a distancia con soporte de las tecnologías de la información y la comunicación (TIC), o *'e-learning'*, para atender las necesidades educativas específicas de los alumnos de alta capacidad. Así, señala como el *'e-learning'* permite, por ejemplo: disponer de un tutor a distancia con el cual desarrollar planes de aprendizaje altamente individualizados; interactuar con otros alumnos de otros centros o escuelas, creando comunidades virtuales de aprendizaje; individualizar el ritmo y el contenido de aprendizaje, etc.

También Jiménez Fernández (2010) señala que los métodos más apropiados para los alumnos de alta capacidad son los que hacen hincapié en el trabajo autónomo, en el dominio de habilidades para aprender a pensar, en el empleo de técnicas de interrogación de alto nivel cognitivo y de los niveles superiores de pensamiento como aplicación, proyección y evaluación de la información, en la experimentación, en la expresión convergente y divergente del contenido, y en la resolución creativa de problemas. También son apropiadas actividades y situaciones para la interacción en grupo en entornos reales y en simulaciones, con reglas que den lugar a la heteroevaluación y a la autocrítica.

Por tanto, parece que, globalmente, la metodología a distancia con soporte tecnológico, con contenidos bien definidos y modulares que permitan el aprendizaje autónomo del alumno, y con sistema de apoyo tutorial o mentores, e interacción en grupo con iguales en capacidad; sería la más viable, desde un punto de vista didáctico, para la enseñanza-aprendizaje de la programación de *'apps'* en alumnos de alta capacidad. A esta viabilidad didáctica, se le suma la viabilidad económica propia de estos sistemas metodológicos a distancia y tecnológicos, que funcionan como *'economías de escala'*¹⁷: es decir, con un número finito de recursos didácticos (videos, textos, materiales, orientaciones, etc.) pueden transmitir conocimiento a un número potencialmente infinito de estudiantes; por tanto, con un reducido coste por alumno.

¹⁷ En microeconomía, se entiende por *'economía de escala'* las ventajas en términos de costos que una empresa obtiene gracias a la expansión. Existen factores que hacen que el coste medio de un producto por unidad caiga a medida que la escala de la producción aumenta. El concepto de *'economías de escala'* sirve para el largo plazo y hace referencia a las reducciones en el coste unitario a medida que el tamaño de una instalación y los niveles de utilización de *inputs* aumentan [Fuente: https://es.wikipedia.org/wiki/Econom%C3%ADa_de_escala]

La realidad reafirma dicha viabilidad. Actualmente, comenzamos a encontrar ejemplos de cursos a distancia y en línea cuyo objetivo es que el alumno aprenda a programar y desarrollar aplicaciones para dispositivos móviles, y que siguen los principios metodológicos señalados. Detectamos en este punto dos tipos básicos:

- a. Cursos con matrícula y coste económico para el estudiante, por ejemplo, “*Desarrollo avanzado de aplicaciones en iOS7 (Héroe de iOS)*”¹⁸ (FUNED, 2014) o “*Desarrollo Rápido de Aplicaciones Android en el Aula de ESO, Bachillerato y FP*”¹⁹ (FUNED, 2015), ambos promovidos desde la Fundación UNED. El coste oscila entre 250€ y 600€, incluyen documentación y videos en alta definición, se desarrollan 100% ‘on-line’, con una dedicación estimada de 5 horas semanales durante 12 semanas, certificados por la Fundación UNED por un total de 60 horas, e incluyen atención al alumno mediante foros y tutorías semanales.
- b. Cursos MOOC (‘*Massive Online Open Courses*’), masivos, en línea, abiertos y gratuitos; bien orientados al desarrollo de ‘apps’ en el ámbito empresarial como “*Emprendimiento y App Inventor*”²⁰ (MiriadaX, 2015), bien en el ámbito educativo como “*MIT – App Inventor*”²¹ (Massachusetts Institute of Technology, 2015), “*Touch Develop*”²² (Microsoft Research, 2015), o “*Khan Academy – Computer Programming*”²³ (Khan Academy, 2015); de éste último, que ya ha sido mencionado algo más arriba, hablaremos en mayor profundidad en el siguiente epígrafe.

La descripción del MOOC “*Emprendimiento y App Inventor*” (MiriadaX, 2015) es muy ilustrativa del espíritu (y metodología) que inspira estos cursos:

“En tiempos de crisis global y transformaciones en el sistema tradicional de aprendizaje y trabajo, las respuestas a los interrogantes surgen de manera clara desde el emprendimiento: la posibilidad de embarcarse en proyectos nuevos e innovadores, aprovechando los avances en las tecnologías móviles y la alta tasa de penetración móvil en la sociedad, atraen oportunidades de negocio imposibles de ignorar (...)

(...) App Inventor es una plataforma que pone al alcance de todos la posibilidad de crear aplicaciones para móviles, sin necesidad de tener conocimientos de programación y utilizando una metodología visual que facilita el desarrollo de la aplicación (...)

(...) En este curso cualquier persona, incluso sin conocimientos técnicos de programación previos, interesada en el diseño de nuevas aplicaciones para dispositivos móviles, podrá crear desde cualquier parte del mundo un nuevo negocio o multiplicar el ya existente con las enormes posibilidades que ofrece el campo de las aplicaciones móviles. La idea es ir un paso

¹⁸ <http://www.fundacion.uned.es/actividad/idactividad/5766>

¹⁹ https://formacionpermanente.uned.es/tp_actividad/idactividad/8018

²⁰ <https://miriadax.net/web/emprendimiento-y-app-inventor1>

²¹ <http://appinventor.mit.edu/explore/>

²² <https://www.touchdevelop.com/>

²³ Este curso no permite en sentido estricto la programación de ‘apps’ que luego se descargan desde tiendas de aplicaciones como ‘App Store’ (iOS) o ‘Play Store’ (Android) al dispositivo móvil del usuario; sino la programación de objetos que pueden ser luego ejecutadas a través de un navegador web en un dispositivo móvil o ‘web apps’. Esta distinción se retomará más adelante cuando tratemos de justificar que la codigoalfabetización debe extender su foco a la programación de todo tipo de objetos digitales y no sólo a las ‘apps’ en sentido habitual.

más allá, sumar sinergias, intercambiar experiencias, y lograr que las ideas innovadoras se canalicen, se enriquezcan, se lancen y viajen (...)

Los participantes del curso obtendrán los conocimientos necesarios en el campo del diseño de aplicaciones para dispositivos móviles del sistema operativo Android y las maneras de monetizar los proyectos resultantes de esas propuestas” (MiriadaX, 2015, en línea)

Revisando la ‘Guía de Aprendizaje’²⁴ del dicho curso MOOC “*Emprendimiento y App Inventor*” (MiriadaX, 2015) encontramos una serie de elementos que reflejan su viabilidad didáctica para alumnos de alta capacidad:

- Es un curso destinado a cualquier persona, desde cualquier parte del mundo, interesada en el diseño de nuevas aplicaciones (‘apps’) para dispositivos móviles.
- Se basa en el uso de ‘App Inventor’, una plataforma-consola de programación desarrollada por el MIT que pone al alcance del alumno la posibilidad de crear aplicaciones para móviles, sin necesidad de tener conocimientos previos técnicos de programación, y utilizando una metodología visual que facilita el desarrollo de la ‘app’ (Figura 1.5)

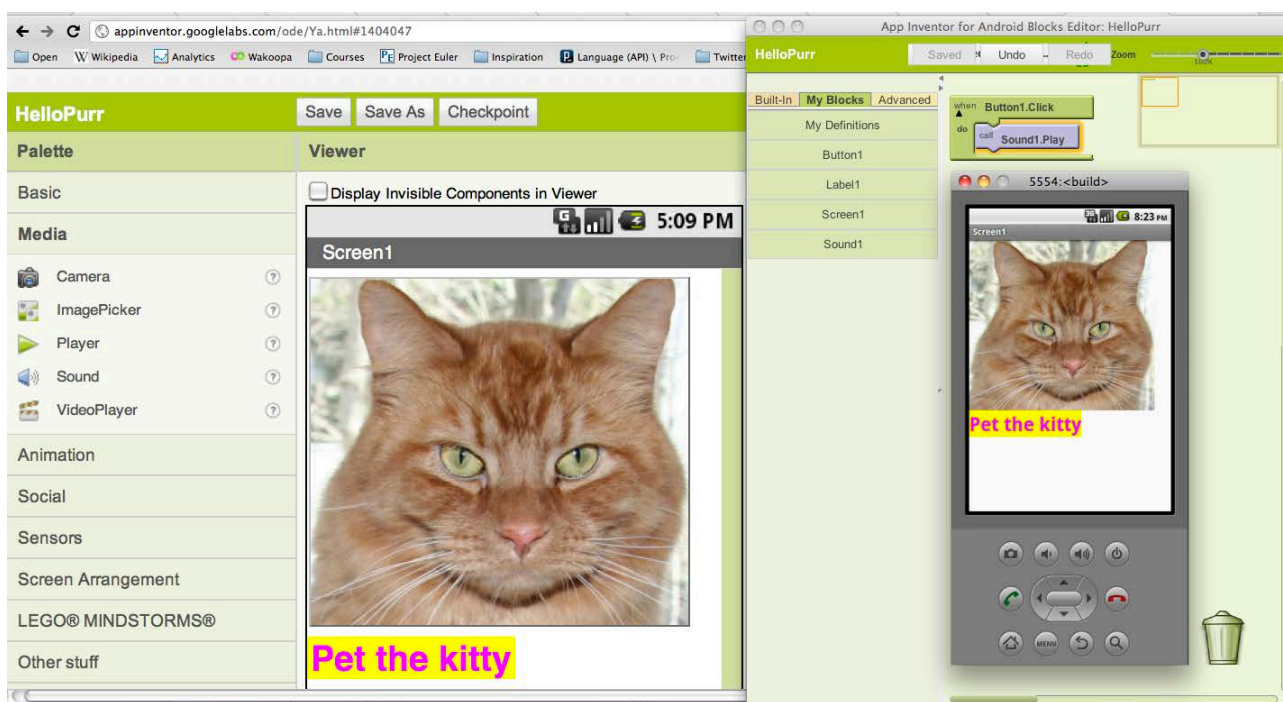


Figura 1.5. Entorno de trabajo de la plataforma ‘MIT – App Inventor’

- Se organiza en módulos llamados ‘Retos’. Los retos son una forma directa y motivadora de que el estudiante logre metas claras, a través de la realización de una serie de tareas prácticas. Así, cada ‘Reto’ incluye una serie de videos y tareas que el estudiante deberá completar para superarlo, junto con sus actividades y pruebas o tests. Antes de empezar un ‘Reto’, el

²⁴ Sólo se ha podido recuperar en PDF la versión de la guía de finales del año 2013, momento en que el curso se encontraba alojado en la plataforma unX: http://www.colmenia.org/documents/17669/27811/CSV12002_Gu%C3%ADa_aprendizaje_v13.pdf/60746a95-c70a-4941-94f5-681b0a2e3cc5

estudiante conoce qué va a aprender tras superarlo, en qué consiste el reto, cuál es el criterio de evaluación que se utilizará y el tiempo de dedicación que supondrá su superación.

- La metodología del curso se basa en un modelo de resolución de problemas donde la motivación es muy importante. En este sentido, tanto el seguimiento como la tutoría se basan en el sistema de aprendizaje *P2P* (*'peer to peer'*), o aprendizaje entre pares, en el que el estudiante puede comentar las actividades con otros usuarios. Además de la interacción con los compañeros, el estudiante cuenta con la ayuda de un dinamizador y con un claustro de profesores.
- La duración estimada del curso es de 1 mes y medio. Los 6 retos propuestos pueden resolverse en un tiempo aproximado de 6 semanas. Se estima que cada reto puede resolverse entre 2,5 horas hasta 1 semana de duración cada uno, dependiendo del grado de experiencia del alumno. En este sentido, la metodología que se utiliza es flexible y asíncrona para que el alumno ajuste autónomamente su estudio a su tiempo disponible.
- El curso pone a disposición del estudiante todo un repertorio de materiales complementarios y herramientas de comunicación e intercambio 2.0 para facilitar el aprendizaje del estudiante: mediateca, tablón de noticias, videos de héroes o emprendedores que cuentan su experiencia, ejercicios con feedback, wiki, foro, blog del curso, etc.
- Respecto a la evaluación y la recompensa del aprendizaje del estudiante, a lo largo del curso, en la medida en que el alumno va superando retos y desafíos educativos, consigue *'badges'*²⁵ (insignias, distintivos): símbolos visibles en el mundo digital que acreditan las competencias y habilidades adquiridas y definen su perfil académico-profesional. Además, la participación del estudiante en la comunidad de aprendizaje le permite obtener *'Karma'*: prestigio que cada individuo tiene en la comunidad, adquirido a través de su actividad social en la propia plataforma y que se traduce en un marcador numérico en el perfil de cada miembro de la misma.
- Se ofrece tutorización esporádica de expertos a través de los profesores y el dinamizador del curso.

Por tanto, parece que el Curso MOOC *"Emprendimiento y App Inventor"* (MiriadaX, 2015) cumpliría con los tres pilares básicos de la metodología óptima de aprendizaje para los más capaces, señalada algo más arriba: metodología a distancia con soporte tecnológico; con contenidos bien definidos y modulares que permitan el aprendizaje autónomo del alumno; y con sistema de apoyo tutorial o mentores, e interacción en grupo con iguales en capacidad.

Sin embargo, el anterior curso está orientado en última instancia a público adulto, y al ámbito empresarial y al emprendimiento. En ese sentido nos hemos dispuesto a buscar otro entorno MOOC de perspectiva más educativa, orientada especialmente a una audiencia joven, y que pueda integrarse en la práctica escolar; y hemos encontrado un magnífico ejemplo, Khan Academy, que desarrollamos a continuación.

²⁵ https://en.wikipedia.org/wiki/Digital_badge

1.2.1. Un ejemplo de plataforma en línea para los más capaces: Khan Academy

Desde hace unos años vienen proliferando por Internet plataformas educativas que tratan de proporcionar en línea a los estudiantes de todo el mundo sus productos multimedia, integrando tecnología de vídeo y texto, con un trabajado esquema didáctico que pretende ser un nuevo camino para el aprendizaje, en muchos casos como sustitutos del libro de texto convencional. Éste es el caso, por ejemplo, de *“Thinkwell: online video learning specialists”*²⁶ cuyos productos se utilizan en multitud de países y niveles educativos; y sus cursos se orientan a dos ámbitos preferentemente: como ayuda al estudio y trabajo en casa, y como una herramienta para los *‘homeschoolers’*, aquellos alumnos que por diversas razones se escolarizan en su hogar, total o parcialmente.

Sin embargo, plataformas como la anterior son de pago y no están pensadas para ser integradas con el aula ordinaria. Estos inconvenientes han sido superados por la plataforma educativa *‘on-line’*, abierta y gratuita, llamada Khan Academy²⁷

Uno de sus lemas proclama su máxima aspiración: *“A free world-class for anyone anywhere”* (“Una clase gratuita y a escala mundial para cualquiera en cualquier lugar”). Y en su información corporativa Khan Academy dice ser:

“...una organización con una misión. Somos una organización sin ánimo de lucro con el objetivo de cambiar la educación a mejor, proporcionando una educación gratuita y de talla mundial, para cualquier persona en cualquier lugar (...) Todos los recursos del sitio web están disponibles para cualquier persona. No importa si usted es un estudiante, un profesor, el director de un centro, un adulto que vuelve a las aulas después de 20 años (...) Materiales y recursos de Khan Academy están a su disposición de forma totalmente gratuita (...)

*(...) Khan Academy ofrece ejercicios prácticos, videos instruccionales-educativos, y un panel de aprendizaje personalizado que posibilita y empodera a los aprendices para el estudio tanto dentro como fuera del aula”*²⁸

Su fundador, Salman Khan, explica magistralmente el origen y la filosofía de este proyecto en una conferencia incluida en la prestigiosa serie *‘TED Talks’* (Khan, 2011), en la cual habla sobre:

“...cómo y por qué creó la extraordinaria Khan Academy, una serie de videos educativos cuidadosamente estructurados que ofrecen completos planes de estudio en matemáticas y, ahora, en otros temas [como, por ejemplo, en programación informática]. Muestra el poder de los ejercicios interactivos e invita a los profesores a considerar invertir el tradicional método en el salón de clases [o metodología ‘flipped classroom’]: Asignar a los estudiantes video-clases para ver en su hogar, y hacer ‘los deberes’ en el salón con el profesor listo para ayudarles.” (Khan, 2011, en línea)

Recientemente, especialistas en la alta capacidad y el talento, como Tourón (2013a) han revisado las características y principios pedagógicos de Khan Academy, y su convergencia con el estilo de

²⁶ <http://www.thinkwell.com/>

²⁷ <https://www.khanacademy.org/>

²⁸ <https://www.khanacademy.org/about>

aprendizaje de los alumnos más capaces; llegando a afirmar que “*estamos ante un hito educativo de primera magnitud*” (Tourón, 2013a, en línea).

Algunas de las características más destacables de Khan Academy (KA) son:

- Es un conjunto de cursos MOOC, en el sentido de que es un entorno ‘*on-line*’, abierto, masivo y gratuito. A todos los recursos se puede acceder de manera externa a la plataforma, aunque conviene que el estudiante se registre (‘*log in*’) para que su actividad quede grabada y así pueda controlar su progreso y tener acceso a las estadísticas de avance.
- Versa sobre diversas disciplinas o áreas de conocimiento: Matemáticas, Ciencias (Química, Física, Biología), Economía, Humanidades o Computación (*Computer Science*).
- Cada área de conocimiento se organiza como un currículum perfectamente organizado, jerarquizado y modularizado. Cada estudiante, previa realización si lo desea de un *pretest* de conocimientos iniciales, se sitúa en un punto determinado de ese conjunto de contenidos, a partir del cual va avanzando a su propio ritmo; avance que queda registrado de manera personalizada. Así, por ejemplo, es destacable el currículum de Matemáticas, que abarca desde la ‘suma de 1 dígito’ hasta ‘derivadas e integrales’: es decir, el currículum completo de Primaria, Secundaria y Bachillerato. En las Figura 1.6 vemos un extracto del mapa de conocimientos (‘*Knowledge Map*’) por el que un estudiante iría avanzando, de manera autónoma y a su propio ritmo, en el área de Matemáticas; y en la Figura 1.7 una relación detallada del progreso de un estudiante en el dominio de las habilidades propias del área (‘*Skill Progress*’)

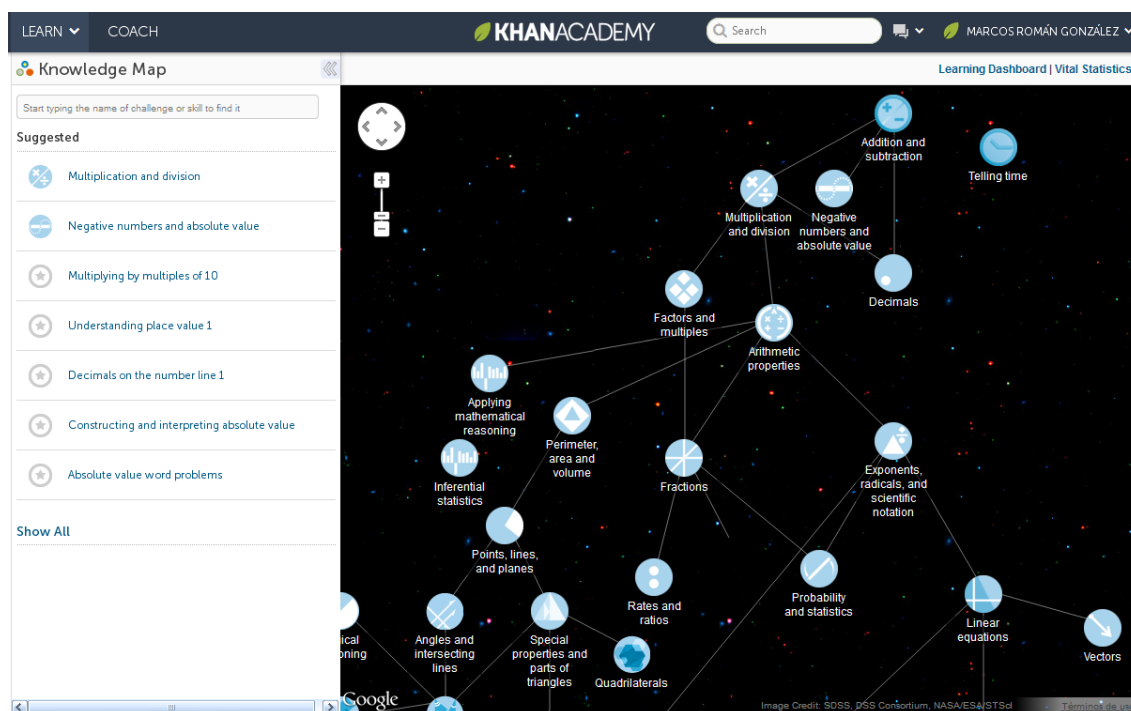


Figura 1.6. Mapa de Conocimiento (‘*Knowledge Map*’) del área de Matemáticas en KA

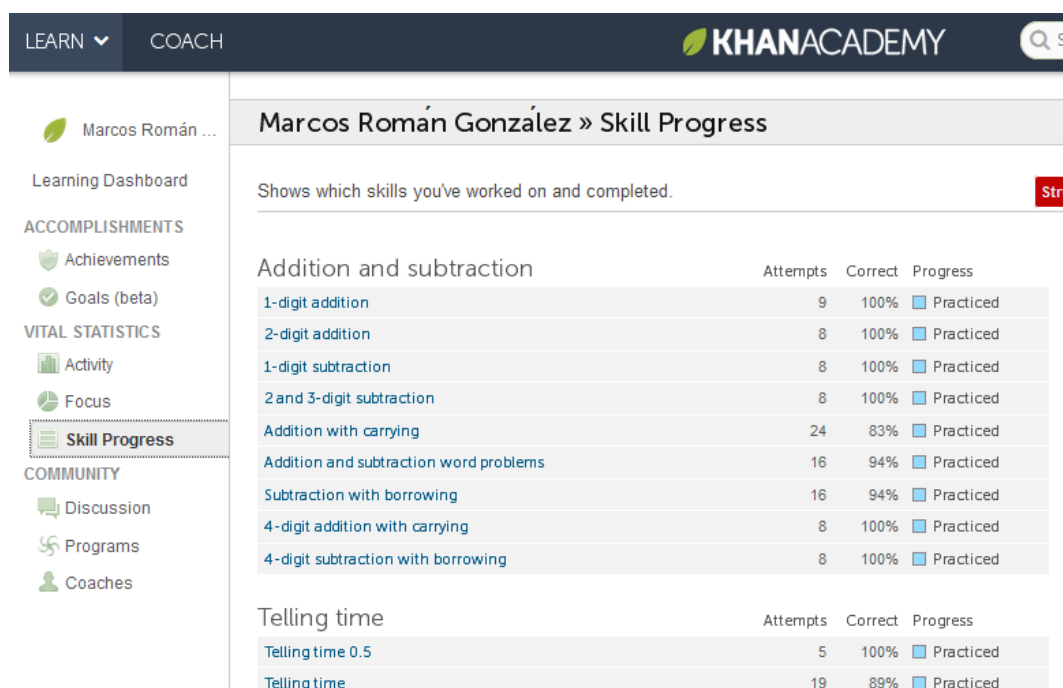


Figura 1.7. Relación detallada del progreso en las habilidades de un estudiante en el área de Matemáticas en KA

- Para el aprendizaje, dominio y avance en cada una de las habilidades que se proponen, el estudiante hace uso de un conjunto de videotutoriales (alojados en abierto en YouTube²⁹), retos interactivos y ejercicios con *feed-back* automático; a los que puede acceder desde cualquier ordenador con acceso a internet. Por ejemplo, en la Figura 1.8. vemos el conjunto de recursos que se disponen para el aprendizaje de la habilidad ‘Basic Division’ (división básica y sin decimales).

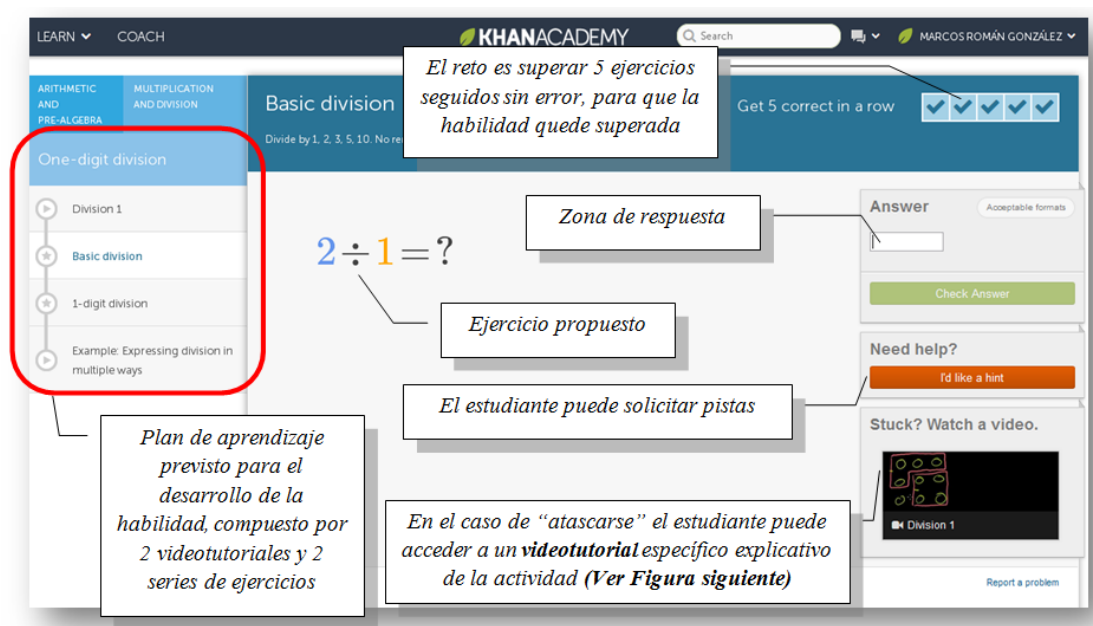


Figura 1.8. Interfaz con el conjunto de recursos para desarrollar la habilidad ‘Basic Division’ en KA

²⁹ El canal de Khan Academy en YouTube [<https://www.youtube.com/user/khanacademy/videos>] acumula un total de 2.289.839 suscriptores y 661.204.401 visualizaciones (datos recuperados el 16 de octubre de 2015) [<https://www.youtube.com/user/khanacademy/about>]

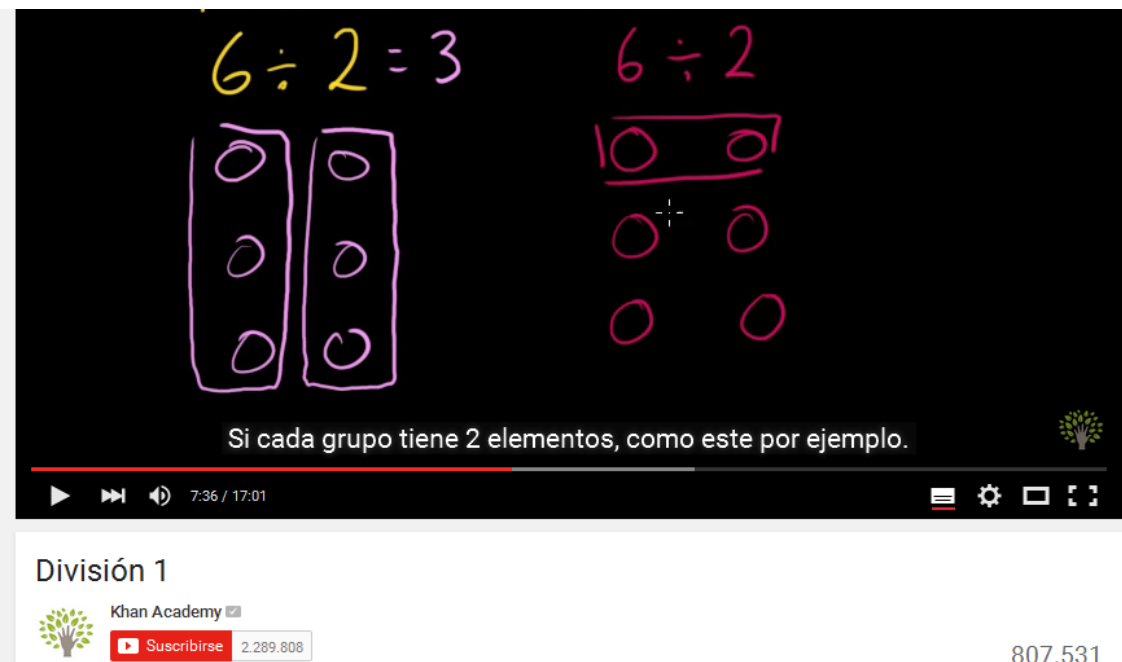


Figura 1.9. Videotutorial de KA en YouTube³⁰, de apoyo al aprendizaje de la habilidad 'Basic Division'

- El estudiante, a medida que supera y domina habilidades de progresiva dificultad, obtiene reconocimiento y reforzamiento en forma de puntos y 'badges' (insignias digitales) en su perfil personalizado (Figura 1.10), incrementándose su reputación y pericia en la plataforma.

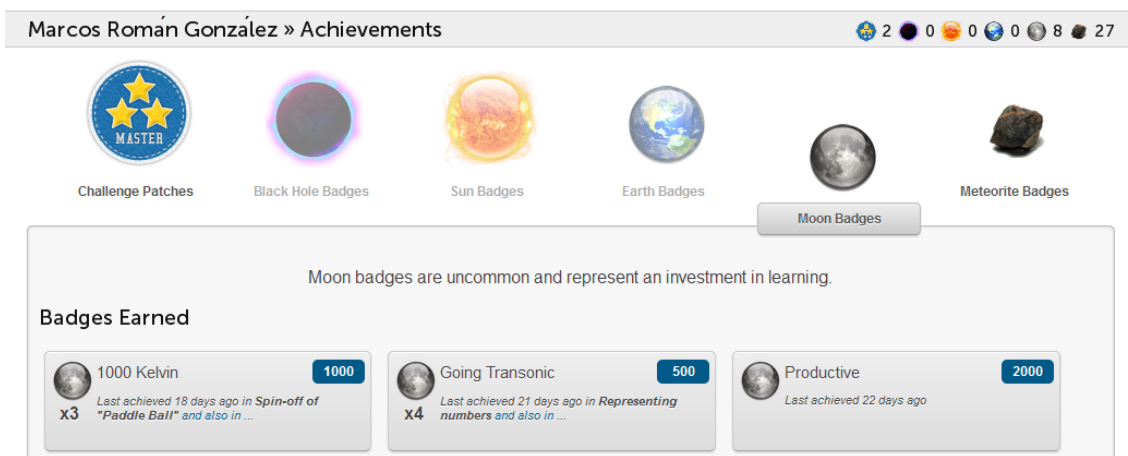


Figura 1.10. 'Badges' (insignias digitales) obtenidas por un estudiante en KA tras el logro de sucesivas habilidades.

- La actividad y progreso del estudiante queda perfectamente recogida en la plataforma, y expresada de manera intuitiva a través de estadísticos gráficos y detallados: día de conexión, habilidades practicadas, % de acierto en ejercicios, videotutoriales visionados, insignias obtenidas, etc. (Figuras 1.11 y 1.12). Este conjunto de datos de seguimiento ('tracking') y análisis del aprendizaje del estudiante ('learning analytics') que proporciona la plataforma Khan Academy, ha sido recientemente mejorado y extendido por investigadores españoles (Muñoz-Merino, Valiente, & Kloos, 2013; RUIPÉREZ-VALIENTE, MUÑOZ-MERINO, LEONY, & KLOOS, 2015)

³⁰ <https://youtu.be/MTzTqvzWzm8>

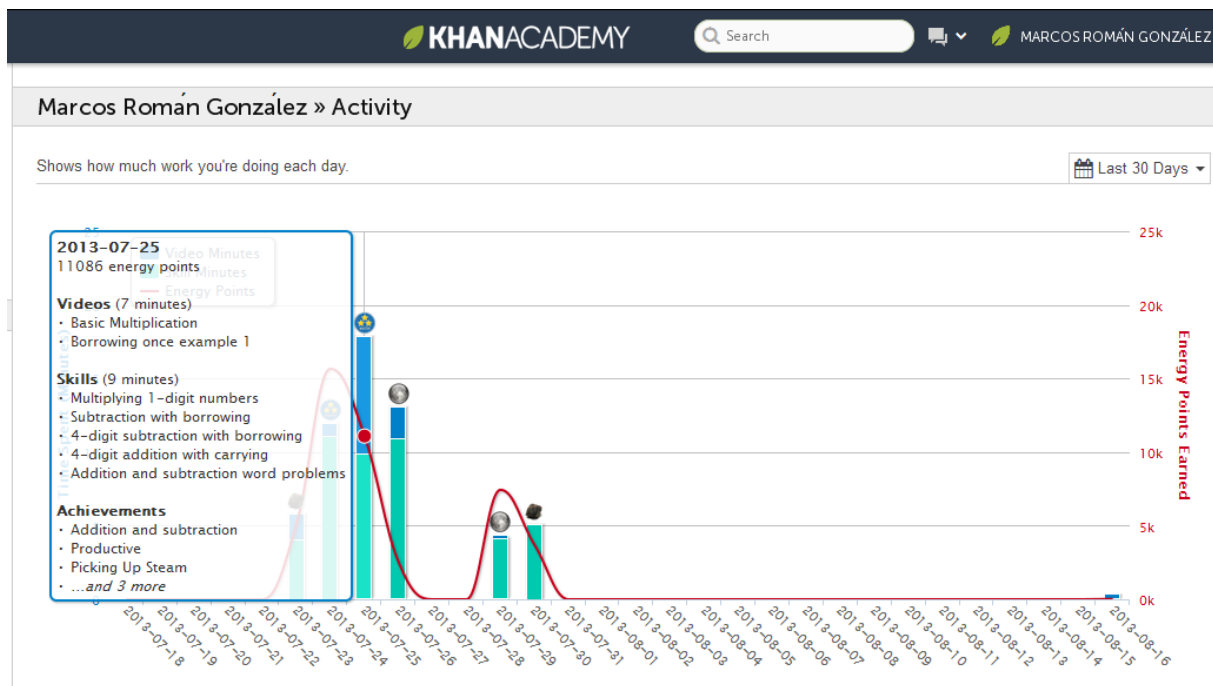


Figura 1.11. Ejemplo de la actividad del estudiante en la plataforma KA a lo largo de ‘los pasados 30 días’

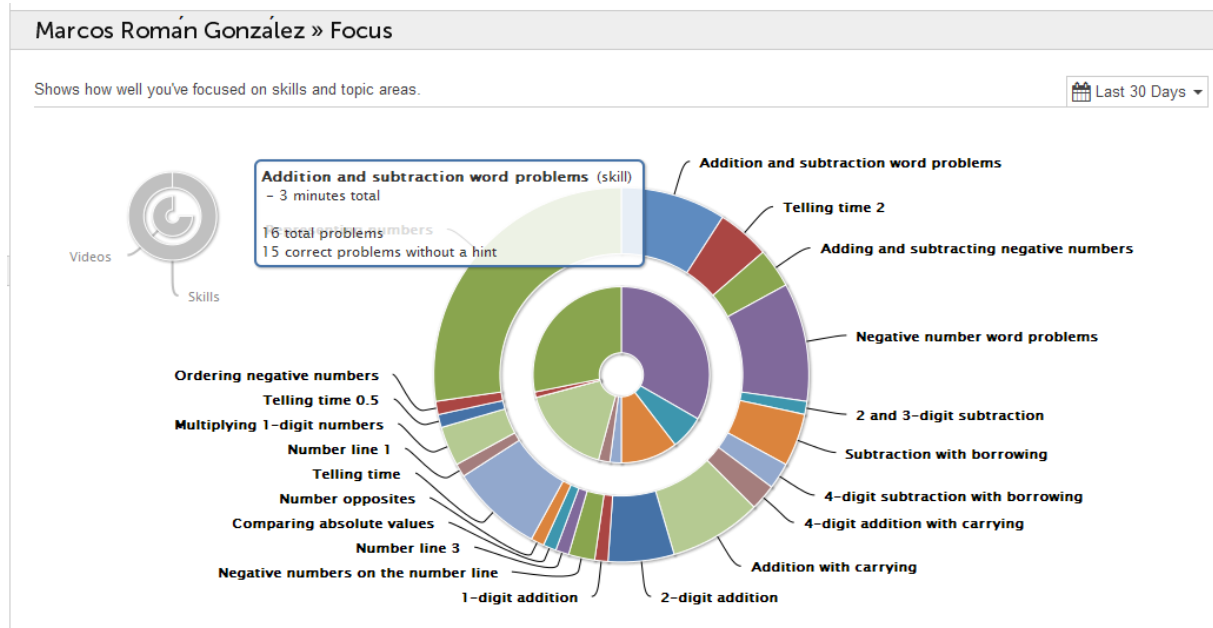


Figura 1.12. Conjunto de habilidades y videotutoriales abordados por el estudiante en ‘los pasados 30 días’, con los respectivos niveles de dominio alcanzados en KA

- Además, los padres, profesores, orientadores o directores de los centros educativos pueden igualmente darse de alta en la plataforma y crear en su interior un ‘aula virtual’ a la que asociar a sus estudiantes; de tal manera que los profesores pueden monitorizar en tiempo real el progreso de sus alumnos y sus estadísticas, así como diferenciar e individualizar su instrucción en función de su avance. Ello posibilita integrar Khan Academy en el aula ordinaria a través de un esquema didáctico de ‘clase invertida’ (*‘flipped classroom’*) tal y como veremos en el siguiente epígrafe.

- En caso de necesitarlo, el estudiante puede contactar a través de foros y pizarras interactivas con otros estudiantes de igual o superior capacidad para ser orientados en su aprendizaje; así como con ‘*coaches*’ (miembros de la plataforma que se ofrecen como entrenadores-tutores).
- Adicionalmente, la plataforma Khan Academy está originalmente en lengua inglesa (aunque progresivamente se están traduciendo sus contenidos y subtitulando sus videotutoriales a otros idiomas), de manera que su uso desarrolla paralelamente las competencias lingüísticas extranjeras de los alumnos españoles.

Así, es tal la magnitud de la irrupción de Khan Academy que tanto desde revistas especializadas en cultura digital (Thompson, 2011) como en medios especializados en economía (Noer, 2012), se afirma que Khan Academy está reinventando y cambiando las reglas para la educación del siglo XXI. Desde un ámbito más académico, Tourón (2013b) expone cómo Khan Academy supone un recurso de primera magnitud para que se pueda realizar una adecuada diferenciación e individualización de la enseñanza-aprendizaje en nuestras aulas. En concreto, nos dice: *“La visión de Khan sobre la clase es muy acertada, y está en línea con la investigación pedagógica solvente que se viene realizando desde décadas atrás sobre las variables críticas en el proceso de enseñanza-aprendizaje”* (Tourón, 2013b, en línea)

Tourón (2013b), recoge los principios metodológicos de Khan Academy, explicados por su propio fundador, que se resumen en los siguientes cuatro puntos, todos ellos en consonancia metodológica con el aprendizaje ideal de los sujetos de alta capacidad:

- a. **‘Mastery-based’** (basado en el dominio): sólo se avanza en las sucesivas habilidades y retos cuando se demuestra un claro dominio de los mismos; habitualmente en forma de series de 5 o 10 ejercicios realizados sin error.
- b. **‘Self-paced’** (al propio ritmo): en palabras del propio Khan, recogidas por Tourón (2013b, en línea): *“Los estudiantes necesitan tiempo y espacio para llegar a dominar los diversos conceptos previos antes de abordar los siguientes. Cuando se les da esta libertad, los estudiantes adquieren confianza y asumen la responsabilidad de dirigir por sí mismos sus experiencias de aprendizaje (...) Lo contrario también es cierto, los alumnos que aprenden más rápido necesitan tener a su disposición la posibilidad de seguir adelante, de otro modo suele producirse tedio y aburrimiento, desinterés y falta de motivación y, finalmente, pereza intelectual”*
- c. **‘Interactive & exploratory’** (interactiva y exploratoria): *“La clase debe brindar a los alumnos la oportunidad de explorar y confrontar problemas de manera tangible”*
- d. **‘P2P Learning: Peer to peer learning’** (aprendizaje entre pares): los pares no son pares en edad, sino pares en capacidad; tal y como siempre han buscado y necesitado los alumnos de alta capacidad.

Como síntesis, en la siguiente Tabla 1.2 tratamos de ilustrar la convergencia entre algunas características propias del estilo de aprendizaje de los alumnos de alta capacidad, y su correspondencia con algunos elementos propios de Khan Academy.

Tabla 1.2. Convergencia entre estilo de aprendizaje de alumnos de alta capacidad y las características de Khan Academy.

<i>Los alumnos de alta capacidad...</i>	<i>Khan Academy ofrece...</i>
<i>Para rendir al máximo nivel necesita ejemplos de excelencia, y expectativas precisas del nivel de ejecución esperado (Tannenbaum, 1991)</i>	<i>Retos y problemas con expectativas explícitas sobre el nivel de ejecución esperado, que es de ‘dominio’ para poder ser superados. Por ejemplo, en el área de programación informática (Computer Science) se ofrecen gran cantidad de modelos de excelencia a partir de los cuales el estudiante puede generar derivados o spin-offs (Figura 1.4)</i>
<i>Preferencia por estar con sus iguales en capacidad, no con sus iguales en edad (Freeman, 1990)</i>	<i>Es una comunidad virtual de aprendizaje en la cual las relaciones y la comunicación se establece en función de la capacidad y la pericia en un campo determinado; no en función de la edad.</i>
<i>Se inclinan hacia un aprendizaje autónomo, independiente y a su propio ritmo, que es favorecido con currículos bien estructurados y organizados, y con paquetes de aprendizaje autodirigidos (VanTassel-Baska, 1989)</i>	<i>Se organiza en áreas de conocimiento, a modo de un conjunto de contenidos perfectamente organizado: secuenciado y jerarquizado; o ‘mapas de conocimiento’ (‘Knowledge maps’) por el que el estudiante avanza a su propio ritmo.</i>
<i>Se benefician extraordinariamente de la tutorización ocasional por expertos, así como de las relaciones de mentoría entre iguales (VanTassel-Baska, 1989)</i>	<i>Ofrece un sistema de aprendizaje entre pares (P2P, peer to peer), con asesoramiento ocasional de ‘coaches’ (tutores)</i>

Hasta este punto, con objeto de describir las características de la plataforma Khan Academy, nos hemos apoyado en ejemplificaciones del área de matemáticas. Nos centraremos ahora en el curso ‘Computer Programming’³¹ (Khan Academy, 2015), por ser el que se relaciona directamente con nuestro problema inicial de investigación; ya que es un curso que permite al estudiante programar y construir ‘web apps’ (gráficos, animaciones o visualizaciones interactivas, que pueden ejecutarse desde un navegador en un dispositivo móvil), así como compartir sus creaciones y explorar los programas realizados por otros. Más concretamente, describiremos con cierto detalle el módulo ‘Intro to JS: Drawing & Animation’³², dado que será el entorno utilizado con posterioridad en el estudio de casos de alumnos con altas capacidades computacionales con el que se cierra la parte empírica de esta tesis doctoral (apartado 7.3)

Dicho módulo ‘Intro to JS’³³: Drawing & Animation’ está compuesto por un itinerario formativo perfectamente delimitado en el que se distinguen 3 tipos de elementos (Figura 1.13): tutoriales de vídeo, tutoriales de ‘consola de programación interactiva’, y retos de programación.

Los tutoriales de vídeo son piezas audiovisuales alojadas en el canal de Khan Academy en YouTube; mientras que los tutoriales de ‘consola de programación interactiva’ son simulaciones en el mismo entorno que utilizará el estudiante para sus retos y proyectos de programación, que reproducen cómo ir utilizando los distintos comandos y parámetros del lenguaje JavaScript & ProcessingJS (Figura 1.3)

³¹ <https://www.khanacademy.org/computing/computer-programming>

³² <https://www.khanacademy.org/computing/computer-programming/programming>

³³ JS son las iniciales relativas a JavaScript, el lenguaje de programación utilizado en el módulo.

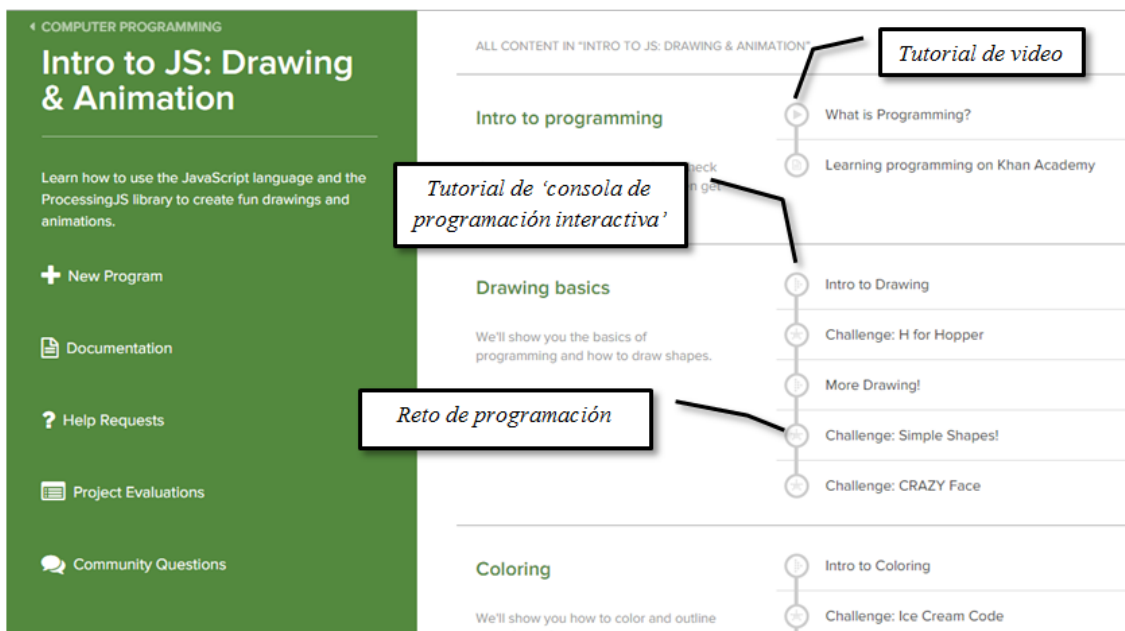


Figura 1.13. Itinerario formativo del módulo 'Intro to JS: Drawing & Animation' en KA

Por ejemplo, en la Figura 1.14 se ilustra uno de estos tutoriales de 'consola de programación interactiva'. Según avanza el tutorial, la tutora va narrando las acciones que desarrolla en la consola de programación (a la izquierda), mientras que la interfaz resultante se va actualizando (a la derecha). Es muy de destacar que la autora y narradora de estos tutoriales, es la miembro de Khan Academy, Jessica Liu³⁴: mujer, 21 años, origen asiático y programadora; rompiendo todos los tópicos sobre la figura del programador y/o el sujeto de alta capacidad aficionado a la informática (varón y de origen caucásico).



Figura 1.14. Tutorial de 'consola de programación interactiva' denominado 'Intro to coloring'³⁵ en KA

³⁴ Foto de Jessica Liu <http://www.khanacademy.org/images/headshots/interns/jessica.png>. Blog de Jessica Liu: <http://cakefordinner.com/>

³⁵ Recuperado de <https://www.khanacademy.org/computing/computer-programming/programming/coloring/p/intro-to-coloring>

El desarrollador de software y diseñador del entorno de trabajo del curso ‘*Computer Programming*’, John Resig, lo describe como una plataforma cuyo objetivo es introducir a las personas, aunque no tengan ningún conocimiento previo, en las tareas y lenguajes de programación; enganchándolas a través de un entorno divertido e interactivo de aprendizaje. Resig (2012) continúa diciendo:

“Por encima de cualquier otra cosa queremos enfatizar la creatividad y la exploración, haciendo la programación accesible a personas de todas las edades, incluyendo jóvenes y niños (...) en nuestra consola de programación interactiva puedes ir introduciendo código y ver en tiempo real el resultado; intuyendo así cómo funciona el lenguaje de programación sin acudir a explicaciones explícitas del mismo” (Resig, 2012, en línea)

En una reciente entrevista en el prestigioso magazine *Computer* de la *IEEE Computer Society* (Severance, 2015), John Resig³⁶ conversa con Pamela Fox³⁷, responsable actual del diseño curricular del curso ‘*Computer Programming*’ (y otros de la plataforma Khan Academy relacionados con Ciencias de la Computación). Pamela Fox es también la autora-narradora de los nuevos tutoriales y retos de programación incorporados recientemente³⁸; así como del nuevo sistema de ‘gamificación’ del curso, que ha sido analizado de manera crítica desde ámbitos de investigación académica (Morrison & DiSalvo, 2014). En la conversación, Resig y Fox muestran acuerdo sobre que el curso ‘*Computer Programming*’ puede comenzar a ser abordado a partir del ‘6th Grade’ (equivalente a 12 años edad, foco evolutivo central de nuestros estudios empíricos)

Finalmente, uno de los jovencísimos colaboradores en el desarrollo inicial del curso ‘*Computer Programming*’, Jamie Wong³⁹ (acaso un estudiante de alta capacidad), explica y se enorgullece en su blog de sus mejores creaciones en la consola de programación (Wong, 2012a), y ofrece un par de razones por las cuales considera que el curso ‘*Computer Programming*’ supone una notable evolución (Wong, 2012b):

- a. **‘Instant gratification’** (gratificación instantánea): la introducción de código en la consola de programación se traduce en tiempo real en un resultado gráfico e interactivo.
- b. **‘Bragging rights’** (derechos de fanfarronear): los programas creados pueden ser instantáneamente compartidos con una comunidad global que los evalúa, los versiona, los reenvía, los mejora.

Parece intuirse que ser capaz de programar se convierte en el mundo actual en una habilidad relevante y de prestigio, de manera transversal a los géneros y las culturas. Argumento que volveremos a retomar en el epígrafe de ‘*Relevancia*’ (ver apartado 1.3)

Pero recordemos, tal y como declarábamos al inicio del capítulo, que nuestra intención es explorar la posibilidad y deseabilidad de aprender a programar aplicaciones para dispositivos móviles (‘apps’), desde nuestros centros educativos y en el marco de la atención a la diversidad, es decir, como un

³⁶ <https://twitter.com/jeresig>

³⁷ <https://twitter.com/pamelafox>

³⁸ Hacemos hincapié en la presencia de estas figuras-modelos femeninos en estos entornos de aprendizaje de la programación informática pues, como tratará de mostrarse a lo largo de la tesis doctoral, la cuestión de género está muy presente en lo que la escritura de código se refiere.

³⁹ <https://twitter.com/jlfwong>

enriquecimiento integrado en el aula ordinaria y articulado con el conjunto del currículo; algo que tradicionalmente no se ha conseguido y ha sido objeto de críticas, dado que el enriquecimiento curricular para sujetos de alta capacidad ha solido tomar la forma de actividades extraescolares y extracurriculares (Jiménez Fernández, 2010).

Por tanto, aún dentro de la exploración de la hipótesis directiva de *viabilidad* llega el momento de preguntarnos, ¿cómo integrar en el aula ordinaria recursos de aprendizaje tan potentes como Khan Academy? En un aula tradicional parece inviable, pero desde el emergente paradigma del *'blended learning'* (*'b-learning'*)⁴⁰, que propone combinar clase física y aprendizaje digital, se abre un camino perfectamente transitable. Más concretamente, se propone explorar el esquema didáctico denominado *'flipped classroom'* (traducido en español como 'clase invertida', 'instrucción inversa' y similares), novedoso e incipiente, como el más apropiado para abordar nuestro objetivo.

Así ya lo propone el propio Salman Khan cuando es requerido sobre la cuestión de cómo integrar su plataforma en el aula ordinaria (Alliance for Excellent Education, 2012); y así es sugerido igualmente por un reciente informe de investigación (SRI Education, 2014) que reporta el impacto de introducir Khan Academy en el aula ordinaria a partir de un estudio en 20 escuelas del entorno de Silicon Valley (California, Estados Unidos), en cursos que van desde *'5th – 10th Grade'* (5º Primaria – 4ºESO en la equivalencia con el sistema educativo español). Los principales resultados de este estudio indican altos niveles de satisfacción, tanto de estudiantes como de profesores, en la implantación de Khan Academy en el aula, especialmente en el área de matemáticas, en donde se observan: mejoras significativas en los resultados académicos, aumento en la confianza de los estudiantes sobre sus habilidades matemáticas, y descenso en los niveles de ansiedad relacionados con la materia. Un 89% de los profesores involucrados declararon que, tras la experiencia, incorporarían definitivamente Khan Academy a sus clases.

1.2.2. Una metodología emergente para los más capaces: la *'flipped classroom'*

Uno de los enfoques pedagógicos que se ha venido abriendo paso en los últimos años de manera decisiva y con prometedores resultados es la denominada *'flipped classroom'* (traducido al castellano como 'aula invertida', 'clase inversa' o similares) o metodología *'flipped learning'* (traducido al castellano como 'aprendizaje inverso', 'aprendizaje volteado', 'aprendizaje al revés' o similares).

Muy recientemente, Tourón, Santiago y Díez (2014) han revisado en el primer monográfico sobre *'flipped classroom'* (FC) en lengua española del que tenemos noticia (*'The Flipped Classroom: Cómo convertir la escuela en un espacio de aprendizaje'*) las características de este modelo, sus ventajas, algunos ejemplos de su implantación y resultados de investigación; en relación con la diferenciación educativa para cualquier estudiante en general, y para los alumnos de alta capacidad en particular. En su prólogo, los autores afirman:

⁴⁰ El aprendizaje semipresencial (de sus siglas en inglés: *Blended Learning* o *B-Learning*) es el aprendizaje facilitado a través de la combinación eficiente de diferentes métodos de impartición, modelos de enseñanza y estilos de aprendizaje, y basado en una comunicación transparente de todas las áreas implicadas en el curso (...) Puede ser logrado a través del uso de recursos virtuales y físicos, mezclados. Un ejemplo de esto podría ser la combinación de materiales basados en la tecnología y sesiones cara a cara, juntos para lograr una enseñanza eficaz [https://es.wikipedia.org/wiki/Aprendizaje_semipresencial]

“Estamos convencidos de que una nueva era se está abriendo para la escuela. Ya no basta con ceñirse a un tema autocontenido en un libro en papel. Es preciso ampliar el foco y permitir que cada alumno vaya tan lejos, tan rápido y con tanta profundidad como su capacidad y grado de dominio le permitan, de modo que la escuela se transforme en un ámbito propicio para el desarrollo del talento de todos los alumnos” (Tourón, Santiago, & Díez, 2014, prólogo)

Y aún más reciente, en el último número monográfico de ‘Revista de Educación’⁴¹ sobre altas capacidades, Tourón y Santiago (2015) analizan las posibilidades que ofrece el modelo de ‘aprendizaje inverso’ para el desarrollo del talento de todos los alumnos y, particularmente, de los más capaces. Los autores dicen que estudian las posibilidades de la ‘*flipped classroom*’ como:

“...un modelo de enseñanza y aprendizaje que recupera para el alumno un papel central convirtiéndolo en protagonista de su propio aprendizaje, de manera que se abre la posibilidad a un desarrollo de los aprendices que respeta su ritmo y profundidad de aprendizaje y les permite, al menos teóricamente, desplazarse por el currículo a la velocidad que su capacidad y nivel de dominio les permite” (Tourón & Santiago, 2015, p. 196)

Veremos a continuación: a) Concepto y características del modelo ‘*flipped classroom*’; b) Ventajas y beneficios del modelo ‘*flipped classroom*’; c) Ejemplos de experiencias de implantación del modelo ‘*flipped classroom*’; d) Resultados preliminares de investigación; y e) Convergencia entre el ambiente de aprendizaje promovido por la ‘*flipped classroom*’ y los alumnos de alta capacidad.

a) Concepto y características del modelo ‘*flipped classroom*’

Tal y como los definen Bergmann y Sams (2012), autores fundacionales y acuñadores del término, el ‘*flipped classroom*’ o ‘*flipped learning*’ es un enfoque pedagógico que transfiere fuera del aula el trabajo de determinados procesos de aprendizaje y utiliza el tiempo de clase, apoyándose en la experiencia del docente, para facilitar y potenciar otros procesos de adquisición y práctica de conocimientos dentro del aula.

Con más detalle, en la web⁴² de referencia en España sobre ‘*flipped classroom*’ (Tourón, Santiago, & Vázquez, 2015), llamada ‘*The Flipped Classroom: Experiencias y recursos para ‘dar la vuelta’ a la clase*’ se define:

“Flipped Classroom es un modelo de trabajo en el aula con el que están experimentando algunos docentes. Si bajo la estructura tradicional el tiempo que estamos en el aula, especialmente en los niveles superiores de secundaria y en enseñanza superior, se dedica a explicar la materia y acercar al alumnado a las ideas fundamentales de cada unidad didáctica, mientras que las tareas se hacen en casa; bajo la estructura que propone la ‘clase del revés’, es precisamente al contrario: en casa los estudiantes acceden a los contenidos mientras que las tareas se desarrollan en el aula (...)

⁴¹ N° 368, disponible al completo en <http://www.mecd.gob.es/revista-de-educacion/numeros-revista-educacion/numeros-antteriores/2015/368.html>

⁴² <http://www.theflippedclassroom.es/>

(...) Para que esto sea posible, el docente facilita a su alumnado materiales audiovisuales, pueden ser vídeos o incluso Podcast, de una duración no superior a cinco minutos a través de los cuales presenta la unidad, las principales ideas o los conceptos fundamentales. Este material puede tratarse de un recurso producido por el propio docente o bien de un recurso ya existente en la red, por ejemplo, Khan Academy (...) Bajo el modelo de la ‘clase del revés’ los docentes no envían problemas a la casa, sino conocimiento, y dejan libre el tiempo del aula para que cada alumno, con su ayuda y la del resto de sus compañeros, pueda trabajar sobre las tareas de cada unidad (...) Los docentes tienen más tiempo en el aula para trabajar con cada estudiante, conocer mejor sus necesidades y sus avances. Por su parte el alumnado tiene la oportunidad de hacer preguntas y resolver los problemas con la guía de sus profesores y el apoyo de sus pares, de modo que se favorece la creación de un ambiente de aprendizaje colaborativo.

En resumen, los principales beneficios de este modelo serían: Permite a los docentes dedicar más tiempo a la atención a la diversidad; Es una oportunidad para que el profesorado pueda compartir información y conocimiento entre sí, con el alumnado, las familias y la comunidad; Proporciona al alumnado la posibilidad de volver a acceder a los mejores contenidos generados o facilitados por sus profesores; Crea un ambiente de aprendizaje colaborativo en el aula; Involucra a las familias desde el inicio del proceso de aprendizaje”

Así, cuando los profesores diseñan y publican el contenido ‘on-line’, el tiempo de clase se libera para que se pueda facilitar la participación de los estudiantes en el aprendizaje activo, a través de preguntas, discusiones y actividades aplicadas que fomentan la exploración, la articulación y aplicación de ideas (Tourón & Santiago, 2015). En las Figuras 1.15 y 1.16 se ilustra gráficamente⁴³ el modelo ‘flipped classroom’:

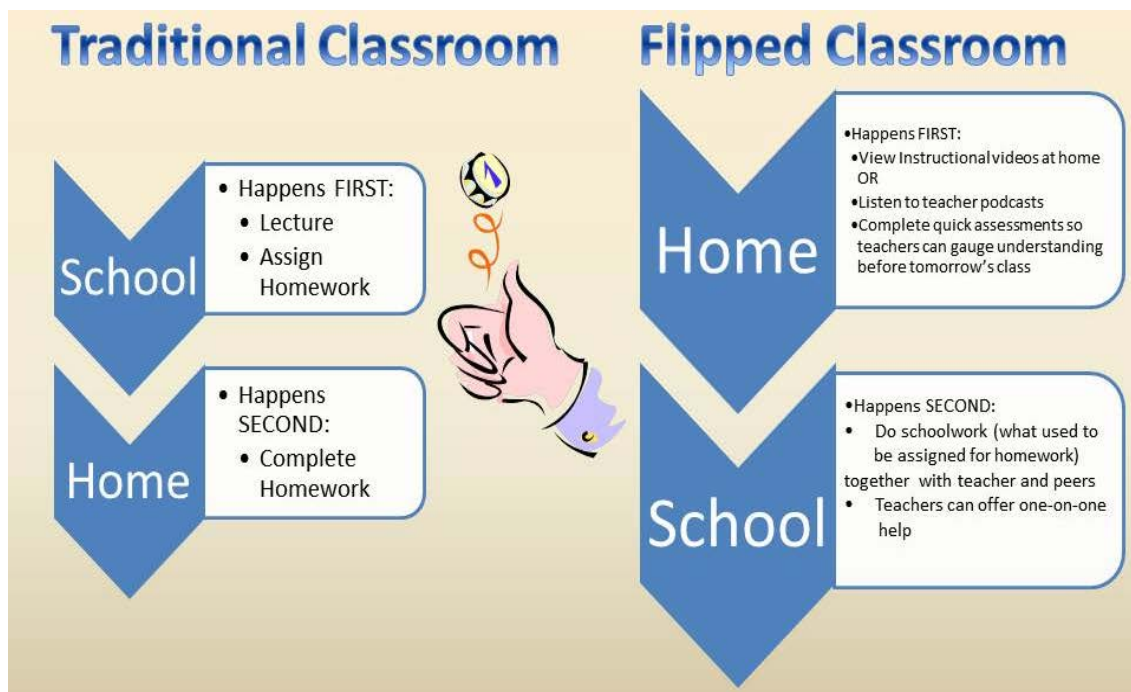


Figura 1.15. Ilustración del modelo ‘flipped classroom’ (1). Tomado de Tourón (2013f)

⁴³ Una infografía completísima sobre el modelo en <https://www.knewton.com/infographics/flipped-classroom/>

Para cerrar la descripción del modelo y prevenir confusiones, Tourón (2013f) nos aclara qué es y qué no es la ‘clase invertida’. Así, la *‘flipped classroom’* **no** es:

- Sinónimo de vídeos *‘on-line’*. Es frecuente que cuando se oye hablar de este modelo se piense inmediatamente en los vídeos, propios o ajenos, que los profesores suelen ofrecer a sus alumnos para que aprendan por sí mismos. Sin embargo, lo que resulta más importante es la interacción y las actividades de aprendizaje significativo que ocurren cuando profesores y alumnos están cara a cara.
- Un procedimiento para reemplazar a los profesores por vídeos.
- Un curso *‘on-line’*.
- Un modelo para que los estudiantes trabajen a su antojo sin estructura o dirección alguna.
- Que los alumnos se pasen todo el tiempo de la clase delante de una pantalla.
- Hacer que los alumnos trabajen solos.

La *‘flipped classroom’* **sí** es:

- Un medio para incrementar la interacción y el tiempo de contacto personalizado entre profesores y alumnos.
- Un ámbito en el que los estudiantes asumen la responsabilidad de su propio aprendizaje.
- Una clase en la que el profesor no es ‘el sabio en el escenario’ (*‘the sage on the stage’*) sino la ‘guía a al lado del alumno’ (*‘the guide on the side’*)
- Una combinación de enseñanza directa y aprendizaje constructivista.
- Un medio por el que los alumnos ausentes, debido a enfermedad u otras actividades que les impiden asistir a clase, pueden seguir el ritmo de desarrollo de las materias.
- Una clase en la que el contenido está permanentemente archivado para que los alumnos lo utilicen en acciones de repaso, recuperación, etc...
- Un modelo en la que todos los alumnos están implicados en su propio aprendizaje.
- Una clase, en suma, en la que todos los alumnos pueden tener una educación personalizada real.

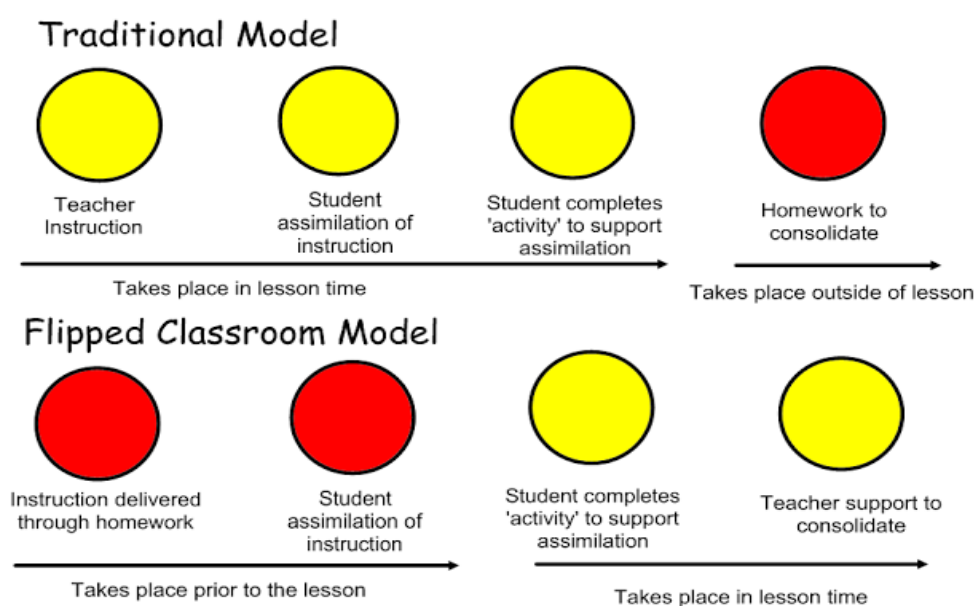


Figura 1.16. Ilustración del modelo *‘flipped classroom’* (2). Tomado de Tourón (2013f)

b) Ventajas y beneficios del modelo ‘*flipped classroom*’

Para este punto, acudimos a un panel de expertos⁴⁴ profesores universitarios que han adoptado e investigado sobre el modelo ‘*flipped classroom*’; panel organizado recientemente por la editorial McGraw-Hill (2013) y aparecido en el influyente blog del profesor Kelly Walsh⁴⁵. En dicho panel se define nuevamente la ‘clase invertida’⁴⁶ y se consensuan algunos de sus beneficios, por ejemplo:

- La ‘clase invertida’ produce una mayor implicación del estudiante: explicar la lección en el aula tradicional es un enfoque de aprendizaje muy pasivo, pero cuando se desplazan las lecciones a un sistema en línea, el tiempo de clase se puede utilizar para la resolución de problemas, las actividades de colaboración y discusión en grupo; incrementándose el compromiso de los alumnos.
- La ‘clase invertida’ produce un aprendizaje más profundo: cuando se mueve la exposición de los contenidos a un sistema en línea y fuera de la clase, se libera una gran cantidad de tiempo para abordar los temas con más detalle y profundidad durante el tiempo de clase. En este sentido, en la Figura 1.17, se ilustra como la ‘clase invertida’ permite invertir un mayor tiempo en clase en las categorías superiores de la taxonomía de Bloom.

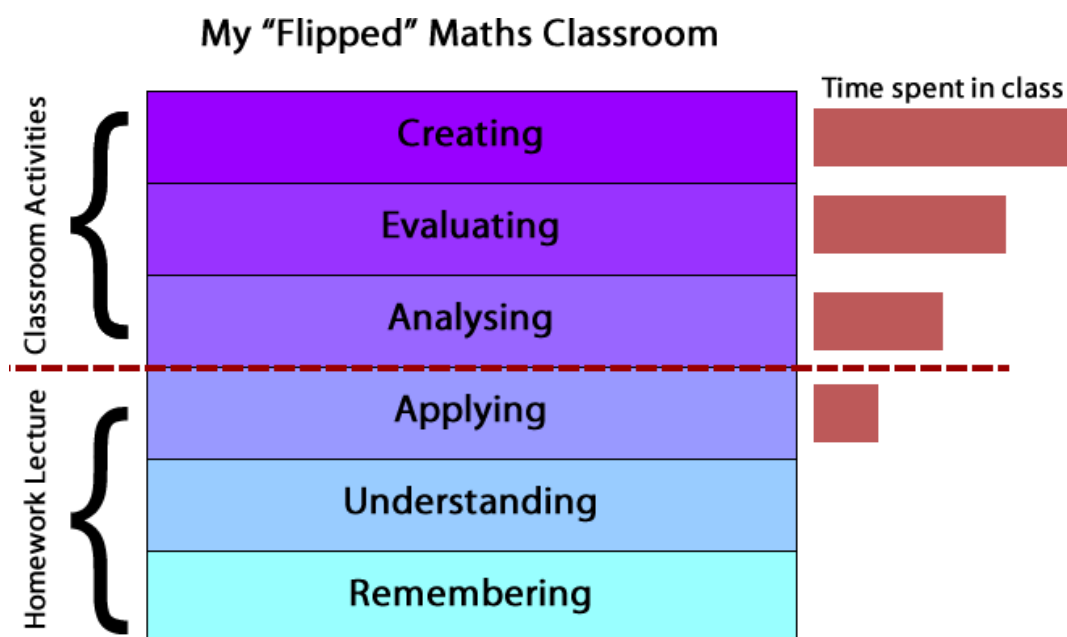


Figura 1.17. Tiempo en la ‘clase invertida’ según la taxonomía de Bloom. Tomado de Tourón (2013e)

- La ‘clase invertida’ permite una mayor adaptación al ritmo de cada estudiante: el aula tradicional generalmente no se adapta a cada alumno (todo el mundo tiene que aprender el tema que se expone básicamente al mismo ritmo). Por el contrario, en el aula invertida, los estudiantes tienen la posibilidad de tener un mayor control de los contenidos de las

⁴⁴ <https://youtu.be/3T8jzCJza0>

⁴⁵ <http://www.emergingedtech.com/>

⁴⁶ “Invertir la clase consiste en hacer que los alumnos ‘consuman’ contenidos de aprendizaje (p.ej. la lección) fuera de la clase, generalmente como trabajo para casa, liberando un tiempo valioso de relación cara a cara con sus profesores realizando las tareas asignadas (trabajos que tradicionalmente suelen ser los deberes) y para reforzar el aprendizaje y profundizar en los materiales adecuados.”

explicaciones y manejar el ritmo (una pausa para tomar notas, o retroceder y aclarar cuando lo necesitan). Esto a su vez libera el tiempo de clase, donde ahora se puede promover un pensamiento de orden superior y aumentar la colaboración y la participación de los estudiantes.

- En un contexto ligeramente distinto, este doctorando ha publicado un artículo (Román-González, 2013c) en la web <http://www.theflippedclassroom.es/> justificando igualmente las ventajas y beneficios del modelo *'flipped classroom'* para profundizar en el enfoque por competencias promovido en el Espacio Europeo de Educación Superior (EEES); y vinculándolo con la metodología más general del *'b-learning'*. Tourón y Santiago (2015) también establecen esta conexión: *"el blended learning (aprendizaje mixto) se puede entender como el modelo más 'maduro' de flipped learning, lo que supone una combinación de herramientas en línea y la instrucción en clase presencial"* (Tourón y Santiago, 2015, p. 212)

Para completar este punto, recogemos las características que, según la experiencia de Bennett, Kern, Gudenrath y McIntosh (2012), reúnen las 'clases invertidas' efectivas y de calidad:

- Los debates y conversaciones están dirigidos por los estudiantes a partir del contenido que han adquirido fuera de la clase; durante ésta se amplía.
- Los debates suelen alcanzar niveles superiores de pensamiento crítico.
- El trabajo colaborativo es fluido, con los alumnos cambiando entre diversas discusiones simultáneas en función de sus necesidades e intereses.
- El contenido se da en un contexto que se refiere, ordinariamente, a situaciones del mundo real.
- Los estudiantes se desafían intelectualmente unos a otros durante la clase respecto al contenido que se discute.
- Se dan formas de aprendizaje dirigidas por los estudiantes a modo de tutoría y de colaboración espontánea entre ellos.
- Los estudiantes se hacen con el contenido y emplean sus conocimientos para dirigirse unos a otros sin que medie necesariamente el profesor.
- Los estudiantes hacen preguntas exploratorias y tienen la libertad para ahondar en temas que van más allá del currículo.
- Los estudiantes participan activamente en la resolución de problemas y el pensamiento crítico se ejercita en ámbitos fuera del curso.
- Los estudiantes se transforman de oyentes pasivos a aprendices activos.

c) Ejemplos de experiencias de implantación del modelo *'flipped classroom'*

En el ámbito anglosajón, especialmente en los Estados Unidos, pueden encontrarse múltiples descripciones de experiencias piloto con el modelo *'flipped classroom'* a través del portal *'Flipped Learning Network'*⁴⁷, una comunidad *'on-line'* en la cual se intercambian experiencias y recursos entre profesores para poder 'invertir' su clases.

⁴⁷ <http://flippedclassroom.org/>

Con respecto a España, en el curso 2012/2013 tuvimos noticia de la primera implantación en territorio español (Hidalgo, 2013): el Colegio San Ignacio de San Sebastián implementó de manera experimental en un par de aulas de la ESO el modelo de ‘clase invertida’ utilizando los recursos de Khan Academy (Figura 1.18)



Figura 1.18. Noticia en prensa de implantación pionera de ‘clase invertida’ en España (Hidalgo, 2013)

Extractamos a continuación lo más destacable de la noticia, que nos parece muy ilustrativa del potencial de la ‘flipped classroom’ para atender a la diversidad en el marco del aula ordinaria:

“Khan Academy ha entrado en San Ignacio por la puerta del taller de Matemáticas de 2º ESO y las clases de Biología de 4º ESO. «Los vídeos, cortos, de entre siete y doce minutos, y en inglés, se ven en casa. El trabajo del profesor es estar en el aula resolviendo las dudas que surgen a los alumnos mientras resuelven los ejercicios que ha propuesto», explica Arzamendi (directora). Sin embargo, esta plataforma es algo más que YouTube disfrazado de colegio. Dos palabras clave: seguimiento y recompensas (...) Cuando un alumno ve un vídeo en Khan Academy, el profesor lo sabe. Cuando se enfrenta a un ejercicio matemático, el profesor sabe cuántas veces lo ha intentado, cuándo lo ha terminado y cuántos más ha hecho después. «Facilita la atención a la diversidad del alumnado», afirma la directora.

El docente sabe quién avanza viento en popa dejando atrás una estela cada vez más larga de respuestas correctas. Sabe quién se ha quedado atascado en un ejercicio y ve en los siguientes un pasillo espinoso. «Podemos dar más trabajo al que tiene menos dificultades o ponerlo en tesitura de acompañar al que encuentra más problemas y potenciar así el aprendizaje cooperativo» (...) Atrás quedan las clases convencionales donde el profesor «explicaba para todos, resolvía algunos ejercicios para todos y corregía para todos». En Khan Academy el lugar del profesor está merodeando entre los pupitres apoyando individualmente el aprendizaje de cada alumno. Mientras tanto, la pizarra digital muestra el

abanico de ejercicios que pueden hacer –determinado previamente por el maestro–: «Cada uno elige su camino, así potenciamos la iniciativa y la autonomía personal», precisa Arzamendi.

¡Y recompensas! Los alumnos no están viendo vídeos ni devanándose los sesos en los ejercicios en vano. Conforme avanzan en los programas propuestos por ‘Khan Academy’, reciben insignias que reconocen su empeño estudiantil y están descritas en la web (...)

*«No te voy a decir que con este sistema el alumno trabaja menos y aprende más. Pero las recompensas permiten mantener en el tiempo la motivación. Empezamos con el proyecto en septiembre, estamos en febrero y el alumno está igual de motivado. De hecho, trabaja más», asegura la directora del centro. Según ella, la posibilidad de que el profesor conozca cada movimiento de cada alumno dentro de la plataforma ha permitido comprobar cómo algunos de ellos se aventuran hacia otros itinerarios y cacharrean entre lecciones de historia del arte o **informática**. «Ven en Khan Academy una ventana impresionante para aprender otras cosas. Lo cierto es que también es un sistema muy adecuado para que los adultos aprendan cultura general», reconoce.» (Hidalgo, 2013, en línea)*

Posteriormente, desde el curso 2013/2014 se han ido conociendo nuevas iniciativas y proyectos piloto en España para implantar la ‘clase invertida’, principalmente en aulas de matemáticas de la ESO; promovidas y recopiladas desde la web⁴⁸ de referencia en España, ya mencionada, sobre ‘*flipped classroom*’ (Tourón, Santiago, & Vázquez, 2015).

d) Resultados preliminares de investigación sobre el modelo ‘*flipped classroom*’

Según afirman Goodwin y Miller (2013), la evidencia sobre el modelo ‘*flipped classroom*’ aún está por llegar, si bien existen evidencias parciales prometedoras, como el informe elaborado por Hamdan, McKnight, McKnight y Arfstrom (2013). Recientemente se ha publicado una extensión del informe citado de Hamdan *et al.*, elaborado por Yarbrow, Arfstrom, McKnight y McKnight (2014), en el que se recogen numerosos estudios de casos que reflejan que las clases, de diversos niveles educativos (desde la enseñanza primaria a la Universidad), experimentan ganancias en rendimiento y satisfacción por parte de profesores y alumnos con el uso de este modelo.

Tourón y Santiago (2015) dan cuenta de una encuesta entre 453 maestros de Estados Unidos que aplicaron el modelo ‘*flipped classroom*’: el 67% informó de un aumento de las puntuaciones en las pruebas de rendimiento, con beneficios particulares para los estudiantes con necesidades educativas especiales; el 80% informó de una mejoría de la actitud por parte de los estudiantes; y el 99% dijo que volvería a utilizar el modelo el año siguiente. En esta misma línea, la Escuela Secundaria Clintondale en Michigan⁴⁹ (pionera a nivel mundial en implantar la metodología ‘*flipped classroom*’ de manera generalizada en sus aulas) comprobó cómo la tasa de fracaso de los estudiantes de matemáticas de grado noveno bajaba del 44% al 13% después de la adopción de la metodología de ‘clase inversa’ (Finkel, 2012).

⁴⁸ <http://www.theflippedclassroom.es/category/experiencias/>

⁴⁹ <http://www.flippedhighschool.com/>

En otro reciente informe de investigación (NFER & Nesta, 2015), de carácter cualitativo, que involucra estudiantes de entre 11 y 14 años de 9 escuelas distintas del Reino Unido, se reportan algunos resultados derivados de la integración de la plataforma Khan Academy en las aulas de matemáticas mediante metodología *'flipped classroom'*. El sumario ejecutivo de dicho informe afirma que la implantación del *'flipped learning'* supuso más tiempo efectivo para: practicar y aplicar los conocimientos y habilidades adquiridos; formular preguntas y mantener debates de mayor nivel cognitivo; fomentar el aprendizaje colaborativo; favorecer un aprendizaje independiente y conducido por el propio estudiante; facilitar apoyo individualizado; incrementar el conocimiento de los estilos de aprendizaje preferidos por cada alumno. Además, y en relación con el área de Matemáticas los estudiantes incrementaron: su compromiso con el aprendizaje; sus conocimientos y comprensión de la materia; su confianza hacia las matemáticas; la conciencia respecto de sus fortalezas y debilidades; sus habilidades para el aprendizaje autónomo; su progreso en la materia y sus logros en la misma (NFER & Nesta, 2015, p. 6)

De manera similar, aunque esta vez referida a profesorado universitario de los Estados Unidos, Tourón (2015) informa sobre una encuesta realizada a 1024 profesores, miembros de la plataforma *'Faculty Focus'*⁵⁰, cuyos principales resultados son:

- El 69,5% de los profesores encuestados han intentado invertir alguna actividad, clase o curso y se plantean hacerlo de nuevo.
- Sólo el 5,49% de los que lo han hecho no se plantean repetir la experiencia.
- La mayor parte de los profesores que han invertido su enseñanza lo han encontrado positivo, tanto para ellos (70,3%) como para sus alumnos (64,8%).
- Las principales razones para invertir sus clases incluyen un deseo de aumentar la participación de los estudiantes (79,3%) y mejorar el aprendizaje de los mismos (75,8%).
- En términos de los beneficios reales, casi las tres cuartas partes veían mayor compromiso de estudiantes (74,9%), mientras que más de la mitad vio evidencias de mejora en el aprendizaje (54,66%).
- Más del 80% señaló que los estudiantes eran más colaborativos y el 76,61% dijo que hacían más preguntas, mientras que casi la mitad (48,75%) también observó cierta resistencia estudiantil.
- La barrera más frecuentemente para el profesorado que quiere probar la inversión es el tiempo limitado del que dispone.
- Casi el 70% dijo que era un reto. Muy significativo (38,1%) o un reto importante (31,61%).
- De los encuestados que no están interesados en el aprendizaje inverso, el 38,9% dijo que no saben lo suficiente sobre él y el 27,4% consideró que era una moda pasajera.

En cualquier caso, parece que en el momento actual, *“es preciso llevar a cabo mucha más investigación para determinar de manera rotunda si el modelo 'flipped classroom' mejora directamente el aprendizaje del estudiante, pero la 'ausencia de evidencia no es evidencia de ausencia'”* (Tourón & Santiago, 2015, p. 223)

⁵⁰ <http://www.facultyfocus.com/>

- e) Convergencia entre el ambiente de aprendizaje promovido por la *'flipped classroom'* y los alumnos de alta capacidad.

Recientemente, desde la *'Flipped Learning Network'*⁵¹ se han fijado los cuatro pilares metodológicos del modelo (acrónimo de FLIP):

- *'F: Flexible environment'* (*'Ambiente flexible'*): Las aulas invertidas permiten una gran variedad de modos de aprendizaje; los profesores a menudo reordenan físicamente su espacio de aprendizaje (su clase) para dar cabida a la lección o unidad, lo que puede implicar trabajo en grupo o estudio independiente. Crean ambientes flexibles en los que los estudiantes eligen cuándo y dónde aprenden. Además, los educadores que invierten sus clases son flexibles en sus expectativas respecto al tiempo que los alumnos necesitan para su aprendizaje y a cómo son evaluados.
- *'L: Learning Culture'* (*'Cultura de aprendizaje'*): En el modelo tradicional centrado en el profesor, éste es la fuente principal de información. En el modelo de aprendizaje invertido, hay un cambio deliberado de un aula centrada en el profesor a un enfoque centrado en el estudiante, donde el tiempo de clase es para explorar temas con mayor profundidad y para la creación de oportunidades de aprendizaje más ricas, a través de diversas pedagogías centradas en el estudiante. Como resultado, los alumnos participan activamente en la formación del conocimiento, a través de las oportunidades de participación y de la posibilidad de evaluar su aprendizaje de una manera que es personalmente significativa.
- *'I: Intentional Content'* (*'Contenido intencional'*): Los profesores *'flipped'* continuamente están pensando en cómo se puede utilizar el modelo de aprendizaje *'flipped'* para ayudar a los estudiantes a adquirir una comprensión conceptual, así como la fluidez de procedimientos. Evalúan lo que necesitan para enseñar y qué materiales deben explorar los estudiantes por su cuenta. Los educadores usan contenido intencional para maximizar el tiempo de clase con el fin de adoptar diversos métodos de enseñanza, como estrategias activas de aprendizaje, la enseñanza entre pares, el aprendizaje basado en problemas, o los métodos de dominio (*'mastery'*), según el nivel educativo y la materia.
- *'P: Professional Educators'* (*'Educadores profesionales'*): El papel de los educadores profesionales es aún más importante y a menudo más exigente en un aula invertida que en una tradicional. Durante el tiempo de clase, los maestros observan continuamente a sus alumnos proporcionándoles información relevante en el momento, así como la evaluación de su trabajo. Los educadores profesionales reflexionan sobre su práctica, se relacionan entre sí para mejorar su trabajo, aceptan la crítica constructiva y toleran en su clase un cierto caos controlado. Aunque los educadores profesionales siguen siendo muy importantes, asumen roles menos visibles en el aula invertida.

A este respecto, observamos una clara convergencia entre el ambiente de aprendizaje que se deriva del modelo *'flipped classroom'* y el estilo de aprendizaje característico de los alumnos de alta capacidad. Se condensa en la siguiente Tabla 1.3.

⁵¹ <http://www.flippedlearning.org/summary>

Tabla 1.3. Convergencia entre estilo de aprendizaje de alumnos de alta capacidad y el modelo 'flipped classroom'

<i>Los alumnos de alta capacidad...</i>	<i>El modelo flipped classroom...</i>
<i>...necesitan un ambiente de aprendizaje flexible: centrado en el estudiante en vez de en el profesor; que promueva la independencia del estudiante; que sea abierto a materiales y experiencias; de aceptación en vez de juicio; complejo y con alta movilidad...</i>	<i>...promueve un ambiente flexible ('Flexible Environment') y centrado en el aprendizaje del estudiante ('Learning Culture')</i>
<i>...necesitan un marco metodológico general que cumpla: Una redefinición del tiempo escolar para proveer oportunidades de tiempo distinto a los alumnos de cara a que completen las tareas y logren los resultados planeados; Currículos bien definidos y estructurados que provean la necesaria secuencialidad y opciones múltiples de aprendizaje; Disponibilidad de materiales accesibles a los alumnos y que pueden proveer una variedad de medios para lograr el dominio deseado; Planes de trabajo individualizados para cada estudiante; Estrategias de retroalimentación que permitan el control periódico del progreso y facilite la toma de decisiones; Reorganización del centro y del aula para que exista mayor flexibilidad en la organización del equipo docente, del espacio y en la asignación del tiempo (Wang y Lindvall, 1984)</i>	<i>...la 'flipped classroom' redefine y amplía el tiempo de aprendizaje en el aula; define en mayor medida los objetivos de instrucción; selecciona y provee al estudiante de una variedad de recursos y materiales accesibles; individualiza la instrucción a través de una retroalimentación personalizada del docente; y dota al ambiente de aprendizaje de una mayor flexibilidad.</i>
<i>...necesitan trabajar en los niveles superiores de la taxonomía de objetivos de Bloom (1972)</i>	<i>...en casa, a través de los materiales on-line se trabajan los niveles inferiores de la taxonomía de Bloom (Recordar, Comprender, Aplicar); liberando así tiempo de aprendizaje en el aula para los niveles superiores (Analizar, Sintetizar, Crear, Evaluar)</i>
<i>...necesita una enseñanza-aprendizaje a su propio ritmo (self-paced)</i>	<i>...permite una mayor adaptación al ritmo de cada estudiante: en el aula invertida, los estudiantes tienen la posibilidad de tener un mayor control de los contenidos de las explicaciones y manejar el ritmo (una pausa para tomar notas, o retroceder y aclarar cuando lo necesitan)</i>
<i>...son especialmente proclives a relaciones de aprendizaje por pares y trabajo colaborativo, entre compañeros de igual capacidad; así como a las relaciones de tutorización y mentoría</i>	<i>...libera tiempo de clase, donde entonces se puede promover un pensamiento de orden superior y aumentar la colaboración y la participación de los estudiantes... ...en las 'clases invertidas' eficaces y de calidad el trabajo colaborativo es fluido, con los alumnos cambiando entre diversas discusiones simultáneas en función de sus necesidades e intereses (Bennett, Kern, Gudenrath, & McIntosh, 2012)</i>
<i>...se orientan hacia la resolución de problemas y el aprendizaje de dominio ('mastery')</i>	<i>...es posible maximizar el tiempo de clase con el fin de adoptar diversos métodos de enseñanza, como estrategias activas de aprendizaje, la enseñanza entre pares, el aprendizaje basado en problemas, o los métodos de dominio ('mastery'), según el nivel educativo y la materia...</i>
<i>...necesitan profundizar, sin 'efecto techo', en los temas que son de su interés.</i>	<i>...en las 'clases invertidas' eficaces y de calidad los estudiantes hacen preguntas exploratorias y tienen la libertad para ahondar en temas que van más allá del currículo (Bennett, Kern, Gudenrath, & McIntosh, 2012)</i>

En suma, un aula inversa permite potencialmente al profesor “ *fijar el contenido por semanas, meses o el curso entero, permitiendo a los estudiantes que aceleren su aprendizaje a través del plan de estudios si están listos*” (Tourón & Santiago, 2015, p. 223). Parece cristalizar el tan añorado sueño de la diferenciación educativa, en su grado máximo personalizada para cada individuo, y ahora posibilitada por los avances de la tecnología digital en línea. Finalizan estos autores diciendo:

“...*el flipped learning nos pone en la senda de una escuela nueva que promoverá, como objetivo esencial, la personalización del aprendizaje para que cada alumno pueda progresar con la velocidad, profundidad y amplitud que sus condiciones intelectivas, afectivas y emocionales le permitan*” (Tourón & Santiago, 2015, p. 225)

Así pues, y siguiendo la ruta argumental que nos hemos fijado (Figura 1.1), consideramos que existe una clara convergencia entre las características del *proceso* de aprendizaje que nos proponemos (‘programación de *apps*’) con arreglo al estilo de los alumnos de alta capacidad; y los medios, métodos y tecnologías actualmente disponibles. Encontramos pues apoyo para enunciar de manera fundamentada la hipótesis directiva de *viabilidad*.

1.3. Criterio 3: Relevancia

En los últimos cinco años, las aplicaciones para dispositivos móviles o ‘*apps*’ han ido adquiriendo de manera vertiginosa una emergente relevancia, configurándose como productos de creciente uso, utilidad y valoración social; y que son puestos a disposición de inmensas audiencias reales y globales.

Así, la tienda en línea de aplicaciones de Apple⁵², o ‘*App Store*’, que funciona en 155 países, dispone de un catálogo de más de 850.000 ‘*apps*’, de ellas 350.000 son para tabletas iPad (Figura 1.19). Desde su creación en 2007, ‘*App Store*’ acumula 45.000 millones de descargas, en ritmo creciente y exponencial. El volumen durante el primer trimestre de 2013 es gigante: 800 descargas por segundo, casi 70 millones al día (Jiménez Cano, 2013); ritmo que va en aumento. El otro gran repositorio de ‘*apps*’ es ‘*Google Play*’⁵³, para dispositivos móviles con sistema operativo Android. Actualmente, pese a que en el parque mundial de móviles el 64% son Android y solo el 19% iPhone, el perfil del usuario de iPhone es un cliente tecnológicamente más avanzado, que descarga más aplicaciones (y no solo gratuitas). Así, en el primer trimestre de 2013, los ingresos de la tienda de Apple fueron 2,6 veces mayores que la de ‘*Google Play*’ (Jiménez Cano, 2013).

En el ámbito español, diversos informes recientes reportan el uso creciente de los dispositivos móviles con conexión a Internet o *smartphones*. Así, según el *Estudio sobre seguridad y privacidad en el uso de los servicios móviles por los menores españoles* realizado por el INTECO (Instituto Nacional de Tecnologías de la Comunicación): en España, la edad media de inicio en la telefonía móvil se sitúa entre los 10-12 años, y el 31% de usuarios de más de 13 años usan *smartphones* (teléfonos inteligentes). El informe destaca además el espectacular avance en el uso de los servicios avanzados que ofrecen este tipo de teléfonos a través de las ‘*apps*’ que en ellos pueden descargarse e

⁵² Para dispositivos móviles con sistema operativo iOS: iPhone (*Smartphone*) y iPad (*Tablet*)

⁵³ <https://play.google.com/store/apps?hl=es>

instalarse: redes sociales (del 7,1% en 2010 se ha pasado al 54,3% en 2011), mensajería instantánea (del 12,4% al 48,3%) y juegos (del 51,6% al 65%) por citar solo algunos (Pérez San-José, 2010)



Figura 1.19. Oferta de 'apps' en 'App Store'

Más recientemente, el último informe publicado por la Fundación Telefónica (2014), *La Sociedad de la Información en España 2014*, reporta que: en España hay 26,2 millones de usuarios de Internet, esto es, el 76,2% de la población accede regularmente a la Red, porcentaje que se dispara al 98,3% si hablamos de la franja entre 16 y 24 años; existen en España 31.700.000 líneas de banda ancha móvil asociadas a *smartphones*, suponiendo por primera vez un volumen de facturación superior a la banda ancha fija; en un 88,8% de los hogares españoles hay, al menos, un *smartphone*, porcentaje que se sitúa en el 54,4% de hogares si nos referimos a *tablets*; España es el país de la Unión Europea con el mayor porcentaje de *smartphones* (el 81%) sobre el total de teléfonos móviles existentes; los *smartphones* se consolidan frente al PC (ordenador fijo): por primera vez, el tiempo de acceso a medios digitales utilizado en dispositivos en movilidad (*smartphones* + *tablets*) supera al empleado en el PC (53% frente a 47%); las aplicaciones que triunfan para dispositivos móviles son aquellas relacionadas con la mensajería instantánea. En síntesis, el informe declara que: “*La digitalización es imparable. Vivimos en hogares cada vez más conectados y organizamos el día con ayuda de nuestros smartphones y apps adaptadas a cada necesidad. Una de cada tres personas consulta el móvil cada 5 minutos (...)*” (Fundación Telefónica, 2014)

Aún más reciente en prensa (Agencia EFE, 2015) se da cuenta del VII Estudio Anual de Mobile Marketing⁵⁴ con más datos reveladores de la expansión de las tecnologías móviles en nuestras vidas: un total de 15,4 millones de personas en España utilizan un teléfono inteligente (un 95% de la población con móvil), lo que supone un aumento del 35% de usuarios si se compara con los datos obtenidos en el año 2012; el teléfono inteligente ya es el principal dispositivo para acceder a internet

⁵⁴ <http://www.abc.es/gestordocumental/uploads/internacional/EstudioMobile2015.pdf>

(es la vía de acceso en un 85% de las ocasiones); en el último año estos dispositivos, en lo relativo a su ‘uso diario’, han dejado de lado a los ordenadores, que quedan en segundo lugar con un 67% y a las tabletas electrónicas, que se afianzan como tercera opción, en un 45%; además, el estudio revela que el 80% de los internautas usan su teléfono móvil como segunda pantalla cuando ve la televisión. En cuanto al uso de aplicaciones, el usuario de *smartphone* tiene una media de trece ‘apps’ en su teléfono inteligente, en donde destaca la aplicación de mensajería instantánea WhatsApp (presente en un 75% de los casos), Facebook (48%), las aplicaciones del tiempo meteorológico que triplican su uso este año y se sitúa en un 25%, y el uso de aplicaciones de banca móvil.

Globalmente, puede afirmarse que la inmersión de nuestra actividad diaria en un ecosistema digital móvil es progresivo e imparable, hasta el punto de que *“en el mundo ya hay casi tantos teléfonos móviles como personas (...) el número de smartphones sigue creciendo, ya existen en el mundo casi tantos teléfonos móviles (6.800 millones) como personas (7.100 millones)”* (PuroMarketing, 2014, en línea)

En este contexto, no es de extrañar que la figura del programador de ‘apps’ esté cobrando auge social y económico. Saltan a la prensa noticias como la de Nick D’Aloisio⁵⁵, un adolescente de 17 años que recientemente ha diseñado una ‘app’ llamada ‘Summly’, que reduce el tamaño de las noticias para hacerlas más fáciles de utilizar en la pantalla de un móvil, y que ha vendido a Yahoo! Por 23 millones de euros (Agencia EFE, 2013). Nick D’Aloisio, residente en Wimbledon, al sur de Londres, y aún estudiante de instituto, desarrolló la ‘app’ en su propia habitación.

Otro caso de máximo interés es el del programador ‘prodigio’ de 14 años, Santiago González, del que hemos tenido noticia a través del canal de YouTube ‘THNKR’⁵⁶, canal especializado en mostrar casos extraordinarios de jóvenes prodigio que brillan por su talento y esfuerzo en diversos campos (Figura 1.20): Santiago González, de 14 años de edad, de nacionalidad estadounidense con ascendencia latina, programador y desarrollador de software habla con fluidez en una docena de diferentes lenguajes de programación y ya cuenta con 15 aplicaciones originales iOS a su nombre. Miles de personas han descargado sus aplicaciones para Mac, iPhone e iPad. A los 12 años, Santiago ya pasó a convertirse en estudiante universitario a tiempo completo y está en camino de obtener su licenciatura en Ciencias de la Computación e Ingeniería eléctrica a los 16 años (THNKR, 2013a) En el vídeo citado también se relata cómo los padres de Santiago le ayudaron a superar un sistema escolar rígido para conseguir desplegar su talento.

En un segundo vídeo (THNKR, 2013b), Santiago nos muestra tres de sus aplicaciones de iPad favoritas y habla de su proceso de diseño: son las aplicaciones ‘Slide Puzzle’, ‘Christmas Tree’ y ‘Solar System’; que están disponibles en ‘App Store’ o directamente a través de su web⁵⁷ (Figura 1.21). Santiago relata cómo le emociona y motiva poner sus ‘apps’ a disposición de un público global, y obtener la valoración y feedback de los usuarios.

⁵⁵ Es interesante ver cómo su foto no se corresponde al estereotipo clásico de ‘superdotado’: http://img01.lavanguardia.com/2013/03/27/Nick-d-Aloisio-muestra-su-apli_54370729186_51348736062_224_270.jpg

⁵⁶ <https://www.youtube.com/user/thnkrtv>

⁵⁷ <http://hacaduda.com/>



Figura 1.20. Santiago González, programador prodigio de 'apps' con 14 años.



Figura 1.21. Web de descarga de 'apps' de Santiago González (se enmarcan sus tres favoritas)

Otro ejemplo de joven prodigio de la programación es el de Thomas Suarez, un joven desarrollador de ‘apps’ para iPhone de 12 años, que está ahora utilizando sus excepcionales habilidades de programación para ayudar, como mentor experto, a otros niños y niñas que desean aprender código informático (Suarez, 2011)

Pero no caigamos en el estereotipo de un programador siempre masculino. Existen igualmente ejemplos de chicas que están destacando en la programación de ‘apps’. Quizás los más reseñables son: el caso ya mencionado de Jessica Liu⁵⁸, mujer, 21 años, origen asiático y programadora; y el caso de Lyndsey Scott (29 años, mujer, y modelo de pasarela afroamericana), que rompe todos los tópicos y clichés sobre la figura del programador y/o el sujeto de alta capacidad aficionado a la informática (tradicionalmente varón y de origen caucásico). Lyndsey Scott afirma en una reciente entrevista: “Tenemos este prejuicio de que la gente experta en tecnología/informática debe ser y parecer de cierta manera (...) ese estereotipo es destructivo, y creo que está en la base de que haya tan pocos programadores de sexo femenino o provenientes de minorías” (Khorram, 2014, en línea). Retomaremos la cuestión de género más adelante en esta tesis doctoral.

Parece pues que programar ‘apps’ está de moda. Tendencia que llega imparable a nuestro país; uno de los indicadores de esta tendencia es la proliferación de concursos y certámenes que tratan de promover, detectar y premiar a los emergentes jóvenes talentos en la programación de aplicaciones móviles. Así, ya en junio de 2013 se celebraron las ‘I Olimpiadas Android’⁵⁹, organizadas por U-Tad, Centro Universitario de Tecnología y Arte Digital (Figura 1.22). Según recoge la nota de prensa (U-Tad, 2013), es:

“Un concurso de desarrollo y programación de aplicaciones móviles, en el que 40 equipos de participantes, de entre 16 y 22 años, competirán para demostrar su talento creando la aplicación Android más original y eficiente (...) el mundo de las aplicaciones móviles ha provocado un gran impacto en la sociedad digital (...) la cuota de mercado de aplicaciones móviles está en pleno auge en España, donde se descargan 1,4 millones al día, lo que supone más de 16 descargas al segundo. Su desarrollo y programación se ha convertido en una de las ramas del software más dinámicas y demandadas por el tejido empresarial (...) con esta iniciativa queremos acercar la ingeniería desde su perspectiva más digital a los jóvenes de nuestro país. Hablamos de uno de los sectores que cuenta con mayor demanda de profesionales especializados en programación de software, plataformas móviles, redes sociales, Apps o entornos colaborativos.” (U-Tad, 2013, en línea)

Más recientemente, en enero de 2014, se entregaron los ‘I Premios Apps de la Fundación Telefónica’⁶⁰, un concurso destinado a chicos y chicas de entre 18 y 25 años para crear aplicaciones móviles en Firefox OS, el sistema operativo de código abierto cuyo lanzamiento ha sido liderado por Telefónica I+D en España. Los 8 ganadores de dicho concurso, 7 chicos y 1 chica, forman parte del grupo de expertos consultado para la validación de contenido de nuestro Test de Pensamiento Computacional (TPC), tal y como se describirá en el apartado 6.3 de esta tesis doctoral.

⁵⁸ En su web <http://cakefordinner.com/> encontramos numerosas muestras de su talento como programadora de juegos y ‘apps’

⁵⁹ <http://www.olimpiadasandroid.com/ganadores.php>

⁶⁰ <http://www.fundaciontelefonica.com/premios-apps/>



Figura 1.22. Página Web de las ‘Olimpiadas Android’

Por otro lado, y ciñéndonos ahora al ámbito educativo, comienzan a proliferar webs que recopilan, clasifican, filtran y recomiendan de manera personalizada ‘apps’ educativas a profesores, padres y estudiantes: así, la web ‘Cappaces.com’⁶¹ especializada en ‘apps’ para niños con alguna discapacidad; o ‘Eduapps.es’⁶² (Díez, Santiago, & Climent, 2015), que según expresa en su web:

“Actualmente existen más de 80.000 aplicaciones educativas. Eduapps nace con la intención de recopilar y analizar las principales apps para el aula. Las apps no tienen valor por sí mismas sin la figura del maestro, que las selecciona e integra para convertir la actividad en el aula en una experiencia motivadora.” (Díez, Santiago, & Climent, 2015, en línea)

‘Eduapps.es’ filtra las aplicaciones en función de la etapa educativa, el área curricular e incluso por el nivel de complejidad cognitivo asociado al uso de la aplicación según la taxonomía de Bloom (1972). Las aplicaciones que incluye ‘Eduapps.es’ han sido valoradas por profesores de acuerdo a determinados criterios educativos, a partir de una rúbrica especialmente diseñada para tal propósito; y se ofrecen comentarios y valoraciones de los usuarios a las ‘apps’ propuestas.

Todo este movimiento, de aprendizaje móvil (‘m-learning’) con conexión a internet (‘i-mode’) comienza igualmente a ser recogido por autoridades reconocidas en el ámbito de la educación de la alta capacidad (Tourón, 2013c, 2013d); que defienden la integración del iPad u otros dispositivos móviles en el aula, no por pura moda, sino porque es una tecnología eficiente, y con ella: es posible recuperar (o hacer real) el aprendizaje al propio ritmo, la evaluación formativa, el feedback, la diversificación curricular, el adaptar el nivel de reto a la competencia de los escolares y tantos otros principios de una escuela que debe poner en primer plano al alumno y no al profesor, el aprendizaje y no la enseñanza. Además Tourón (2013c) nos indica que reciente investigación demuestra, mayoritariamente, que el uso del iPad y otras tabletas en la clase mejora el aprendizaje (Webb, 2011). En España, el reciente libro ‘Mobile learning: nuevas realidades en el aula’ (Santiago,

⁶¹ <http://cappaces.com/presentacion-de-cappaces/>

⁶² <http://www.eduapps.es/>

Trabaldo, Kamijo, & Fernández, 2015) explora y recopila las prácticas educativas más innovadoras de nuestro país en lo relativo a la integración de los dispositivos móviles en los centros educativos.

En este punto, para no perder nuestro hilo discursivo central, es preciso recordar que en nuestra aproximación inicial al problema de investigación nos planteamos no sólo (o no principalmente) el consumo de ‘apps’ en el aula; sino precisamente, y yendo a un nivel superior, la creación y programación de las mismas, algo que puede ser especialmente posible y deseable para los alumnos de alta capacidad. Y es que en la sociedad 2.0, y la 3.0 que ya asoma, se espera de los ciudadanos en su máximo nivel de competencia digital, no sólo el consumo de aplicaciones sino, sobre todo, su creación y compartición (Resnick *et al.*, 2009).

Así empieza a debatirse abiertamente en el foro público sobre la conveniencia de que nuestros estudiantes aprendan a programar como se aprende a leer (Silió, 2013), con el objetivo de formar a creadores de aplicaciones, y no sólo a meros consumidores.; reconociéndose progresivamente cómo este aprendizaje potencia la creatividad y la mente lógica. Y es que un número creciente de países enseña a los alumnos a escribir código: en el recién publicado informe ‘*Computing our future. Computer programming and coding: priorities, school curricula and initiatives across Europe*’ (European Schoolnet, 2015) ya se informa de 15 países de la Unión Europea que han integrado, en mayor o menor medida, la programación informática (o, coloquialmente, ‘coding’) en sus currículos educativos⁶³ (Figura 1.23)

Recentrando el problema en los alumnos de alta capacidad, y para cerrar sintéticamente la fundamentación de la hipótesis directiva de **relevancia**, ponemos ahora en relación las ‘apps’ como productos de aprendizaje y los alumnos de alta capacidad como potenciales sujetos del mismo. Observamos una clara convergencia entre los parámetros indicados a la hora de exigir y valorar productos de aprendizaje a los alumnos de alta capacidad (Jiménez Fernández, 2010) y las características propias de las ‘apps’. Se condensa en la siguiente Tabla 1.4.

Tabla 1.4. Convergencia entre los parámetros de valoración de los productos de aprendizaje de alumnos de alta capacidad, y las características propias de las ‘apps’

<i>Los productos de aprendizaje de los alumnos de alta capacidad...</i>	<i>Las ‘apps’...</i>
<i>...deben estar relacionados con problemas reales y significativos para ellos...</i>	<i>...versan potencialmente sobre cualquier disciplina; en relación con el mundo real y los intereses del programador...</i>
<i>...deben estar dirigidos a un público real...</i>	<i>...se ponen a disposición de un público de escala mundial a través de tiendas on-line como ‘App Store’ o ‘Google Play’...</i>
<i>...deben ser valorados y evaluados por las respectivas audiencias...</i>	<i>...son comentadas, puntuadas, valoradas por los usuarios. Este feed-back sirve tanto al programador como a otros usuarios...</i>
<i>...deben ser productos originales o transformaciones de otros (Jiménez Fernández, 2010)</i>	<i>...son programas originales o código transformado a partir de programas de otros.</i>

⁶³ Retomaremos en profundidad este asunto en el Capítulo 2 sobre ‘Codigoalfabetización’, que, como adelantábamos en la introducción, amplía el foco de la enseñanza-aprendizaje de los lenguajes informáticos de programación hacia una nueva práctica de lectoescritura, un nuevo alfabetismo, extensible a la población escolar general (no sólo a los alumnos de alta capacidad) y a todo tipo de objetos digitales (no sólo ‘apps’)

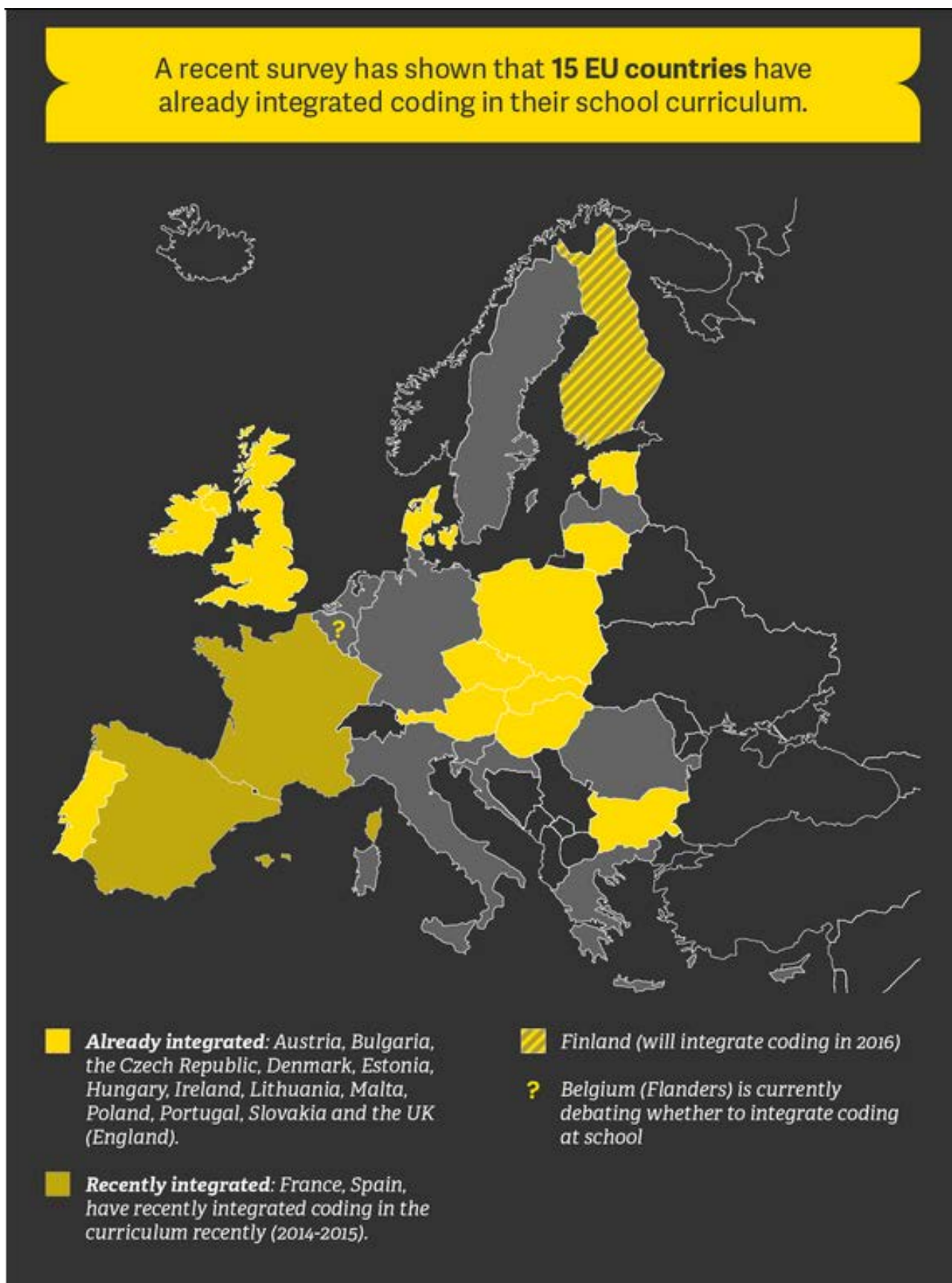


Figura 1.23. Países de la UE que ya integran, en alguna medida, la programación informática ('coding') en sus currículos (European Schoolnet, 2015). Infografía completa disponible en http://www.euractiv.com/sites/default/files/infographic_coding_at_school.png

Así pues, y siguiendo la ruta argumental que nos hemos fijado anteriormente (Figura 1.1), parece que existe convergencia entre las características del *producto* de aprendizaje que nos proponemos (las ‘apps’) con arreglo a los parámetros exigibles a los alumnos de alta capacidad; y el conjunto de necesidades, estándares y valores sociales imperantes en la actual Sociedad de la Información y el Conocimiento. Encontramos pues apoyo para enunciar de manera fundamentada la hipótesis directiva de *relevancia*.

1.4. Enunciación inicial de hipótesis directivas

Nos planteábamos al comienzo de este capítulo la siguiente pregunta de investigación, primera tentativa de esta tesis doctoral: ¿es posible y deseable aprender a programar aplicaciones (‘apps’) para dispositivos móviles (*smartphones* y *tablets*) como enriquecimiento curricular indicado especialmente para alumnos y alumnas de alta capacidad? (Figura 1.1)

Tras la exploración realizada en los anteriores apartados, parece quedar fundamentada la siguiente triple hipótesis directiva, que funcionará como guía o brújula inicial de esta tesis doctoral:

- i. **Hipótesis de Adecuación (H_A):** Aprender a programar ‘apps’ es *adecuado* para los sujetos de alta capacidad dado que existe correspondencia entre las demandas cognitivas de la tarea y las características cognitivas, creativas y de personalidad de los más capaces, en especial, su alta habilidad para el manejo de lenguajes abstractos y formales.

En una formulación alternativa, aprender a programar ‘apps’ es *adecuado* dado que existe convergencia entre las características del *contenido* de aprendizaje que nos proponemos (‘programación de apps’) y las características del aprendiz destinatario, en este caso los alumnos de alta capacidad.

- ii. **Hipótesis de Viabilidad (H_V):** Aprender a programar ‘apps’ es *viable* y económico para los sujetos de alta capacidad dado que para ello se pueden establecer fácilmente metodologías didácticas a distancia y soportadas tecnológicamente, a través de cursos en línea y en abierto con tutorización esporádica, que se adaptan perfectamente al estilo de aprendizaje de estos sujetos.

En una formulación alternativa, aprender a programar ‘apps’ es *viable* y económico para los alumnos de alta capacidad, dado que existe convergencia entre las características del *proceso* de aprendizaje que nos proponemos (‘programación de apps’) con arreglo al estilo de los alumnos de alta capacidad; y los medios, métodos y tecnologías actualmente disponibles.

- iii. **Hipótesis de Relevancia (H_R):** Aprender a programar ‘apps’ es una habilidad emergente y *relevante* para los estudiantes actuales en general y para los más capaces en particular, dado que les permite generar un producto final, la aplicación, de creciente uso, utilidad y valoración social, y que potencialmente puede ser puesto a disposición de inmensas audiencias reales y globales.

En una formulación alternativa, aprender a programar ‘apps’ es una habilidad emergente y *relevante*, para los estudiantes actuales en general y para los más capaces en particular, dado que existe convergencia entre las características del *producto* de aprendizaje que nos

proponemos (las ‘apps’) con arreglo a los parámetros exigibles a los alumnos de alta capacidad; y el conjunto de necesidades, estándares y valores sociales imperantes en la actual Sociedad de la Información y el Conocimiento.

Adicionalmente, la utilización para abordar dicho aprendizaje de nuevos recursos ‘on-line’, como Khan Academy, integrados en el aula ordinaria a través de una metodología de ‘*flipped classroom*’ (‘clase invertida’), supondría una transformación de una cuarta dimensión transversal a las mencionadas: el ambiente de aprendizaje, que se torna extraordinariamente flexible, satisfaciendo el estilo de los alumnos de alta capacidad. Todos estos aspectos se sintetizan y diagraman conceptualmente en la Figura 1.24.

Por tanto, hemos procedido a la enunciación fundamentada inicial de las hipótesis directivas de ***adecuación, viabilidad y relevancia*** de aprender a programar ‘apps’ en sujetos de alta capacidad. El contraste positivo de dichas hipótesis supondría encontrar entonces una auténtica ‘adaptación curricular significativa’ para los alumnos de alta capacidad, dado que enriquecería su currículum en todas las dimensiones posibles (Jiménez Fernández, 2010): en el ***contenido*** (tanto en extensión como en profundidad, aumentando su complejidad, abstracción y variedad); en el ***proceso*** (haciendo hincapié en el trabajo autónomo, el ritmo personalizado, en la resolución creativa de problemas, y en la expresión convergente-divergente del contenido aprendido); y en el ***producto*** (objeto final original que se basa en un problema real, y se orienta hacia una audiencia real que puede disfrutar y evaluar abiertamente la ‘app’ generada).

Ello permitiría superar reduccionismos propios de los programas de enriquecimiento parciales que se limitan sólo a una de las dimensiones mencionadas; pudiendo constituirse como base para un ‘Modelo de Currículo Integrado’ para los más capaces en el área de Informática/Computación, al estilo de los recientemente propuestos por VanTassel-Baska (2015) en las áreas de Ciencias, Lengua, Matemáticas y Estudios Sociales. Se acomete dicho contraste a través de la revisión sistemática que se describe en el siguiente apartado.

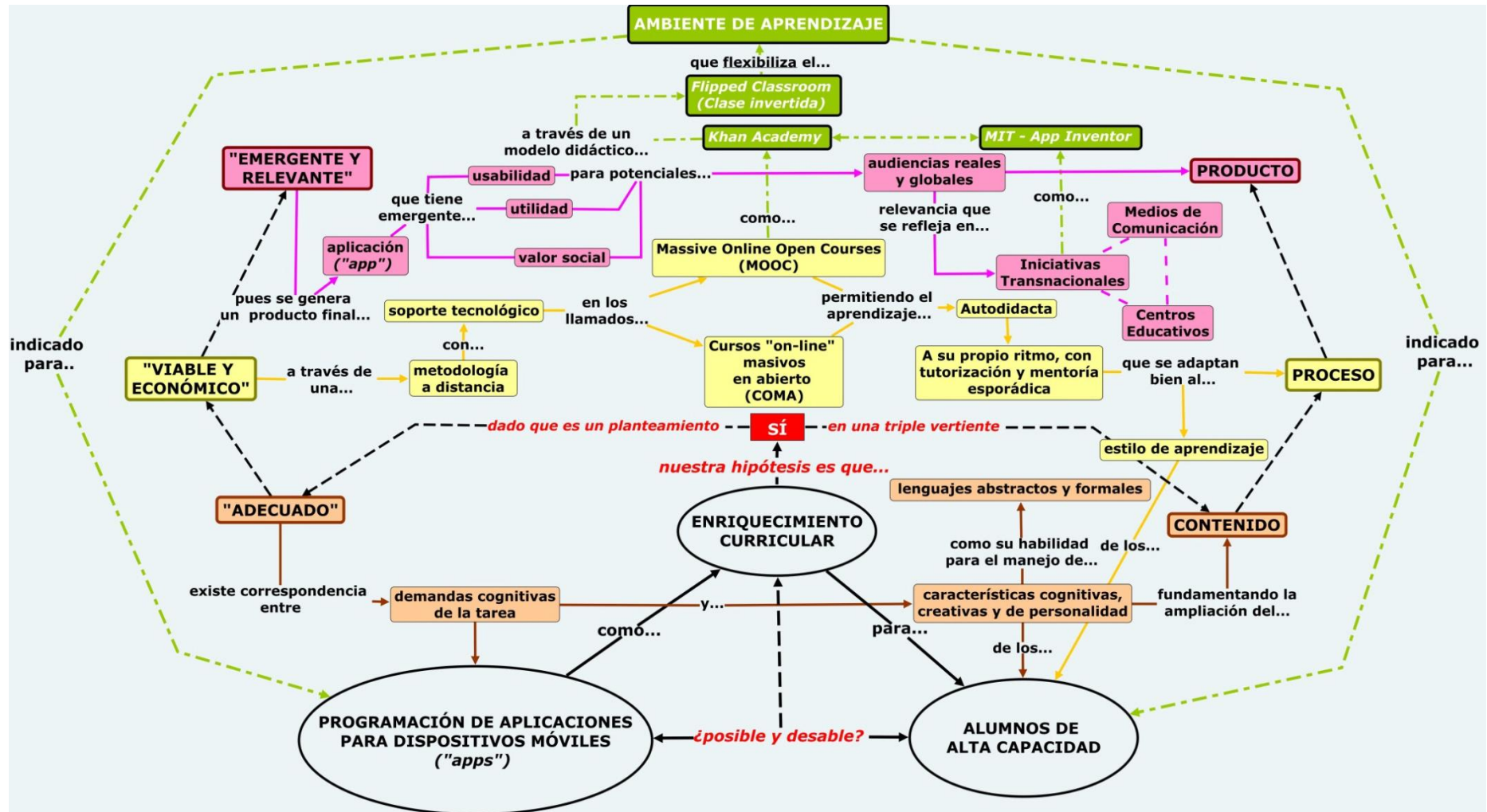


Figura 1.24. Mapa conceptual sobre la adecuación, viabilidad y relevancia de la programación de 'apps' como enriquecimiento curricular en alumnado de alta capacidad

1.5. Revisión sistemática

Una vez delimitado, en primera tentativa, nuestro problema de investigación, y enunciadas inicialmente nuestras hipótesis directivas de manera fundamentada, nos proponemos realizar una revisión sistemática de fuentes cuyos objetivos se explicitan a continuación:

- **Objetivo General:** Encontrar publicaciones que contrasten la posibilidad y deseabilidad de aprender a programar aplicaciones para dispositivos móviles (*apps*) como enriquecimiento curricular indicado especialmente para alumnado de alta capacidad.
 - **Objetivo Específico 1:** Encontrar publicaciones que sometan a contraste la Hipótesis de Adecuación (H_A)
 - **Objetivo Específico 2:** Encontrar publicaciones que sometan a contraste la Hipótesis de Viabilidad (H_V)
 - **Objetivo Específico 3:** Encontrar publicaciones que sometan a contraste la Hipótesis de Relevancia (H_R)

1.5.1. Proceso de fijación de los parámetros de búsqueda

Para acometer estos objetivos comenzamos un proceso exploratorio en la base de datos especializada en Educación, ERIC (Ebsco), que nos permite fijar finalmente unos parámetros de búsqueda óptimos para nuestra revisión sistemática. El conjunto del proceso, que se diagrama en la Figura 1.25, fue realizado inicialmente en el mes de julio de 2013, y replicado en abril de 2014.

Así, comenzamos la exploración en ERIC (Ebsco) con el siguiente Booleano/Frase de búsqueda: *AB gift* AND AB app**, publicaciones en cuyo *abstract* se incluya la raíz “*gift*” (que remite al conjunto de la alta capacidad) y la raíz “*app*” (que remite en principio a las aplicaciones informáticas); encontrando la elevada cifra de 4.253 resultados (Figura 1.25), número inmanejable para nuestra revisión. Ante tal cantidad de resultados, hacemos una nueva simulación de búsqueda incluyendo los siguientes dos limitadores: “*Texto completo disponible*” y “*Fecha de publicación desde 2008 hasta 2013*” (ambos inclusive); obteniendo 220 resultados (Figura 1.25). Al revisar los *abstract* de los primeros cinco artículos, ordenados por ‘relevancia’, nos percatamos de que el término de búsqueda “*app**” es excesivamente genérico y no discriminatorio; pues devuelve resultados como “*application*”, “*approach*”, “*appendix*”, etc... que no tienen que ver con nuestros objetivos de búsqueda.

Buscamos entonces en el Tesoro de ERIC algún término que nos sirva para acotar nuestra búsqueda y esté reconocido en el campo de estudio. Encontramos como aparentemente más adecuados: “*Computer*” (ordenador), pues sirve de término raíz para múltiples descriptores que entran dentro de nuestros objetivos de búsqueda como “*Computer Assisted Instruction*”, “*Computer Oriented Programs*”, “*Computer Literacy*”, etc.; y “*Programming*” (programación), descriptor que recomienda el Tesoro en vez de “*Computer Languages*”, y que dirige aún mejor nuestra búsqueda hacia publicaciones en donde se hable sobre programación con lenguajes informáticos.

Así, con esta intención de acotar nuestra búsqueda anterior hacemos las siguientes simulaciones de búsquedas más específicas:

- a. Booleano/Frase de búsqueda: *AB gift* AND AB app* AND SU⁶⁴ computer*, manteniendo los limitadores anteriores; obteniendo un escaso número de 5 resultados (Figura 1.25)
- b. Booleano/Frase de búsqueda: *AB gift* AND AB app* AND SU programming*, eliminando el limitador de fecha; y obteniendo escasamente 1 único resultado (Figura 1.25), además muy antiguo (1997).

Ampliando entonces la búsqueda al término raíz “*program**”, con el Booleano/Frase de búsqueda *AB gift* AND AB app* AND SU program**, desactivando el limitador de “*Texto completo disponible*”⁶⁵, obtenemos una interesante cifra de 101 resultados (Figura 1.25); a priori manejable para nuestra revisión sistemática. Sin embargo, al revisar los *abstract* de los primeros cinco artículos, ordenados por ‘relevancia’, nos percatamos de que el término raíz de búsqueda “*program**” es excesivamente genérico y no discriminatorio, pues devuelve resultados como “*summer programs*”, “*programs evaluation*”, etc... que no tienen que ver con nuestros objetivos de búsqueda.

En este punto intuimos que la combinación óptima de búsqueda es:

- Excluir el término de búsqueda “*app**” por no ser discriminatorio en ningún caso.
- Simultanear en nuestra búsqueda (además del término “*gift**”) los términos “*computer*” y “*program**”; este último sí es discriminatorio en combinación con el anterior.
- Desdoblar el término “*computer*” en (“*comput**” OR “*tablet**”); para incluir publicaciones bien que traten del uso del ordenador (necesario para la programación de la ‘*app*’), bien que traten del uso de la tableta (dispositivo móvil necesario para la ejecución de la ‘*app*’).
- El booleano de búsqueda resultante es una triple conjunción, a priori muy restrictiva; por lo que se decide ampliar el campo desde *AB* (*Abstract*) a *TX* (*All Text*), quedando como booleano resultante final: *TX gift* AND TX program* AND TX (comput* OR tablet*)*
- Decidimos eliminar definitivamente el limitador de “*Texto completo disponible*” e incluir todas las publicaciones desde el año 2000.

Encontramos entonces 77 resultados (Figura 1.25) de contenido relacionado con nuestros objetivos de búsqueda, siendo un número de publicaciones asequible para la revisión sistemática que nos proponemos.

⁶⁴ SU es el código relativo a “*Descriptors*” (Descriptores)

⁶⁵ Se entiende que, aunque el texto completo no esté disponible en esa base de datos, una vez localizada la referencia podrá conseguirse por otros medios alternativos.

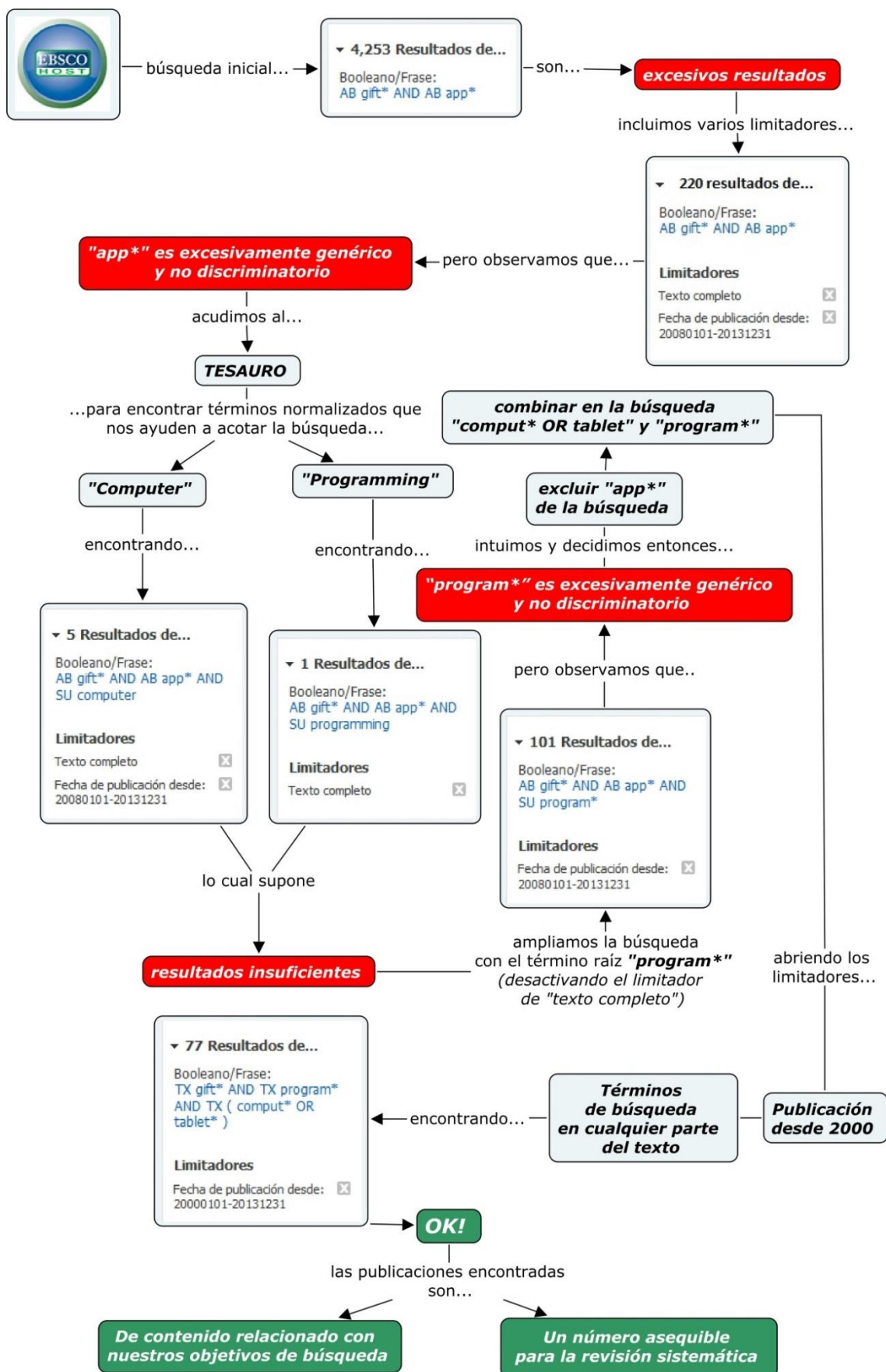


Figura 1.25. Proceso de fijación de los parámetros de búsqueda

Llegados a estos parámetros de búsqueda, se replica la misma en las siguientes bases de datos: ERIC (ProQuest), especializada también en Educación, que arroja 83 resultados; PsycINFO, especializada en Psicología, que devuelve 59 resultados; EconLit, especializada en Economía, que devuelve 17 resultados; y la ACM Digital Library, que arroja 592 resultados. Se ilustra comparativamente en la siguiente Tabla 1.5.

Tabla 1.5. Comparativa de resultados de búsqueda por bases de datos

<i>Base de datos</i>	<i>Booleano de búsqueda</i>	<i>Campo de búsqueda</i>	<i>Limitadores</i>	<i>Resultados de búsqueda</i>
<i>ERIC (Ebsco)</i>				77
<i>ERIC (ProQuest)</i>				83
<i>PsycINFO</i>	<i>gift* AND program* AND (comput* OR tablet*)</i>	<i>All Text</i>	<i>Publicación 2000-2013</i>	59
<i>EconLit</i>				17
<i>ACM Digital Library</i>				592

1.5.2. Criterios de inclusión y de exclusión

A partir de la tentativa anterior, en orden a realizar una revisión sistemática *eficaz* (que cumpla con nuestros objetivos de búsqueda) y *eficiente* (con un número asequible de elementos a analizar); optamos por los siguientes criterios de inclusión:

- Publicaciones catalogadas en las bases de datos: ERIC (Ebsco), ERIC (ProQuest), PsycINFO, y EconLit.
- Que respondan al booleano de búsqueda: *TX gift* AND TX program* AND TX (comput* OR tablet*)*, es decir, publicaciones en las cuales a lo largo de su texto aparezca la raíz “*gift**” (que remite al campo de la alta capacidad); y la raíz “*program**” (que remite a la programación de lenguajes informáticos); y, o bien la raíz “*comput**” (que remite a los ordenadores, necesarios para la programación de la ‘*app*’), o bien la raíz “*tablet**” (que remite a los dispositivos móviles necesarios para la ejecución y disfrute de la ‘*app*’).
- Que hayan sido publicadas desde el año 2000 hasta el 2013, ambos incluidos.
- Que hayan sido publicadas en lengua inglesa o española.
- Que respondan a un tipo de publicación ‘*Journal Article*’, es decir, ‘Artículo de Revista Científica’.

Serán criterios de exclusión:

- Publicaciones catalogadas en otras bases de datos distintas a las indicadas más arriba; por ejemplo, se excluyen las de la base ACM Digital Library, especializada en informática, pues el booleano de búsqueda utilizado contiene múltiples términos centrales en esa área de estudio, devolviendo un número inmanejable de resultados (592 resultados). Sería necesario

investigar sobre un booleano de búsqueda específico para esta base, algo que excede a esta revisión sistemática en este momento⁶⁶.

- En el caso de las publicaciones catalogadas exclusivamente en la base de datos EconLit, especializada en Economía, y dado que es algo tangencial a nuestros objetivos de búsqueda; se excluyen aquellas publicaciones en cuyo ‘Título’ no haya indicio alguno de relación con nuestro problema de investigación.
- Que no respondan al booleano de búsqueda indicado: *TX gift* AND TX program* AND TX (comput* OR tablet*)*.
- Que hayan sido publicadas fuera del periodo con anterioridad al año 2000.
- Publicaciones en lenguas distintas al inglés o al español.
- Tipos de publicaciones distintos a ‘*Journal Article*’; por ejemplo, informes, tesis, disertaciones, actas de congresos, etc... quedan excluidas de nuestra revisión.

1.5.3. Procedimiento de tabulación

Recopilamos en el gestor bibliográfico RefWorks todas las publicaciones que cumplan con los criterios fijados anteriormente, eliminando duplicados. De cara a su posterior análisis y presentación de resultados, cada referencia es tabulada en una hoja de cálculo⁶⁷ (Figura 1.26, izquierda) en las siguientes variables: ‘¿Publicación arbitrada?’ (Sí / no); ‘Naturaleza’ (Investigación, Experiencia, Ensayo...); ‘Autor/es’; ‘Título’; ‘Revista de publicación’; ‘Año de publicación’; ‘Idioma’; ‘Nivel educativo sobre el que trata’; ‘Base/s de dato/s en las que se encuentra’. Y ‘¿Adecuación?’, ‘¿Viabilidad?’ y ‘¿Relevancia?’ (Sí / No), en función de si se encuentran o no en el respectivo *abstract* evidencias de contraste de las distintas hipótesis.



Figura 1.26. Código QR de enlace a la hoja de cálculo en donde se han tabulado las referencias de la revisión sistemática (izquierda); y código QR de enlace al documento con la recopilación de *abstracts* (derecha)

A través del código QR de la Figura 1.26 (derecha), el lector puede acceder a un documento⁶⁸ en el que se reproduce el *Resumen/Abstract* (y los *Descriptors*) de todas las publicaciones recopiladas y analizadas en la revisión sistemática, por orden alfabético de autor; y sobre el mismo se marcan, según un código de colores, los indicios que se encuentran para el contraste de las respectivas hipótesis directivas enunciadas.

⁶⁶ Sin embargo, como podrá comprobarse más adelante en los capítulos sobre ‘Codigoalfabetización’ y ‘Pensamiento Computacional’, se citarán en otros momentos del desarrollo teórico de esta tesis, múltiples publicaciones extraídas de la ACM Digital Library.

⁶⁷ Disponible en <https://db.tt/e8OdYpcJ>

⁶⁸ Disponible en <https://db.tt/K9pmjMXK>

1.5.4. Resultados

Atendiendo a los criterios de inclusión y de exclusión fijados, obtenemos los siguientes resultados: **67 artículos** académicos publicados en revistas científicas en el periodo 2000-2013, en lengua inglesa o española; y catalogados en alguna de las bases de datos incluidas en la revisión. De ellos, 60 artículos (89,6%) corresponden a revistas arbitradas (habitualmente por el sistema de ‘peer review’ o ‘doble ciego’) y 7 (10,4%) a revistas no arbitradas. Con respecto a la naturaleza u orientación de los 67 artículos obtenidos, se representa en la siguiente Tabla 1.6.

Tabla 1.6. Naturaleza u orientación de los artículos revisados

	Frecuencia	Porcentaje	Porcentaje acumulado
Research (Investigación empírica)	29	43,3	43,3
Descriptive (Descripción de una experiencia)	24	35,8	79,1
Evaluative (Investigación evaluativa)	8	11,9	91,0
Opinion Paper (Ensayo de opinión)	4	6,0	97,0
Book Review (Recensión de libro)	1	1,5	98,5
Special Issue (Número especial: monográfico)	1	1,5	100,0
TOTAL	67	100,0	

Con respecto a la autoría de los artículos, se detectan algunos nombres recurrentes de investigadores. En concreto, destacan Del Siegle⁶⁹, profesor de la Universidad de Connecticut especialista en alta capacidad, con 7 artículos (Siegle, 2003, 2004, 2005, 2009, 2011, 2013; Siegle & Powell, 2004); Paula Olszewski-Kubilius⁷⁰, directora del Northwestern University’s Center for Talent Development (CTD), con 3 artículos (Olszewski-Kubilius, 2001; Olszewski-Kubilius & Lee, 2004, 2005); y James Gentry⁷¹, formador de formadores en la Tarleton State University, con 2 artículos (Gentry, 2008; Gentry, Fowler & Nichols, 2007). El resto de autores recogidos en la revisión sistemática sólo aportan un artículo.

Con respecto a las revistas científicas en las cuales han sido publicados los artículos, se representan en la Figura 1.27 de mayor a menor número de referencias (en la categoría ‘Otros’ se agrupan las revistas que sólo aportan un artículo a la revisión). Se detallan todas las revistas en la Tabla 1.7.

De los 67 artículos revisados, en 49 (73,1%) se tuvo acceso ‘on-line’ al texto completo del mismo; en los 18 restantes (26,9%) sólo se pudo acceder al correspondiente *abstract*. Y sólo 1 artículo fue escrito en lengua española (Hernández & Borges, 2005); los 66 restantes están publicados en lengua inglesa.

Con respecto al año de publicación de los artículos, se representa en la Figura 1.28.

⁶⁹ <http://delsiegle.education.uconn.edu/>

⁷⁰ <http://www.ctd.northwestern.edu/team-bios/paula-olszewski-kubilius>

⁷¹ <http://www.learningwithjamesgentry.com/>

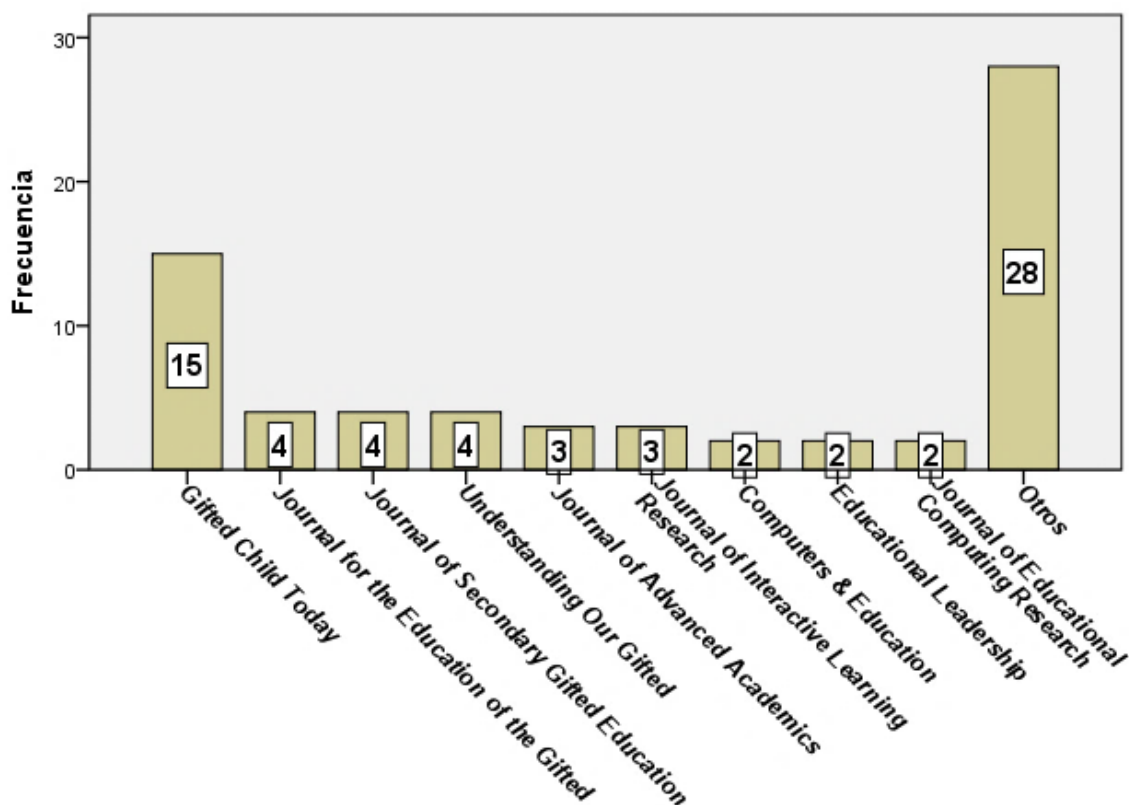


Figura 1.27. Revistas de publicación de los artículos revisados

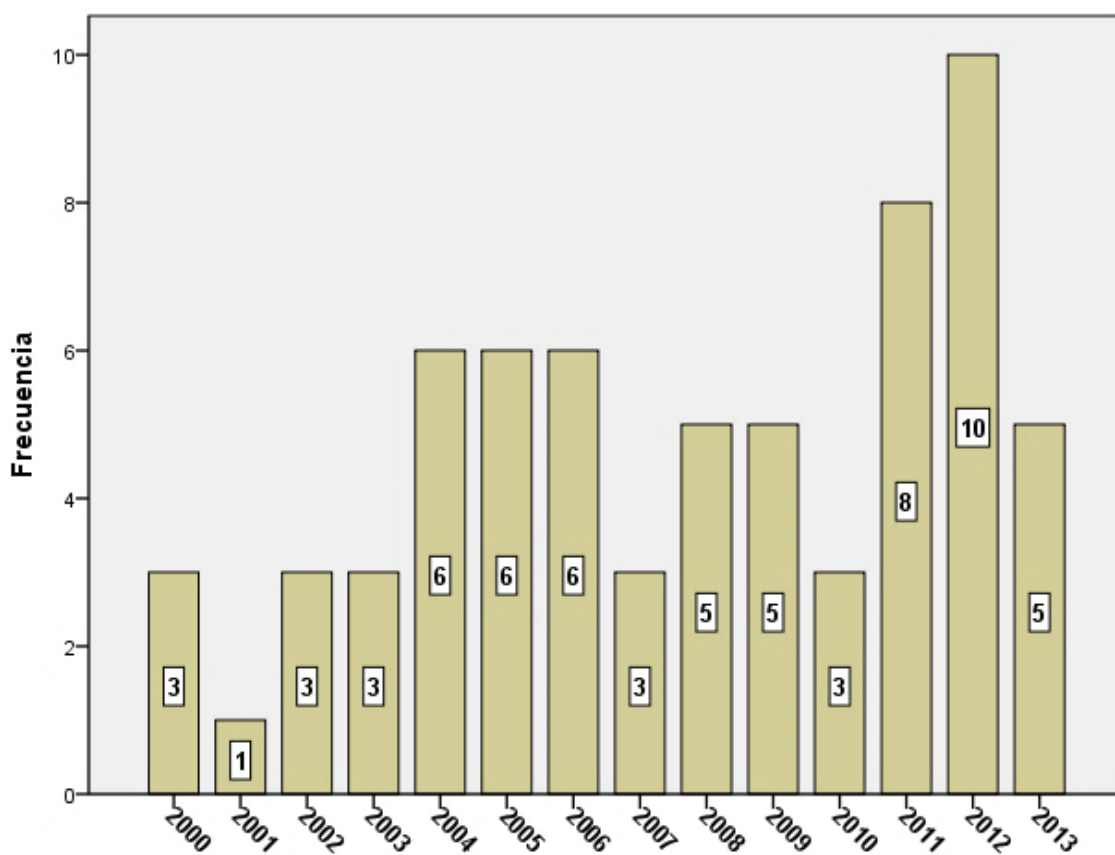


Figura 1.28. Año de publicación de los artículos revisados

Tabla 1.7. Revistas de publicación de los artículos revisados

	Frecuencia	Porcentaje	Porcentaje acumulado
Gifted Child Today	15	22,4	22,4
Journal for the Education of the Gifted	4	6,0	28,4
Journal of Secondary Gifted Education	4	6,0	34,3
Understanding Our Gifted	4	6,0	40,3
Journal of Advanced Academics	3	4,5	44,8
Journal of Interactive Learning Research	3	4,5	49,3
Computers & Education	2	3,0	52,2
Educational Leadership	2	3,0	55,2
Journal of Educational Computing Research	2	3,0	58,2
American Journal of Distance Education	1	1,5	59,7
British Journal of Educational Technology	1	1,5	61,2
British Journal of Music Education	1	1,5	62,7
Campus-Wide Information Systems	1	1,5	64,2
Computers in Libraries	1	1,5	65,7
Connection Science	1	1,5	67,2
CyberPsychology & Behavior	1	1,5	68,7
District Administration	1	1,5	70,1
Educational Assessment	1	1,5	71,6
Electronic Journal of Research in Educational Psychology	1	1,5	73,1
General Music Today	1	1,5	74,6
Gifted Child Quarterly	1	1,5	76,1
IEEE Transactions on Education	1	1,5	77,6
Information Technology in Childhood Education Annual	1	1,5	79,1
Instructional Science	1	1,5	80,6
International Journal of Science Education	1	1,5	82,1
Journal of Asynchronous Learning Networks	1	1,5	83,6
Journal of Behavioral Decision Making	1	1,5	85,1
Journal of Comparative Economics	1	1,5	86,6
Journal of Management & Organization	1	1,5	88,1
Journal of Science Education and Technology	1	1,5	89,6
Learning & Leading with Technology	1	1,5	91,0
Merrill-Palmer Quarterly	1	1,5	92,5
Neural Networks	1	1,5	94,0
Roeper Review: A Journal on Gifted Education	1	1,5	95,5
the Behavior Therapist	1	1,5	97,0
The Journal of Creative Behavior	1	1,5	98,5
Turkish Online Journal of Distance Education	1	1,5	100,0
TOTAL	67	100,0	

Con respecto a las bases de datos en las que se catalogan los artículos, en la Tabla 1.8 se presenta la contingencia de aparición de los mismos en las distintas bases consultadas:

Tabla 1.8. Contingencia de aparición de los artículos en las distintas bases de datos

BASE DE DATOS	...en exclusiva	...también en ERIC (Ebsco)	...también en ERIC (ProQuest)	...también en PsycINFO	...también en EconLit	TOTAL
ERIC (Ebsco)	0 (0%)	-	43 (100%)	6 (14%)	0 (0%)	43 (100%)
ERIC (ProQuest)	2 (4%)	43 (96%)	-	6 (13%)	0 (0%)	45 (100%)
PsycINFO	21 (78%)	6 (22%)	6 (22%)	-	0 (0%)	27 (100%)
EconLit	1 (100%)	0 (0%)	0 (0%)	0 (0%)	-	1 (100%)

Con respecto al nivel educativo sobre el que tratan los artículos revisados, se detalla en la Tabla 1.9 y se ilustra en la Figura 1.29.

Tabla 1.9. Nivel educativo sobre el que se centran los artículos

	Frecuencia	Porcentaje	Porcentaje acumulado
Secondary Education (Educación Secundaria y Bachillerato)	25	37,3	37,3
N/A⁷²	22	32,8	70,1
Elementary Education (Educación Primaria)	13	19,4	89,6
Higher Education (Educación Superior y Universitaria)	6	9,0	98,5
Preschool Education (Educación Infantil)	1	1,5	100,0
TOTAL	67	100,0	

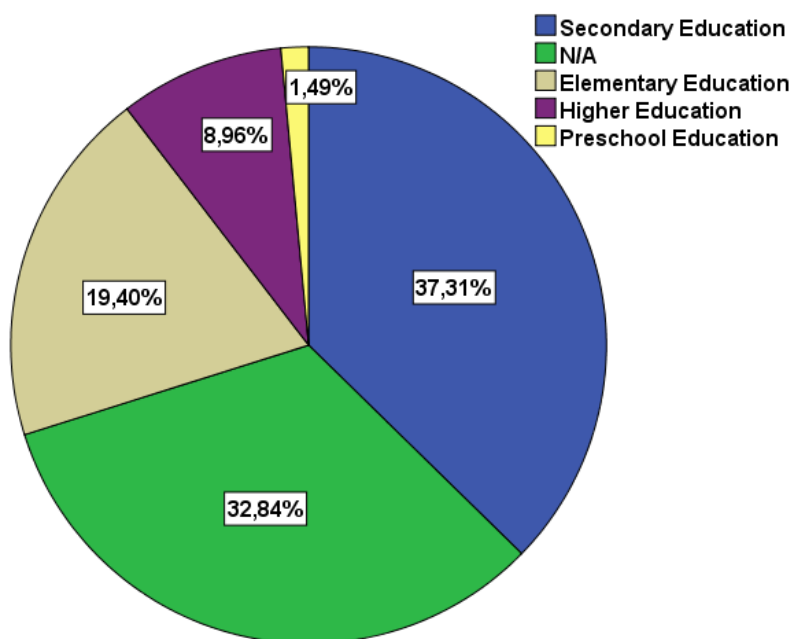


Figura 1.29. Nivel educativo sobre el que se centran los artículos

⁷² N/A = Not available (información no disponible o sin especificar)

Finalmente, con respecto a si se encuentran o no en el *abstract* de los respectivos artículos, indicios que permitan contrastar las hipótesis directivas planteadas (Figura 1.30):

- En 30 de los 67 *abstract* (44,8%) se encontraron indicios para contrastar la Hipótesis de Adecuación (H_A)
- En 42 de los 67 *abstract* (62,7%) se encontraron indicios para contrastar la Hipótesis de Viabilidad (H_V)
- En 13 de los 67 *abstract* (19,4%) se encontraron indicios para contrastar la Hipótesis de Relevancia (H_R)
- Sólo en 6 de los 67 *abstract* (8,9%) no se encontró indicio alguno.

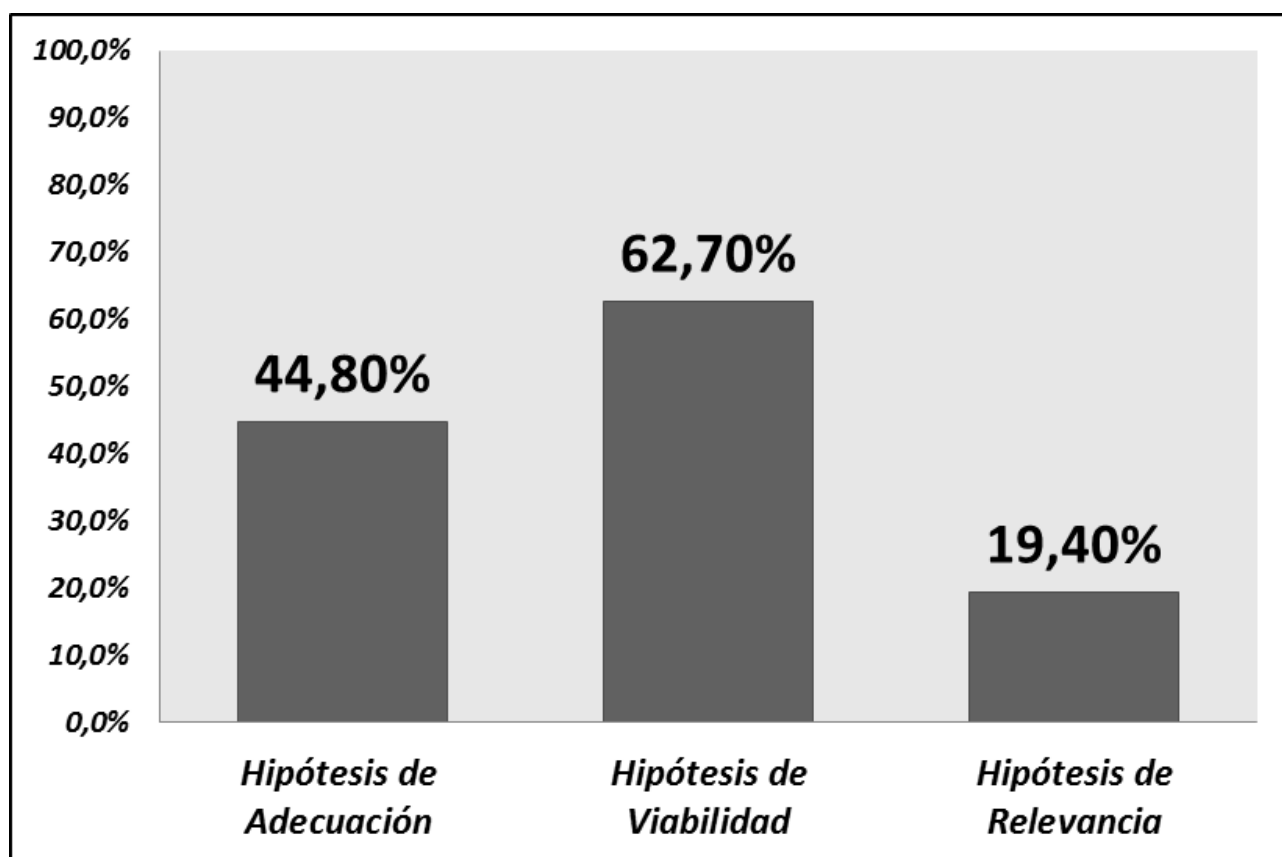


Figura 1.30. Porcentaje de artículos revisados en cuyo *abstract* se encuentran indicios para el contraste de las respectivas hipótesis directivas enunciadas

Con más profundidad, en la siguiente Tabla 1.10 se detallan para las 67 referencias revisadas, los indicios encontrados en cada una de ellas para el contraste de las distintas hipótesis directivas enunciadas.

Tabla 1.10. Indicios de contraste, encontrados en los *abstract* de las 67 referencias revisadas

INDICIOS ENCONTRADOS EN EL ABSTRACT...	Nº DE ARTÍCULOS	REFERENCIAS
<i>...ningún indicio...</i>	6 (8,9%)	<i>Gombachika & Khangamwa (2013)</i> ; <i>Gould et al. (2012)</i> ; <i>Huang et al. (2011)</i> ; <i>Nikolova & Taylor (2003)</i> ; <i>Plummer et al. (2011)</i> ; <i>Prince et al. (2012)</i>
<i>...sólo para la Hipótesis de Adecuación...</i>	13 (19,4%)	<i>Coxon (2012)</i> ; <i>Hoover & Stanley (2009)</i> ; <i>Kim (2011)</i> ; <i>Kim et al. (2013)</i> ; <i>Mann (2006)</i> ; <i>O'Brien et al. (2005)</i> ; <i>Olszewski-Kubilius & Lee (2005)</i> ; <i>Quartararo (2002)</i> ; <i>Salaman (2008)</i> ; <i>Siegle (2005)</i> ; <i>Siegle & Powell (2004)</i> ; <i>Sullivan & Lin (2012)</i> ; <i>Zhuang & Morgera (2007)</i>
<i>...sólo para la Hipótesis de Viabilidad...</i>	22 (32,8%)	<i>Adams & Cross (2000)</i> ; <i>Besnoy et al. (2012)</i> ; <i>Blair (2011)</i> ; <i>Bohemia & Ghassan (2012)</i> ; <i>Bullard (2005)</i> ; <i>Cope & Suppes (2002)</i> ; <i>De Wet (2006)</i> ; <i>Eristi (2012)</i> ; <i>Grimes & Warschauer (2008)</i> ; <i>Hernández & Borges (2005)</i> ; <i>Koivunen (2009)</i> ; <i>Larson & Murray (2008)</i> ; <i>McLester (2012)</i> ; <i>Olszewski-Kubilius & Lee (2004)</i> ; <i>Pedersen & Liu (2002)</i> ; <i>Shaklee & Landrum (2000)</i> ; <i>Siegle (2003)</i> ; <i>St. Cyr (2004)</i> ; <i>Thomson (2010)</i> ; <i>Weber & Cavanaugh (2006)</i> ; <i>Yang et al. (2011)</i> ; <i>Zucker (2009)</i>
<i>...sólo para la Hipótesis de Relevancia...</i>	4 (6,0%)	<i>Bitzer et al. (2007)</i> ; <i>Gentry (2008)</i> ; <i>Shivers (2012)</i> ; <i>Somyürek & Coskun (2013)</i>
<i>...para la Hipótesis de Adecuación + Hipótesis de Viabilidad...</i>	13 (19,4%)	<i>Burns (2006)</i> ; <i>Chan et al. (2010)</i> ; <i>Clark (2005)</i> ; <i>Dove & Zitkovich (2003)</i> ; <i>Gadanidis et al. (2011)</i> ; <i>Gilbert-Macmillan (2000)</i> ; <i>Lee (2011)</i> ; <i>Olszewski-Kubilius (2001)</i> ; <i>Rotigel & Fello (2004)</i> ; <i>Siegle (2009)</i> ; <i>Suppes et al. (2013)</i> ; <i>Wilson et al. (2012)</i> ; <i>Ysseldyke et al. (2004)</i>
<i>...para la Hipótesis de Adecuación + Hipótesis de Relevancia...</i>	2 (3,0%)	<i>Connolly (2010)</i> ; <i>Siegle (2004)</i>
<i>...para la Hipótesis de Viabilidad + Hipótesis de Relevancia...</i>	5 (7,5%)	<i>Eckstein (2009)</i> ; <i>Gentry et al. (2007)</i> ; <i>Siegle (2013)</i> ; <i>Sun et al. (2006)</i> ; <i>Wurst et al. (2008)</i>
<i>...para las 3 Hipótesis</i>	2 (3,0%)	<i>Reid & Roberts (2006)</i> ; <i>Siegle (2011)</i>
TOTAL	67 (100%)	

1.5.5. Discusión

Al comienzo de este capítulo enunciábamos nuestra primera tentativa del problema de investigación. Tras la exploración de fuentes, formulábamos de manera fundamentada la enunciación inicial de nuestras hipótesis directivas y, en consonancia, establecíamos los objetivos de búsqueda para la revisión sistemática, así tratando de encontrar evidencias de contraste de dichas hipótesis. Como resultado, en 61 de los 67 artículos revisados (91,0%) se encontraron indicios en el correspondiente *abstract* para someter a contraste al menos una de las hipótesis específicas: *Adecuación* (H_A), *Viabilidad* (H_V) y *Relevancia* (H_R). Podemos pues afirmar la eficacia del procedimiento de búsqueda,

dado que la práctica totalidad de los artículos revisados contienen indicios de información conducente al contraste de las hipótesis.

La hipótesis sobre la que se encuentran mayor número de indicios de contraste es la de **Viabilidad** (H_V): un total de 42 artículos (62,7%). En términos globales, estos indicios se refieren a la implantación de metodologías de ‘*e-learning*’ (a distancia y con soporte tecnológico) para la enseñanza-aprendizaje del alumnado en general y para los estudiantes de alta capacidad en particular; mostrando las ventajas de dichas metodologías para la flexibilización, diferenciación e individualización curricular (véase, por ejemplo, Adams & Cross, 2000; Cope & Suppes, 2002; Dove & Zitkovich, 2003; Olszewski-Kubilius & Lee, 2004; St. Cyr, 2004; Thomson, 2010). Algún estudio (p.e., Yang *et al.*, 2011) informa de diferencias de género en el sentido de que los estudiantes masculinos de alta capacidad parecen más proclives al aprendizaje en estos entornos en línea, mientras que sus iguales femeninas tenderían en mayor medida a los entornos ‘cara a cara’.

Abundando en la hipótesis de viabilidad, Blair (2011) hace hincapié en la posibilidad que ofrecen las metodologías de ‘*e-learning*’ para que los alumnos más dotados entren en contacto con ‘iguales en capacidad’ (en vez de con ‘iguales en edad’ tal y como les obliga la escuela ordinaria). Pedersen & Liu (2002) constatan los beneficios de contar con modelado cognitivo experto en entornos de aprendizaje y resolución de problemas en línea; y en ese mismo sentido, Siegle (2003) destaca los beneficios de la ‘*e-mentoría*’ para los alumnos más capaces. Eristi (2012) pone el acento en la virtualidad que ofrece el ‘*e-learning*’ para que los estudiantes de alta capacidad puedan satisfacer su curiosidad y hambre de conocimiento. Por su parte, Hernández & Borges (2005) presentan una experiencia en España con el curso-programa “*Navegando en el aprendizaje*”, orientado a estudiantes de alta capacidad y con el objetivo de desarrollar su autorregulación en el aprendizaje. Larson & Murray (2008) informan del enorme impacto que puede suponer para los estudiantes de alta capacidad acceder a recursos educativos ‘*on-line*’ y en abierto (‘*open educational resources*’), para la aceleración de su aprendizaje en el área de matemáticas a través de una metodología ‘*b-learning*’ (combinación de clases presenciales y del uso de dichos recursos ‘*on-line*’). O también, McLester (2012), que reporta un intento de implantar de manera ‘*on-line*’ el modelo de triple enriquecimiento de Renzulli.

Por otro lado, Lee (2011) justifica como la aparición de plataformas ‘*on-line*’ para el aprendizaje de las programación informática a través de lenguajes visuales, como Scratch⁷³, hacen accesible y viable, de manera sencilla, un enriquecimiento curricular en esa área para los más capaces.⁷⁴

No es de extrañar que sea la hipótesis directiva de viabilidad la que cuente con mayor número de indicios pues el período estudiado, siglo XXI (2000-2013), coincide con la eclosión de Internet y la aparición y extensión del aprendizaje ‘*on-line*’ a través de la Red (Sacristán, 2013). Parece pues que la hipótesis de viabilidad se encuentra suficientemente sometida a contraste.

⁷³ <https://scratch.mit.edu/>

⁷⁴ Y no sólo para los más capaces. Tal y como veremos en el capítulo 2 sobre ‘Codigoalfabetización’, uno de los factores principales para que el aprendizaje de la lectoescritura con lenguajes informáticos de programación se pueda extender a la población general tiene que ver con el surgimiento de lenguajes visuales, como Scratch, que hacen mucho más accesible al aprendiz iniciarse en este campo.

A continuación, encontramos un nivel intermedio de indicios de contraste para la hipótesis de **Adecuación** (H_A): 30 artículos (44,8%). En términos generales, estos indicios se refieren a las especiales características del alumnado talentoso y de alta capacidad en lo relativo al manejo de lenguajes formales, no sólo la programación informática (Gilbert-Macmillan, 2000; Lee, 2011; Siegle, 2009), sino también las matemáticas⁷⁵ (Gadanidis *et al.*, 2011; Mann, 2006; Rotigel & Fello, 2004) o la música (Burns, 2006; Hoover & Stanley, 2009); en ocasiones música electrónica generada computacionalmente (Salaman, 2008). Dado lo sistemático de dichos lenguajes formales, su aprendizaje acelerado puede ser promovido adecuadamente en los alumnos más capaces. Coxon (2012) completa el panorama destacando cómo los sujetos de alta capacidad pueden destacar igualmente en habilidades espaciales y creativas, ambas puestas en juego a la hora de enfrentarse a tareas que involucren la programación informática.

Adicionalmente, Chan *et al.* (2010) resaltan las incipientes disciplinas STEM (ciencia, tecnología, ingeniería y matemáticas) como ámbito de excelencia prioritario a ser promovido e investigado entre los alumnos de alta capacidad de su país (Hong-Kong), dadas sus importantes implicaciones para el desarrollo económico del mismo. Lo propio hacen Wilson *et al.* (2012) en los Estados Unidos, incluyendo aquí también la robótica (Sullivan & Lin, 2012)

La hipótesis directiva de adecuación es una hipótesis ya clásica del campo de la alta capacidad, fundamentada sólidamente a lo largo del siglo XX (Clements, 1986; Clements & Swaminathan, 1995; McAllister, 1993), por lo que no es de extrañar que, ya en el siglo XXI, descienda el número de publicaciones con ese foco. Parece pues que la hipótesis de adecuación se encuentra ya sólidamente sometida a contraste.

Por último, la hipótesis sobre la que se encuentran menor número de indicios de contraste es la de **Relevancia** (H_R): tan sólo 13 artículos (19,4%). En términos generales, estos indicios se refieren al impacto que se produce en el estudiante, tras compartir y publicar en Internet alguno de sus productos de aprendizaje (no sólo ‘apps’ sino igualmente otros objetos digitales); que potencialmente son accesibles a una audiencia inmensa y global. Así, por ejemplo, encontramos el trabajo de Eckstein (2009), focalizado en un programa de enriquecimiento 2.0 para estudiantes de alta capacidad que implica la generación de productos y servicios que atienden a necesidades reales de la comunidad circundante (*‘focused toward the production of real-world products or services’*); el trabajo de Gentry (2008), que estudia los efectos sobre los estudiantes de alta capacidad de publicar y compartir *‘on-line’* sus propios libros de texto (construidos por los propios estudiantes); Siegle (2011) destaca las posibilidades de la ‘nube informática’ (*‘iCloud’*) como ubicación compartida de archivos digitales en donde los alumnos más capaces pueden presentar sus productos; o desde otra perspectiva, el trabajo Bitzer *et al.* (2007) resalta la importancia de una cultura basada en compartir ‘código informático’ para que el desarrollo del software libre sea posible, o el trabajo de Sun *et al.* (2006) que estudia el aumento de la motivación que experimentan los jugadores en entornos ‘gamificados’ en línea al ganar reputación dentro de la comunidad por compartir pistas e informaciones importantes para progresar en el juego.

⁷⁵ En el caso concreto de chicas con talento en matemáticas de Educación Secundaria, Reid & Roberts (2006) informan de la utilidad de usar mentoras *‘on-line’* expertas de cursos superiores universitarios para acelerar el aprendizaje y la autoconfianza matemáticas de aquellas.

No es de extrañar que sea la hipótesis directiva de relevancia la que cuente con menor número de indicios pues se focaliza en un hecho emergente y reciente: la web 2.0, cuyo nacimiento se suele datar alrededor de 2004 (O'Reilly, 2006) y cuya generalización podemos situar a partir de 2008 (Sacristán, 2013) a raíz de la extensión de las redes sociales; y que se basa en la idea y práctica de que el usuario de la Red no sólo se limita a recibir y consumir contenido digital sino también se dispone a generarlo y compartirlo. Parece pues que la hipótesis de relevancia se encuentra aún en vías de contrastación científica, y sobre ella deberían focalizarse las futuras investigaciones.

Y es que la actual Sociedad del Conocimiento 2.0, y en especial la 3.0 que emerge⁷⁶, demanda de las escuelas que formen ciudadanos que no sólo consuman información y aplicaciones, sino que también participen en su producción y desarrollo. Ahora bien, tal y como nos hacen notar Somyürek & Coskun (2013), no podemos dar por hecho que los jóvenes actuales, incluso los más capaces, por el hecho de haber nacido ya en un mundo digital ('nativos digitales') utilicen la tecnología para usos avanzados (p.e., recuperar y comparar información de manera crítica, trabajo colaborativo de orden superior, resolución de problemas a través del ordenador, generación de productos digitales elaborados mediante programación, etc...); por el contrario, los autores presentan datos de varios informes internacionales que indican que el uso de la tecnología por parte de estos 'nativos digitales' se circunscribe casi exclusivamente al 'entretenimiento' (videojuegos) y a la 'comunicación' (mensajería y redes sociales). Para los usos de orden superior parece necesaria la acción intencional y planificada desde las escuelas. En el caso de los alumnos de alta capacidad, es una excelente ocasión para que desde el sistema educativo se promuevan sus fortalezas; si queremos que las altas capacidades cristalicen en talentos específicos para la programación informática necesitamos un entrenamiento sistemático desde un entorno educativo que ejerza como catalizador ambiental de todo el proceso (Gagné, 2004, 2015). Con otras palabras, desde la aproximación evolutiva recientemente formulada por Olszewski-Kubilius, Subotnik, & Worrell (2015), para que el potencial de los alumnos de alta capacidad se transforme en competencia, ésta se transforme en pericia, y ésta finalmente en eminencia, deben concurrir las adecuadas oportunidades educativas que permitan el desarrollo pleno de su talento, en este caso en el ámbito informático.

En síntesis, lejos del estereotipo del 'superdotado' (masculino, de raza blanca, de clase media-alta) diseñando en soledad un programa informático (como mero juego formal), la programación de 'apps' emerge como una competencia con fuertes connotaciones sociocomunicativas, transversal a todos los géneros y culturas. (Román-González, 2013b). De hecho, transversal hasta el punto de erigirse como un 'nuevo alfabetismo', susceptible de extenderse a la población escolar general y a la creación de todo tipo de objetos digitales (no sólo 'apps') como animaciones, juegos, webs, etc. De ello nos ocuparemos en el próximo apartado Capítulo 2 sobre 'Codigoalfabetización'.

1.5.5.1. ¿Altas capacidades digitales?

En este punto, evocamos a Tannenbaum (1986, 1991) que, desde su perspectiva social de la alta capacidad, nos recuerda que es la sociedad (cada sociedad) y su cultura la que determina la valía de un producto, la que hace acreedores de capacidad y talento a aquellos capaces de elaborarlos, y la que facilita o dificulta su realización desde el propio sistema educativo. En este sentido, y dado el

⁷⁶ En el próximo Capítulo 2 sobre 'Codigoalfabetización' se expondrá cómo precisamente la capacidad de leer y escribir con lenguajes informáticos está emergiendo como competencia central de la incipiente sociedad 3.0

progresivo valor que nuestra sociedad de la información está concediendo a las aplicaciones para dispositivos móviles, cabría preguntarse si son los denominados alumnos de alta capacidad los más indicados para la programación de ‘apps’; o si acaso son aquéllos que tengan la habilidad para programar ‘apps’ los que pueden denominarse alumnos de alta capacidad.

Dicho de otra manera, en este juego de espejos: ¿es la alta capacidad un indicador para acometer la programación de ‘apps’? (perspectiva que hemos seguido hasta ahora) ¿O será la programación de ‘apps’ un indicador para la determinación de la alta capacidad? (perspectiva inversa que anima este sub-epígrafe de discusión) Interesante debate que es abordado en algunas de las publicaciones incluidas en nuestra revisión sistemática, en la que encontramos autores que se han interesado por la definición y estudio explícito del constructo ‘altas capacidades digitales’ (*‘digital gifts and talents’*).

Así, por ejemplo Siegle (2004), afirma que:

“El talento y las altas capacidades se presentan en muy diversas formas. Una posible área de expresión del talento es la tecnología digital. Y estudiantes con altas capacidades en esta área podrían no ser identificados correctamente como tales a través de los métodos tradicionales. Como cualquier talento, es de vital importancia reconocer a los estudiantes tecnológica-digitalmente dotados con objeto de promover sus altas capacidades en ese campo” (Siegle, 2004, p. 30)

Según Siegle (2004) el talento tecnológico-digital se expresa a través de una ejecución experta en dos sub-áreas de actividad: a) pericia en la programación informática de computadoras (foco central de nuestra tesis); y b) pericia como consumidor-usuario de hardware y software. Siegle (2004) afirma que los sujetos con desempeño excelente en ambas sub-áreas pueden considerarse como ‘tecnológica-digitalmente dotados’; y a su vez desarrolla con más profundidad la segunda de ellas (consumidor-usuario experto de hardware y software), que estaría internamente compuesta por las siguientes cuatro dimensiones:

- a. Pericia usando la tecnología digital: los sujetos dotados digitalmente a menudo adquieren las habilidades de uso de la tecnología más rápido y a más temprana edad que sus iguales. Además, es común que estos sujetos digitalmente dotados sean capaces de transferir fácilmente lo aprendido con una determinada tecnología a otra similar (p.e. no sólo aprenden rápido y con facilidad a manejar un determinado software sino que son capaces de transferir dicho aprendizaje a software similares, dado que tienen la habilidad de detectar los patrones comunes a ambos)
- b. Interés e iniciativa en el uso de la tecnología digital: los sujetos dotados digitalmente suelen experimentar y ser autodidactas en el aprendizaje del nuevo software y hardware que va apareciendo; en muchas ocasiones no necesitan de entrenamiento formal para realizar sus progresos. Siegle (2004) destaca que en esta dimensión concreta se encuentran diferencias de género a favor de los chicos, que se muestran más proclives a la experimentación; es posible que las chicas necesiten de modelos femeninos de excelencia en el campo del talento digital para que ganen confianza e iniciativa.

- c. Mentoría a terceros sobre tecnología digital: los dotados digitalmente suelen asistir y ayudar espontáneamente a sus iguales en los problemas que encuentran con el uso de la tecnología.
- d. Integración de tecnologías digitales: por último, los sujetos con altas capacidades digitales son capaces de incorporar de manera integrada y creativa distintos tipos de programas-software y/o hardware en productos digitales de alto nivel (p.e. al diseñar una web que incluye texto, hipertexto, imagen, un reproductor de audio incrustado y un *script*⁷⁷ de Java para dotar a la página de interactividad)

Por su parte, O'Brien, Friedman-Nimz, Lacey y Denson (2005) parten de la constatación de que en el emergente mundo digital es obvia la existencia de estudiantes que muestran una habilidad excepcional para aprender y trabajar con ordenadores (p.e. citan los casos de Steve Jobs, que funda Apple con tan solo 21 años, o Bill Gates, que hace lo propio con Microsoft a los 20). En ese sentido se plantean explorar y determinar de qué está compuesto el talento tecnológico-digital. Una primera afirmación que realizan estos autores es que los dotados digitalmente usan el ordenador como 'compañero cognitivo' (*'cognitive partner'*), es decir, como un amplificador-acelerador de sus procesos de pensamiento y solución de problemas; en vez de usar el ordenador simplemente como un buscador-transmisor de información (*'computer as a tutor'*). En esta misma línea, constatan como los dotados digitalmente utilizan los ordenadores para resolver problemas con una mayor planificación previa, y sin recurrir tanto al ensayo y error como los menos dotados; como resultado, los dotados digitalmente utilizan el software de manera más eficiente y rápida.

O'Brien *et al.* (2005) tratan entonces de caracterizar a los sujetos con talento tecnológico-digital, partiendo de algunos estudios previos de la psicóloga del MIT, Sherry Turkle (1995), que describió dos tipos de talentos avanzados con el manejo del ordenador: los *'hackers'* (focalizados en grandes problemas y sistemas informáticos complejos, para llevarlos a sus límites de funcionamiento); y los *'hobbyists'* (cuyo objetivo es reducir al ordenador a sus elementos básicos y más sencillos, para comprender y mejorar su uso). Finalmente, O'Brien *et al.* (2005) categorizan dos tipos de talentos tecnológico-digitales:

- 'Programadores' (*'programmers'* = orientados al interior de la máquina): son estudiantes que destacan y disfrutan trabajando con los lenguajes informáticos de programación. Pueden pasar largos periodos de tiempo escribiendo y reescribiendo código de manera persistente, y suelen preferir el trabajo independiente. Se caracterizan a sí mismos con altas capacidades lógicas, espaciales y de solución de problemas.
- 'Intefacers' (*'intefacers'* = orientados al punto de encuentro entre máquina y seres humanos): son estudiantes que destacan y disfrutan ayudando a sus iguales (y profesores) con las dificultades informáticas que surgen de su uso cotidiano. No les interesa tanto el ordenador en sí, como la interacción social que surge de ayudar y colaborar con otros en resolver las dificultades con el manejo de la informática. Su estilo de trabajo es más colaborativo.

Estas nuevas reformulaciones de las altas capacidades y del talento en el mundo digital (que dejamos por ahora planteadas para ser retomadas más adelante cuando tengamos que interpretar los resultados

⁷⁷ *Script*: programa informático sencillo, apenas compuesto por unas pocas líneas de código.

de nuestros estudios empíricos) también traen consigo una actualización de los modelos de intervención educativa con los más capaces. Por ejemplo, puede consultarse el modelo de enriquecimiento 2.0 de Eckstein (2009); la intervención con alumnos de altas capacidades a través de entornos de programación multimedia como Scratch⁷⁸ (Lee, 2011); o nuestro propio modelo de enriquecimiento curricular a través de la programación de ‘apps’, que se ha venido presentando en detalle a lo largo de este capítulo, y que en versión abreviada ya ha obtenido aval de la comunidad científica (Román-González, 2014a)

1.6. Conclusiones parciales

Para finalizar este Capítulo 1, se pueden extraer ya algunas conclusiones parciales derivadas de la revisión sistemática:

- La casi inexistencia de artículos científicos sobre el problema de investigación publicados en lengua española y sobre contextos educativos españoles. Existe pues un vacío por completar en España al respecto.
- Hemos detectado la revista científica más prolífica e interesada sobre el problema de investigación: *Gifted Child Today*; se trata de una publicación que aúna el foco de la alta capacidad con asuntos de actualidad como la tecnología, frente a revistas más ortodoxas del área como *Gifted Child Quarterly*.
- Con respecto al año de publicación de los artículos, se observa un lento crescendo a lo largo del siglo XXI con un pico destacable en el último trienio: 23 artículos en el periodo 2011-2013; lo que puede ser indicativo de la potencia, actualidad y previsible crecimiento de este campo de estudio.
- Finalmente, hay que destacar la preponderancia de experiencias en la etapa de Educación Secundaria. Ello da una pista para posteriormente saber dónde acotar los estudios empíricos sobre el problema.

En síntesis, y dando respuesta a nuestro problema de investigación, tal y como ha sido formulado inicialmente, consideramos que hay evidencia suficiente en la literatura científica actual para contrastar la posibilidad y deseabilidad de aprender a programar ‘apps’, como enriquecimiento curricular indicado especialmente para alumnado de alta capacidad. Dicho enriquecimiento sería significativo y completo, al profundizar respectivamente en el contenido, en el proceso y en el producto de su aprendizaje.

Queda pendiente la contrastación empírica de las hipótesis directivas enunciadas en el contexto del sistema educativo español. Ello podría acometerse a través de la implantación y evaluación de un programa-curso educativo de enriquecimiento para los más capaces con las siguientes coordenadas: a) contenido de aprendizaje: programación de ‘apps’ o ‘web-apps’; b) metodología de aprendizaje: ‘flipped classroom’; c) recursos de aprendizaje: curso ‘Computer Programming’ (Khan Academy, 2015); d) etapa educativa: Educación Secundaria (Aula de Informática o Tecnología). Se abordará, en un juego de espejos simétricos, al final de la parte empírica de esta tesis doctoral.

⁷⁸ <https://scratch.mit.edu/>

CAPÍTULO 2

CODIGOALFABETIZACIÓN

Este segundo capítulo de nuestra tesis doctoral amplía el foco de investigación desde la población específica de sujetos de alta capacidad a la población escolar general; y desde la programación de aplicaciones para dispositivos móviles o ‘apps’, a la programación de cualquier objeto digital (animaciones, juegos, simulaciones, aplicaciones, etc...). Para dicha ampliación de foco, será central concebir la lectoescritura con los lenguajes informáticos de programación como un nuevo alfabetismo de la incipiente sociedad 3.0, crítico para la formación de ciudadanos y profesionales plenamente digitales.

En consecuencia, en este capítulo se procederá a reformular nuestro problema general de investigación y, finalmente, a la enunciación definitiva de nuestras hipótesis directivas.

2.1. Algunas definiciones previas

Como paso previo, ofrecemos algunas definiciones y precisiones terminológicas que esperamos contribuyan a una lectura más ágil y precisa de las páginas que siguen. Primeramente, tal y como ya dijimos en la introducción, a lo largo de la tesis doctoral se utilizan alternativamente como sinónimos los términos ‘ordenador’ (de mayor uso en el ámbito español) y ‘computadora’ (de mayor uso en el ámbito sudamericano). La siguiente distinción importante a realizar, es la de los términos ‘algoritmo’, ‘programa’ y ‘código’.

- ✓ Un ‘algoritmo’ es una secuencia ordenada y exhaustiva de operaciones o pasos, que permite llegar unívocamente a la solución de un cierto tipo de problemas. Los algoritmos son producciones de la mente humana, y se formulan para ser comprendidos por humanos. Por ejemplo, una receta de cocina es un algoritmo, en la medida que explicita el conjunto ordenado de acciones a seguir para la realización de un plato; otros ejemplos de algoritmos serían: una secuencia de indicaciones que damos a un conductor perdido en busca de su destino; el folleto de instrucciones para montar un mueble de IKEA, etc.
- ✓ Un ‘programa’ (entiéndase ‘programa informático’) es un caso particular de ‘algoritmo’. Así, decimos que un programa es un algoritmo particularmente expresado, de forma que pueda ser entendido, procesado y ejecutado por un ordenador. Para que un ‘algoritmo’ se convierta en ‘programa’, el problema subyacente que soluciona debe poder ser expresado como un conjunto de instrucciones computables por el ordenador.
- ✓ A su vez, un mismo ‘programa’ puede ser escrito-codificado a través de diferentes lenguajes informáticos de programación, cada uno de ellos con sus reglas sintácticas propias. Así, por ejemplo, el famoso programa ‘Hello World!’⁷⁹ puede escribirse a través de código HTML, JavaScript, Python, Ruby, etc... Se ilustra en la Figura 2.1.

⁷⁹ El programa ‘Hello World!’, en español ‘¡Hola Mundo!’, es un programa informático que muestra el texto “Hello World!” en el dispositivo de visualización, habitualmente una pantalla. Es un programa extraordinariamente sencillo, que

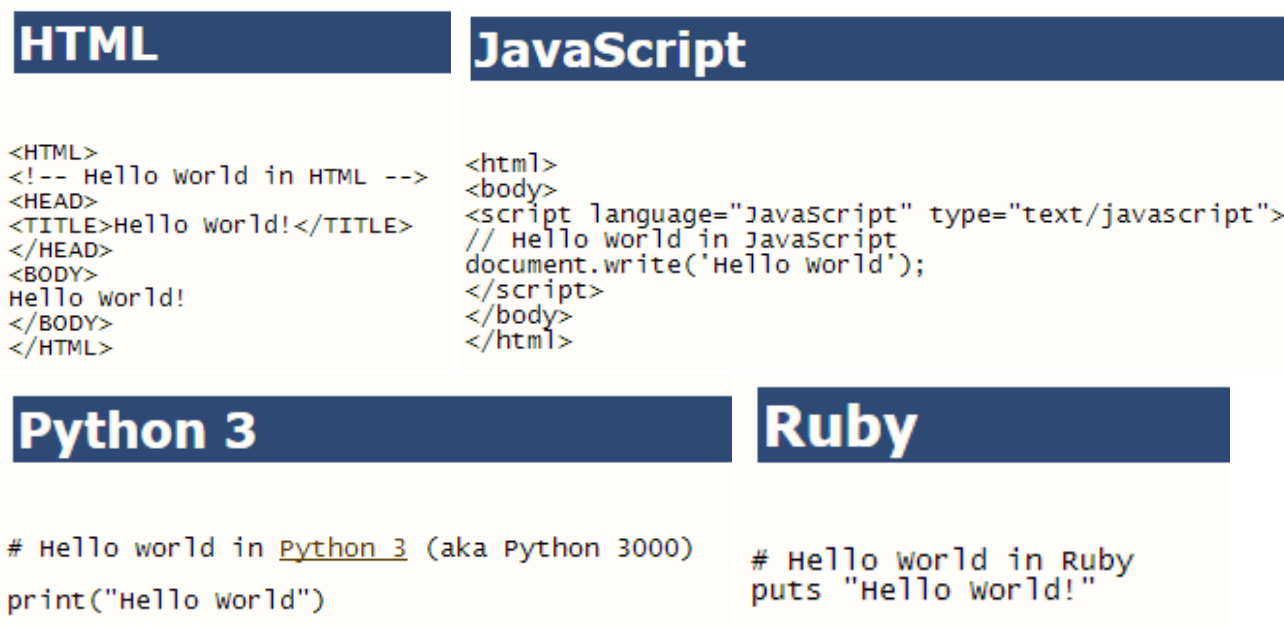


Figura 2.1. El programa ‘Hello World!’⁸⁰, escrito en código HTML, JavaScript, Python (versión 3) y Ruby

Así pues, todo ‘programa’ está basado en un ‘algoritmo’ subyacente; pero no todo ‘algoritmo’ puede ser expresado como ‘programa’. Por otro lado, un mismo ‘programa’ puede ser escrito a través de distinto ‘código’; y también puede existir ‘código muerto’, mal escrito o disfuncional, que no responda a ningún ‘programa’.

En cualquier caso, y a efectos de esta tesis doctoral, consideraremos sinónimos los términos ‘programming’ (‘programación’) y ‘coding’ (‘codificación’). Aunque, en sentido estricto, el primero es un término más genérico al hacer referencia a la concepción de un programa informático, respecto al segundo término que se refiere a su escritura en una sintaxis concreta; en la realidad, ambos se utilizan como sinónimos (quizás con el único matiz de que ‘programming’ es un término más antiguo, ya existente desde el nacimiento de los ordenadores y que es utilizado mayoritariamente por los informáticos de 40 años en adelante; mientras que ‘coding’ es un término más novedoso, que ha cogido fuerza a raíz de la reciente explosión digital, y es más utilizado entre la población joven).

Por otro lado, es muy revelador que el término inglés ‘Computer Science’ (literalmente ‘Ciencias de la Computación’) tenga como equivalencia en español ‘Informática’. Tanto ‘Computer Science’ como ‘Informática’ remiten al estudio combinado del ‘hardware’ (componentes físicos) y del ‘software’ (componentes lógico-simbólicos) de un ordenador o conjunto de ordenadores. A lo largo de esta tesis doctoral, utilizamos ‘Computer Science’ e ‘Informática’ como sinónimos, pero nos parece necesario recalcar, en este punto, el distinto campo semántico asociado a cada uno de ellos:

- ✓ Así, ‘Computer Science’ nos lleva a un campo semántico en el cual la disciplina nombrada adquiere rango de ciencia, y en donde sus expertos son científicos en computación (‘computar’, a su vez, significa contar o calcular algo mediante un sistema numérico; en este

se suele utilizar para introducir e ilustrar a los estudiantes novatos sobre la sintaxis básica de un lenguaje de programación. También se suele utilizar para testar si un sistema operativo está funcionando correctamente.

⁸⁰ Se puede consultar el programa ‘Hello World!’ escrito en código de más de un centenar de lenguajes informáticos diferentes en <http://helloworldcollection.de/>

caso, el sistema numérico binario). Así, el centro de gravedad del término ‘*Computer Science*’ está desplazado hacia una aproximación activa-productiva al hardware y, especialmente, al software.

- ✓ En contraste, el término ‘*Informática*’ nos lleva a un campo semántico diferente en el cual la disciplina nombrada se sitúa en el rango de los saberes instrumentales, y en donde sus expertos son concebidos más bien como técnicos. Así las cosas, no es de extrañar que ‘saber de Informática’ esté más vinculado a un uso-consumo experto de aplicaciones y equipos; o que los cursos o asignaturas de Informática se encuentren frecuentemente vinculados al aprendizaje de herramientas predefinidas de ofimática o navegación web.

La anterior distinción terminológica no es en absoluto baladí, y tiene unas implicaciones fundamentales en la esfera educativa (Dans, 2015a). De hecho, no es de extrañar que, como veremos en este capítulo, los nuevos y punteros currículums educativos en el mundo anglosajón orientados hacia la programación informática lleven el nombre de ‘*Computer Science Education*’; frente a los currículums anteriores, orientados al mero uso de aplicaciones informáticas, conocidos como ‘*ICT Education = Information and Communication Technology Education*’ (‘Educación en TIC = Tecnologías de la Información y la Comunicación’).

En el corazón de este capítulo estará, precisamente, la fundamentación de una nueva aproximación a la ‘*Informática*’ en nuestro sistema educativo que incluya a las ‘*Ciencias de la Computación*’ y, por tanto, la códigoalfabetización de nuestros estudiantes.

2.2. Ampliación del foco de investigación

Para abordar la ampliación del foco de nuestra investigación, nos introducimos por 3 líneas de indagación teórica: en primer lugar, nos interrogamos sobre si está emergiendo un nuevo tipo de inteligencia, la ‘inteligencia digital’ (Battro & Denham, 2007), en paralelo a la eclosión de los miles de objetos digitales que nos rodean y que median nuestra acciones cotidianas; en segundo lugar, tratamos de encajar la lectoescritura con los lenguajes informáticos de programación dentro del paradigma de ‘nuevos alfabetismos’ digitales (Lankshear & Knobel, 2008); en tercer lugar, describimos el nacimiento y rápida expansión del movimiento ‘*code-literacy*’ (traducido por nosotros como ‘códigoalfabetización’) desde los Estados Unidos (Rushkoff, 2010, 2012), y que ha cristalizado en el evento mundial de promoción del ‘*coding*’ en las aulas de todo el planeta denominado ‘*The Hour of Code*’⁸¹ (‘La Hora del Código’)

2.2.1. Hacia una inteligencia digital

Si queremos extender la lectoescritura con lenguajes informáticos de programación, como una habilidad predicable no sólo de los sujetos de alta capacidad sino también de la población general; entonces deberíamos fundamentar una inteligencia subyacente a dicha habilidad, igualmente predicable para todos los sujetos, más allá de la existencia de distintos niveles en la misma. En este sentido, y en el marco de la teoría de las Inteligencias Múltiples (IM) de Howard Gardner (1983,

⁸¹ <https://hourofcode.com>

1995, 1999⁸²), se vienen realizando en los últimos años algunos esfuerzos por describir una nueva inteligencia, la ‘inteligencia digital’; siendo la lectoescritura de código informático una de sus manifestaciones al máximo nivel.

Para comenzar esta indagación, acudimos a la definición de inteligencia que ofrece Gardner (1999): una *inteligencia* es un “*potencial biopsicológico para procesar información que se puede activar en un marco sociocultural para resolver problemas o crear productos que tienen valor para dicha cultura*”. Pues bien, tal y como dijimos en la introducción a esta tesis, parece evidente que vivimos inmersos en un ecosistema digital, rodeados por objetos e informaciones en formato digital, resolviendo problemas de manera continua a través de medios y herramientas digitales (ordenadores, tabletas, *smartphones*...), y que los productos digitales (p.e. páginas web, ‘*apps*’, software diverso...) se han constituido como objetos de gran valor en nuestra sociedad y economía actuales. En este contexto, nos preguntamos si, tal y como afirma Adams (2004), de esta continua interacción con la tecnología digital está emergiendo un nuevo tipo de inteligencia: la ‘inteligencia digital’.

A priori, podríamos inclinarnos a decir que lo digital no es tanto una inteligencia con entidad propia sino más bien un nuevo *ámbito*. Gardner (1999) ya advierte del error habitual de confundir una *inteligencia* con un *ámbito*: “*una inteligencia es un nuevo tipo de constructo que se basa en potencialidades y capacidades de carácter biológico y psicológico. No se debe confundir con ningún ámbito o disciplina, que son esferas de actividad humana construidas socialmente*” (Gardner, 1999, p. 82). En un determinado ámbito se pueden proyectar distintas inteligencias. Así, por ejemplo, el ámbito de la actuación musical supone poner en juego las inteligencias personales, la corporal-cinestésica y la musical.

De hecho, el propio Gardner (1995) describe de manera clara cómo programar un ordenador, diríamos una competencia digital de nivel superior, es una actividad que implica simultáneamente múltiples inteligencias:

“La inteligencia lógico-matemática parece central, porque programar se basa en la implementación de procedimiento estrictos [algoritmos] para resolver un problema o alcanzar una meta en un número finito de pasos. La inteligencia lingüística también es relevante, en la medida que los lenguajes de programación hacen uso del lenguaje ordinario. Un individuo con una fuerte inclinación musical podría ser introducido más fácilmente en la programación intentando dominar un programa que componga música o intentado el propio individuo programar una pieza musical. Un individuo con fuertes habilidades espaciales podría ser iniciado en la programación a través del diseño de gráficos en el ordenador, y puede apoyarse para sus tareas de programación en el uso de diagramas de flujo u otros tipos de representaciones espaciales. Las inteligencias personales pueden jugar un papel importante: la extensa planificación de pasos y metas que debe establecer el individuo involucrado en la programación se basa en formas intrapersonales de pensamiento; así como la cooperación necesaria para la realización de tareas complejas de programación o para aprender nuevas habilidades computacionales descansan en la capacidad del individuo

⁸² En su formulación actual, la teoría IM postula la existencia de 8 inteligencias, relativamente independientes: lingüística, lógico-matemática, musical, corporal-cinestésica, espacial, interpersonal, intrapersonal, y naturalista.

para trabajar en equipo. Finalmente, la inteligencia cinestésica puede jugar un papel en el trabajo con la computadora en sí, facilitando el manejo habilidoso con el terminal”

Sin embargo, hay varias razones que nos llevan a especular sobre la posibilidad de que lo digital vaya más allá de ser un simple ámbito de proyección de las inteligencias ya conocidas, y emerja como una nueva inteligencia con entidad propia, que cumpla con los criterios establecidos por la teoría IM:

- Siguiendo a Manovich (2013) en su reciente libro *“Software takes command”* (*“El software toma el mando”*), el ordenador se ha convertido en un meta-medio, un nuevo canal a través del cual los medios tradicionales (prensa, televisión, radio, etc...) se vehiculan. En esa misma medida, el lenguaje para manejar los ordenadores, es decir los distintos lenguajes de programación que indican al ordenador qué hacer, se convierte en una especie de meta-lenguaje. Así, a través de un lenguaje de programación informático podemos ‘escribir’ texto, imágenes, sonidos, música, animaciones, aplicaciones en formato digital. Siguiendo con esa misma analogía, la inteligencia digital, que es la que permitiría desplegar ese meta-lenguaje (el lenguaje de programación) para manejar ese meta-medio (el ordenador), emerge como una nueva meta-inteligencia. De esta misma manera, como meta-inteligencia, se refiere Adams (2004) a la inteligencia digital al certificar cómo la tecnología está transformando cualitativamente la estructura de las inteligencias humanas.
- Ya decía McLuhan (1994) que *“en primer lugar los seres humanos damos forma a la tecnología; y posteriormente dicha tecnología da forma a la mente humana”*. Efectivamente, el ser humano ha creado la tecnología digital y ahora parece plausible que la inmersión continua en dichas tecnologías digitales esté reconfigurando nuestra mente y alumbrando una nueva forma de inteligencia.
- El propio Gardner (1999) reconoce que la aparición y extensión masiva de las nuevas tecnologías digitales en Internet pueden traer consigo nuevas habilidades y perfiles como el *‘symbol analysts’* (‘analistas de símbolos’) o el *‘masters of change’* (‘maestros del cambio’). Estas habilidades se relacionan con la capacidad primero de buscar, manejar y seleccionar de manera crítica entre el mar de datos masivos (*big data*) e informaciones no lineales e hipertextuales que pueblan la red; y la capacidad segundo de ordenar, detectar, manipular y mostrar las tendencias y procesos de cambio subyacentes a esos datos.

Tal y como dice Adams (2004):

“Nos hemos trasladado a un espacio digital multidimensional. La información ya no se presenta de forma lineal sino a menudo agrupada de manera significativa y orientada hacia otros objetos. Debido a las nuevas funciones que ofrecen las tecnologías digitales, la información/conocimiento debe ser personalmente organizado y reorganizado de manera continua. Podría decirse que aquellos con la habilidad de comprender e interactuar con este océano digital de información, y con la capacidad de organizarlo, manipularlo y mostrarlo (visualizarlo) de acuerdo con sus percepciones y objetivos poseen ya otra inteligencia (...) tal y como describe Gardner, existen individuos virtuosos con la característica de ser analistas de símbolos y maestros del cambio. Aquellos con este talento podrían ser definidos como

digitalmente inteligentes (...) Continuando con el criterio de Gardner de la presencia de un sistema de codificación simbólico y universal para definir la existencia de una inteligencia con entidad propia, no hay duda de la actual universalidad de los medios digitales a través de las distintas culturas. El desarrollo de los iconos de ordenador utilizados para la comunicación dentro de un entorno digital satisface el criterio de codificación en un sistema de símbolos. Cuando se utiliza el criterio propio de Gardner para la clasificación de una inteligencia se puede afirmar lógicamente la existencia de una inteligencia digital” (Adams, 2004, p. 95)

Todo ello ha dado pie a someter a la ‘inteligencia digital’ a un escrutinio más detallado de los criterios de la teoría IM, con objeto de comprobar si es una candidata válida a engrosar la lista de inteligencias múltiples. En este contexto, nos parecen especialmente destacables los trabajos de Antonio M. Battro (Battro, 2009a; Battro, 2009b; Battro & Denham, 2007). En el libro “*Hacia una inteligencia digital*” (Battro & Denham, 2007), los autores declaran que:

“Es un ensayo para responder al desafío planteado por la teoría de las inteligencias múltiples de Gardner en la era del cerebro y las computadoras (...) Vivimos en una era digital. Es imposible negarlo. La historia de la humanidad ha cambiado con el advenimiento de la computadora y de Internet. Millones de mujeres y varones, niños, jóvenes y ancianos, usan hoy los nuevos instrumentos digitales y muchos lo hacen con una facilidad pasmosa. Después de haber trabajado durante casi dos décadas en el tema pensamos que hay suficientes pruebas para afirmar la existencia de una ¿nueva? capacidad en la mente humana. Proponemos identificar esta capacidad intelectual como una verdadera inteligencia digital, origen y, también resultado, de la tecnología digital de nuestros días” (Battro & Denham, 2007, p. 2-5)

Enmarcan pues su estudio en la teoría IM de Gardner, con la confianza de que la ‘inteligencia digital’ sea una buena candidata para engrosar la lista de las ocho inteligencias ya aceptadas. Para ello, someten estrictamente la ‘inteligencia digital’ a los ocho criterios de la teoría IM (Battro & Denham, 2007):

- *La inteligencia digital puede ser aislada en caso de lesión cerebral.* Se comprueba que en muchos casos de acalculia y afasia no se pierden las capacidades digitales, lo que iría a favor de una relativa independencia de la inteligencia digital respecto de la inteligencia lógico-matemática y de la inteligencia lingüística. Existen observaciones recientes que permiten inferir la presencia de circuitos neuronales dedicados a la ejecución de las actividades cognitivas de carácter digital. Así, por ejemplo, Battro (2009a) sugiere que algunos circuitos neuronales se están reciclando y diferenciando para adaptarse al nuevo entorno digital, especializándose en las tareas de búsqueda, exploración y navegación por Internet: son los antiguos circuitos neuronales dedicados a la formación de una ‘*imagen de la ciudad = espacio físico*’ (que se representa a través de cinco elementos invariantes: caminos, bordes-fronteras, zonas-áreas, nodos-atractores, e hitos-singularidades) que se reciclan para la formación de una ‘*imagen de la red = hiperespacio digital*’ (que se representa a través de cinco elementos invariantes análogos: secuencias de links, marcos, directorios de carpetas y menús, botones, e iconos).

- *La inteligencia digital tiene una historia evolutiva propia.* Se pueden rastrear las trazas de una inteligencia digital en la era previa a la aparición de la tecnología digital que hoy conocemos; especialmente en el uso de instrumentos que se basan en la activación manual de un dispositivo simple, mediante un simple clic. Esta acción elemental, de carácter binario, es seguramente el fundamento evolutivo de una inteligencia digital, que se manifiesta también en otras especies. Así, por ejemplo, muchos experimentos de laboratorio están basados en animales que deben realizar decisiones binarias (elecciones simples entre dos alternativas) a través del accionamiento (o no) de botones, interruptores y palancas.
- *La inteligencia digital está compuesta por dos sub-inteligencias, que definen sus funciones esenciales.* Se entiende que la inteligencia digital está formada, al menos, por los siguientes dos componentes: la ‘opción clic’ y la ‘heurística binaria’. La ‘opción clic’, selección de una alternativa simple a través de un dispositivo binario (una palanca, un interruptor, o en la actualidad un clic de ratón) es la operación mínima de la inteligencia digital; la ‘opción clic’ está presente a lo largo de todas las épocas y culturas, pero no es hasta la irrupción de los ordenadores, que permiten el encadenado veloz y sucesivo de ‘clics’, cuando emerge con toda su potencia: *“Es evidente que la superioridad de una computadora sobre cualquier otra máquina reside en su flexibilidad ilimitada para encadenar acciones elementales y formar sistemas complejos. Su versatilidad y velocidad de procesamiento convierten a la computadora en la condición necesaria para el desarrollo de una inteligencia digital. Sin computadoras la inteligencia digital no se podría desarrollar ni expresar en su plenitud, estaría latente (...) Dicho de otro modo, la computadora es la condición necesaria y la opción clic la condición suficiente en la expresión de una inteligencia digital”* (Battro & Denham, 2007, p. 21-22). Ese encadenamiento de ‘clics’ que permite el ordenador para el manejo eficaz de cualquier software (p.e. para la navegación por Internet) constituye una ‘heurística binaria’, el otro componente central de la inteligencia digital. La siguiente cita lo ilustra perfectamente:

“Lo que sí nos queda claro es que las computadoras e Internet han provocado la eclosión de los comportamientos digitales, haciéndolos ‘escalables’, al permitir la prodigiosa expansión en cascada de las opciones elementales. Esto posibilita, por primera vez en la historia evolutiva, el despliegue de una inteligencia digital genuina. Por ejemplo, en Internet un clic abre una página con múltiples opciones, al hacer otro clic en un botón se abre una nueva página o se desencadena alguna acción (se controla un robot, se escucha una música, se ve una fotografía o video) y así siguiendo. Este proceso de decisiones elementales sucesivas en la computadora parece no tener límites, como lo demuestra la combinatoria booleana subyacente” (Battro & Denham, 2007, p. 30)

- *La inteligencia digital se puede codificar en un sistema simbólico propio,* con elementos sintácticos, semánticos y pragmáticos propios. En su nivel más avanzado, la inteligencia digital está codificada a través de los distintos lenguajes informáticos de programación (JavaScript, C+, Python, Ruby, HTML, CSS, etc...), cada uno con su propia sintaxis, que ‘por debajo de la pantalla’ indican al ordenador qué hacer. En un nivel más básico, en la ‘interfaz-pantalla’, punto de contacto entre humano y máquina, la unidad simbólica básica

sería el enlace o botón, entendido como cualquier elemento donde se pueda aplicar un ‘clic’. A su vez, estas unidades simbólicas básicas se combinan según sintácticas, semánticas y pragmáticas propias en menús, submenús, listas de iconos, desplegados de carpetas, etc... que los nativos digitales, como una lengua materna, ya tienen perfectamente interiorizados.

- *La inteligencia digital se desarrolla desde un estado novicio a un estado experto, llegando a un estado final.* Éste es seguramente el criterio más evidente y reconocible; tanto en la familia, la escuela o en el trabajo todos tenemos experiencia del progresivo despliegue de capacidades digitales que se exigen en las sucesivas etapas vitales. A este respecto, los sistemas educativos y de capacitación profesional tienen establecidos de manera explícita niveles jerárquicos de competencia digital (p.ej.: nivel de ofimática básico o avanzado; programador junior o senior). Battro & Denham (2007) recalcan que la diferencia entre un novato y un experto reside en que los primeros se centran en la herramienta digital concreta que están utilizando (los ‘sustantivos’ en terminología de Marc Prensky, 2011), mientras que los segundos se centran en cumplir sus objetivos cognitivos (los ‘verbos’ en terminología de Prensky, 2011) siendo la herramienta algo subsidiario; por ello los expertos son capaces de moverse sin problemas de unas herramientas digitales a otras.
- *La inteligencia digital tiene sus casos excepcionales, talentos y discapacidades.* La existencia reconocida de talentos precoces, prodigios y ‘savants’ en el dominio digital parece satisfacer este criterio. A modo de ejemplo, ya citado en el Capítulo 1 desde la perspectiva de las altas capacidades, cabe referir el caso de Santiago González⁸³ un joven prodigio de la programación que a la edad de 14 años, de manera autodidacta, ya podía leer y escribir de manera fluida en una docena de lenguajes informáticos de programación, y tenía publicadas a su nombre 15 aplicaciones en la ‘AppStore’ de Apple. Los ‘hackers’, en todas sus variedades, son claros ejemplos de talentos digitales hiperdesarrollados. Con respecto a la discapacidad digital, Battro & Denham (2007) refieren algunos casos de niños autistas que muestran un espectro de conductas bizarras (repetitivas, redundantes, sin sentido comunicativo) con el ordenador; lo cual constituye además un nuevo criterio de diferenciación relativa entre la inteligencia digital, que incluye una dimensión comunicativa en la que los autistas muestran discapacidad, y la inteligencia lógico-matemática, en la que muchos autistas destacan.
- *La inteligencia digital puede interferir y transferir con otras inteligencias.* En la actualidad se están realizando las primeras investigaciones, aún no concluyentes, sobre las eventuales interferencias de la inteligencia digital con otras inteligencias. En cuanto a las transferencias de la inteligencia digital, sabemos que la ejercitación continuada en el uso de los ordenadores puede facilitar la comunicación entre personas (emails, chats) y mejorar la escritura en los niños; es decir que hay efectos positivos de la inteligencia digital sobre las inteligencias interpersonal y lingüística, por ejemplo.
- *La inteligencia digital puede medirse.* Por último, Battro & Denham (2007) consideran que se puede y se debe ampliar el alcance de la psicometría tradicional para incluir la incipiente ‘inteligencia digital’, que se presta a mediciones y evaluaciones novedosas, y sus distintos constructos cognitivos derivados-operacionalizados.

⁸³ https://youtu.be/DBXZWB_dNsw

Precisamente, Battro & Denham (2007) señalan que este último criterio, la medición de la ‘inteligencia digital’, es el que debe concentrar más atención y esfuerzo por parte de los investigadores, si se desea que dicha inteligencia adquiera entidad propia. Para ello, la ‘inteligencia digital’ debe irse concretando en constructos cognitivos derivados y susceptibles de ser operacionalizados, para su posterior medición. Uno de estos constructos, el específicamente subyacente a la lectoescritura con los lenguajes informáticos de programación, es el denominado ‘**pensamiento computacional**’ (PC); que será objeto del Capítulo 3.

2.2.1.1. *Implicaciones para la generación ‘app’*

La explosión digital en la sociedad contemporánea ha inducido a muchos a imaginar un nuevo ‘tipo’ humano donde la capacidad de interactuar con las máquinas digitales es el elemento clave. Así pues, los ordenadores se constituyen en algo más que una herramienta, son un nuevo entorno, un nuevo ecosistema que está modificando la mente, y en el que una emergente inteligencia digital puede estar alumbrando. Y éste es un proceso imparable y sujeto a una tremenda aceleración.

Ya anunciaba Nicholas Negroponte (1995) que “*cada generación será más digital que la precedente*”. Efectivamente, nos encontramos ante un punto de no retorno en la que cada cohorte de estudiantes es más digital que la anterior. Además, su proceso de aprendizaje digital sigue en la mayoría de las ocasiones una trayectoria horizontal (más entre pares que recibida verticalmente de los adultos) y a través de medios informales (más en las distintas redes sociales digitales que a través de las instituciones educativas). Esta espiral de aprendizaje acelerada por millones de jóvenes conectados en red intercambiando habilidades digitales puede hacer posible que en una sola generación una nueva inteligencia, la inteligencia digital, se haga patente.

¿Cuál es el papel de la educación ante esta realidad digital? ¿Tiene la escuela algo que decir? O enunciada la pregunta según la formulación que hacen Somyürek y Coşkun (2013): ¿Es la inteligencia-competencia digital algo innato de las nuevas generaciones, o algo que debe ser educado y desarrollado a través de procesos intencionales de alfabetización? Estos autores nos recuerdan que en estos últimos años, con el uso indiscriminado de etiquetas como ‘*nativo digital*’ se ha dado por supuesto que por el hecho de serlo, de estar rodeado e inmerso en lo digital desde el nacimiento, automáticamente se adquiere competencia digital. Se ha asumido que la generación de ‘*nativos digitales*’, por serlo, adquiere de manera natural competencias digitales clave. Sin embargo, diversos estudios empíricos e informes internacionales muestran que el uso de la tecnología que hacen los ‘*nativos digitales*’ se circunscribe casi exclusivamente al ‘entretenimiento’ (videojuegos) y a la ‘comunicación’ (mensajería y redes sociales); y en mucha menor medida se aplica sobre usos cognitivos superiores (recuperar y comparar información de manera crítica, trabajo colaborativo de orden superior, resolución de problemas a través del ordenador, generación de productos digitales elaborados mediante programación, etc...). Para dar este paso entre el uso del ordenador como ‘fuente de entretenimiento y canal de comunicación’ a otros usos ampliados del ordenador como ‘acelerador cognitivo para la solución crítica y creativa de problemas’ es necesaria la mediación intencional de la escuela (Kim, Chung, & Yu, 2013).

Para cerrar el círculo de este sub-epígrafe volvemos a Gardner que en su reciente libro, “*La generación APP: Cómo los jóvenes gestionan su identidad, su privacidad y su imaginación en el*

mundo digital” (Gardner & Davis, 2014) analiza las implicaciones que la inmersión total en los medios digitales tiene sobre la mente de las nuevas generaciones. En el libro los autores tratan de analizar “cómo la disponibilidad, la proliferación y la potencia de las aplicaciones (‘apps’) definen a la juventud de nuestro tiempo como algo distinto y característico; en otras palabras, cómo se desarrolla la conciencia de los jóvenes a partir de su inmersión en un amplio surtido de aplicaciones” (Gardner & Davis, 2014, p. 26-27).

Gardner define a los jóvenes actuales como *generación APP*⁸⁴ porque dichos jóvenes no es sólo que crezcan rodeados de aplicaciones, sino que incluso “han llegado a entender el mundo como un conjunto de aplicaciones, a ver sus vidas como una serie de aplicaciones ordenadas o quizás, en muchos casos, como una única aplicación (‘superapp’) que se prolonga en el tiempo” (Gardner & Davis, p. 21). La tesis de los autores es que esta omnipresente tecnología digital está reconfigurando significativamente la identidad, la intimidad y la imaginación de las nuevas generaciones:

“La visión ‘app’ del mundo modela y, quizá, limita el modo en que se abordan los retos principales de la adolescencia y de la primera madurez. En cuanto a la identidad, cabe decir que hay una gran presión para que nos presentemos como personas impresionantes y deseables, y para garantizar que todas las señales (y posts) confirmen esa sensación de identidad que, quizá, cristalice de manera prematura. Del mismo modo, si hablamos de intimidad, la posibilidad de anunciar (en realidad, de definir) las relaciones con los demás puede impedir la exploración más plena, que si bien aumenta la vulnerabilidad, también ofrece un mayor potencial para forjar relaciones profundas y en evolución constante con personas verdaderamente importantes para nosotros. Para terminar con una nota positiva, en lo que respecta a la imaginación y la creatividad, las tecnologías digitales ofrecen un enorme potencial para la innovación, ya sea individual o en grupo, siempre que las aplicaciones que existen se entiendan como algo sobre lo que seguir construyendo (nos capacitan) en lugar de algo que nos limita o que restringe los medios de que disponemos y los objetivos que nos planteamos (nos hacen dependientes)” (Gardner & Davis, 2014, p. 156)

Es justo en este punto donde reside el papel central de la educación de las nuevas generaciones: promover jóvenes ‘app-capacitados’ (que utilicen y, sobre todo, programen aplicaciones que les permitan e inciten a buscar posibilidades nuevas, a explorar caminos y relaciones más profundas, que se sirvan de estas tecnologías como expansores-aceleradores de sus procesos creativos y de solución de problemas); frente a jóvenes ‘app-dependientes’ (que utilicen exclusivamente aplicaciones que limiten o determinen sus actos, sus elecciones y objetivos, que no les permitan plantearse preguntas nuevas, ni establecer relaciones significativas, ni modelar una identidad adecuada). El reto educativo está planteado; como ejemplo de aplicación educativa y capacitadora, el propio Gardner cita el caso de Scratch⁸⁵, un lenguaje de programación visual que permite a niños y jóvenes crear de manera sencilla historias, juegos y animaciones digitales, y compartirlas con gente de todo el mundo; y que se está utilizando en escuelas de todo el planeta como entorno de desarrollo de la inteligencia digital.

⁸⁴ Una ‘app’ o aplicación es un programa informático, generalmente diseñado para funcionar en dispositivos móviles, que permite que el usuario lleve a cabo una o varias operaciones (...) una ‘app’ puede ser limitada o amplia, sencilla o compleja... y tanto en un caso como en el otro, está perfectamente controlada por la persona u organización que la haya diseñado (...) Lo más importante es que son rápidas, satisfacen una demanda y aparecen justo a tiempo (Gardner & Davis, 2014, p. 20)

⁸⁵ <https://scratch.mit.edu/>

2.2.2. Código y nuevos alfabetismos en la era digital: hacia la sociedad 3.0

El término ‘nuevos alfabetismos’ remite a un marco conceptual-explicativo definido por los autores Lankshear y Nobel (2008). Si los ‘viejos alfabetismos’ se refieren a los procesos de lectoescritura de los textos tradicionales analógicos; los ‘nuevos alfabetismos’ hacen lo propio con respecto a los incipientes textos en formato digital. Pues bien, ¿dónde ubicar la ‘codigoalfabetización’ (*code-literacy*), esto es, los procesos de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación, dentro de los ‘nuevos alfabetismos’; diferenciándolo de otros afines como la ‘alfabetización digital’ (*digital-literacy*) o la ‘alfabetización web’ (*web-literacy*)?

Por analogía, podríamos definir la ‘alfabetización digital’ como el proceso de enseñanza-aprendizaje de la lectoescritura de textos digitales. Aquí entendemos ‘texto’ en su sentido amplio: cualquier contenido cognitivo o unidad de sentido que queda fijado según ciertas reglas en un elemento o soporte material estable de modo que pueda transmitirse en el espacio y en el tiempo (Sacristán, 2013); y ‘digital’ haría referencia al forma en que ha sido fijada la información transmitida por dicho texto: un sistema binario (*unos y ceros*).

Con el formato digital, las informaciones pueden adquirir muy diferentes representaciones: imágenes, sonidos, escritos, etc...; y además pueden almacenarse, modificarse, recuperarse y enviarse fácilmente (Sacristán, 2013). Desde esta perspectiva, un archivo *.doc* que contiene un escrito, un archivo *.jpg* que contiene una foto, un archivo *.mp3* que contiene una canción, o un archivo *.mp4* que contiene un vídeo, son ejemplos de ‘textos digitales’. Y, por tanto, entrarían dentro de la ‘alfabetización digital’ aprendizajes como el retoque fotográfico de imágenes digitales (p.ej.: a través de Photoshop⁸⁶), la edición de audios digitales (p.ej.: a través de programas como Audacity⁸⁷), o el manejo de herramientas de ofimática⁸⁸, como el paquete Microsoft Office⁸⁹, entre otros; independientemente de que estas prácticas se realicen ‘*on-line*’ u ‘*off-line*’. Podría pues afirmarse que la ‘alfabetización digital’ es un nuevo alfabetismo que se sitúa en un nivel 1.0.

Por otro lado, desde el proyecto Webmaker⁹⁰ de la Fundación Mozilla se ha definido recientemente la ‘alfabetización web’ como el conjunto de habilidades y competencias que te permiten leer, escribir y participar en la web (Belshaw, 2014a). Por tanto, la ‘alfabetización web’ sería un caso particular de la ‘alfabetización digital’, cuando ésta se produce en un entorno ‘*on-line*’. Así, la alfabetización web añade a las características de lo digital, señaladas anteriormente, las propias de la Red: integración multimedia, hipertextualidad, accesibilidad y conectividad casi ilimitada, y ambiente de participación y colaboración.

Entrarían entonces dentro de la ‘alfabetización web’ aprendizajes como la generación y gestión de un perfil en redes sociales, y la publicación de informaciones en las mismas; la edición y mantenimiento de un blog; la creación y difusión de memes a través de Internet; la curación de contenidos en la Red; la participación en comunidades de aprendizaje conectadas por Internet; el diseño de una página Web

⁸⁶ <http://www.adobe.com/es/products/photoshop.html>

⁸⁷ <http://audacity.es/>

⁸⁸ La ofimática designa al conjunto de técnicas, aplicaciones y herramientas informáticas que se utilizan en funciones de oficina para optimizar, automatizar, y mejorar tareas y procedimientos relacionados.

⁸⁹ <https://products.office.com/es-ES/>

⁹⁰ <https://webmaker.org/es/about>

y un largo etcétera. Podría pues afirmarse que la ‘alfabetización web’ es un nuevo alfabetismo que se sitúa en un nivel 2.0.

Volviendo al proyecto Webmaker de la Fundación Mozilla, recientemente han publicado sus estándares⁹¹ de ‘alfabetización web’ (Mozilla Webmaker, 2015): un mapa de conocimientos y habilidades que se consideran imprescindibles para afirmar que uno sabe leer, escribir, y participar en la web (Figura 2.2)



Figura 2.2. Estándares de ‘alfabetización web’ de la Fundación Mozilla (Mozilla Webmaker, 2015)

Como puede observarse, el ‘coding’ es sólo uno de los elementos entre otros muchos que forman parte de ese mapa de la ‘alfabetización web’, concretamente dentro del área de ‘Construcción’ (‘Build’) de la misma. Belshaw (2014a) ha señalado la utilidad de estos estándares de alfabetización web para el diseño e implantación de nuevos itinerarios curriculares en los sistemas educativos. Y remarca la importancia de contextualizar el aprendizaje del ‘coding’ dentro del marco más amplio que supone la alfabetización web (Belshaw 2014b).

Así pues, como ya podrá suponer el lector, ubicamos la ‘codigoalfabetización’ como parte del paisaje más amplio que dibuja la ‘alfabetización web’; que a su vez se inscribe en un terreno aún mayor como es la ‘alfabetización digital’.

Ahora bien, la ‘codigoalfabetización’ se sitúa en un nivel superior, un nivel 3.0, porque trabaja directamente con la materia prima común a todos los productos que se generan desde los dos anteriores alfabetismos de orden inferior. Este materia prima es el código informático: a través de los lenguajes informáticos de programación podemos ‘escribir’ directamente cualquiera de los ‘textos digitales’ mencionados anteriormente (imagen, sonido, música, vídeos, webs...) y algunos otros más avanzados e interactivos (animaciones, simulaciones, aplicaciones...); todos ellos pueden ser escritos como código informático. Como ya comentamos, hace unas páginas, en la medida que el ordenador se ha convertido en un ‘meta-medio’ a través del cual se vehiculan el resto de medios tradicionales (prensa, radio, televisión...); el lenguaje primario para comunicarse con el ordenador, es decir, el

⁹¹ <https://teach.mozilla.org/activities/web-literacy/>

código informático, se erige en un ‘meta-lenguaje’; y, por extensión, la codigofabetización emerge como un ‘meta-alfabetismo’ de orden superior 3.0⁹². En consecuencia, se viene señalando como la lectoescritura de código informático está en la base de la construcción de productos y narrativas digitales que atraviesan todo tipo de medios o ‘*transmedia*’ (Jenkins, 2010)

Estar codigofabetizado nos permite, de alguna manera, ver lo que ‘hay debajo de la pantalla’, entrar en las tripas de la máquina y poder interactuar con ella a mayores cotas de profundidad (3.0), trascendiendo el mero compartir y reenviar (a lo que en ocasiones queda reducida la alfabetización 2.0). La lectoescritura de código nos libera de las limitaciones de las opciones predefinidas y configuraciones por defecto de las aplicaciones informáticas que utilizamos cada día: por ejemplo, no es lo mismo diseñar una web o un blog a partir de las plantillas (*templates*) prediseñadas que nos ofrecen herramientas como Blogger o Wordpress, que saber leer y escribir código HTML y CSS para construir una web o blog a ajustada a nuestro gusto y necesidades; no es lo mismo aplicar filtros predefinidos por Instagram a una foto digital, que generar un filtro de color propio a partir del conocimiento de los códigos RGB⁹³

En la siguiente Figura 2.3 se ilustra la ‘alfabetización digital’, la ‘alfabetización web’, y la ‘codigofabetización’ como la sucesión jerárquica de nuevos alfabetismos que hemos descrito en las líneas anteriores.

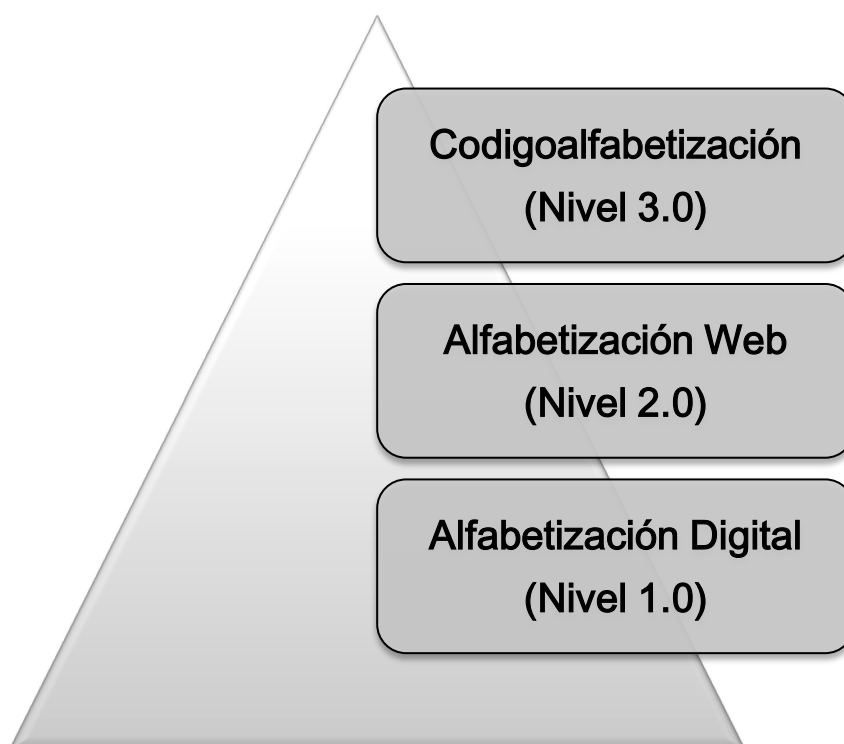


Figura 2.3. Alfabetización Digital, Alfabetización Web, y Codigofabetización

⁹² A modo ilustrativo, deseamos hacer notar que la revista electrónica “*Educación 3.0*” tiene entre sus áreas prioritarias de contenido la “programación informática” [<http://www.educaciontrespuntocero.com/tag/programacion>], y su integración en la escuela. Todos los viernes publica un artículo sobre cómo realizar algún proyecto en el aula utilizando el lenguaje visual de programación Scratch [<http://www.educaciontrespuntocero.com/tag/el-proyecto-scratch-de-los-viernes>]

⁹³ https://es.wikipedia.org/wiki/RGB#Uso_de_RGB_en_HTML_y_en_lenguajes_de_programaci.C3.B3n

2.2.3. El movimiento ‘code literacy’

El movimiento ‘code-literacy’ surge en los Estados Unidos alrededor del año 2010 y tiene como axioma fundamental el considerar, y promover, la programación informática (coloquialmente ‘coding’) como el alfabetismo clave en el siglo XXI. Consideramos que son tres sus figuras fundamentales, a saber: Marc Prensky⁹⁴, Douglas Rushkoff⁹⁵, y Mitch Resnick⁹⁶. Dedicamos unas breves páginas a revisar sus argumentos fundamentantes del movimiento ‘code-literacy’.

Marc Prensky es mundialmente conocido por acuñar, a principios de siglo (Prensky, 2001), los términos ‘nativos digitales’ (‘digital natives’) e ‘inmigrantes digitales’ (‘digital immigrants’). Los primeros son los nacidos ya en la era digital, que tienen interiorizados los textos digitales y los manejan como una ‘lengua materna’; frente a los segundos, nacidos y educados en la era analógica, y que han debido aprender a manejar los textos digitales como si se tratara de una ‘lengua extranjera’. Sin embargo, con el paso del tiempo estas etiquetas han perdido peso explicativo por dos razones. En primer lugar, el mero hecho de que, por la propia dinámica demográfica, cada vez son más los nativos y menos los inmigrantes digitales (hasta llegar a un punto en que sólo haya nativos digitales, y el término ya no discrimine en absoluto). En segundo lugar, el término de ‘nativos digitales’ introdujo un prejuicio que se ha demostrado falso: que los nativos, por el hecho de serlo, son espontáneamente capaces de utilizar la tecnología en sus niveles más avanzados (cuando la realidad viene mostrando que los usos habituales que los nativos hacen de lo digital son más bien pasivos, a lo sumo limitados a herramientas de comunicación y ocio).

Así las cosas, Prensky introduce posteriormente un nuevo concepto: la ‘sabiduría digital’ (‘digital wisdom’), ya desvinculada del año de nacimiento del sujeto, y que se define por la capacidad de utilizar las tecnologías digitales para la resolución práctica y creativa de problemas significativos para el individuo. Sabio digital es aquel que sabe combinar sus propias capacidades y habilidades con las potencialidades técnicas de las nuevas herramientas digitales. (Prensky, 2009)

En este contexto, de ‘sabiduría digital’, Prensky (2008) afirma que “*Power will soon belong to those who can master a variety of expressive human-machine interactions: Programming Is the New Literacy*” (“El poder pertenecerá pronto a aquéllos que dominen una variedad de interacciones expresivas humano-máquina: la programación informática es el nuevo alfabetismo”). Traducimos a continuación algunas líneas del autor al respecto:

“Creo que la habilidad que, por encima de cualquier otra, distinguirá a una persona alfabetizada es la codigoalfabetización (Prensky la denomina literalmente ‘programming literacy’); esto es, la capacidad de hacer que la tecnología haga cualquier cosa que uno desee hacer con ella, la capacidad de plegar la tecnología digital a las propias necesidades y propósitos, igual que siempre hemos hecho con las palabras (...) Hoy en día, las personas con habilidades altamente desarrolladas en esta área son vistas aún como ‘nerds’⁹⁷. Pero ten en cuenta que, en la medida que las máquinas se están volviendo progresivamente elementos más fundamentales de nuestra comunicación, nuestro trabajo, nuestra educación, nuestros

⁹⁴ <http://marcprensky.com/>

⁹⁵ <http://www.rushkoff.com/>

⁹⁶ <http://web.media.mit.edu/~mres/>

⁹⁷ ‘Nerds’ ≈ ‘Cerebritos’, ‘Empollones’

viajes, nuestros hogares, y nuestro ocio; la capacidad de hacer que las máquinas hagan lo que queremos se convertirá en algo cada vez más valioso (...) Según la programación se vuelva más y más importante, abandonará el ‘cuarto trasero’ y se erigirá en una competencia clave, atributo de nuevas élites sociales e intelectuales, un papel análogo al que saber y leer escribir tuvo en el pasado. Recuerda cómo, sólo hace unos siglos, leer y escribir estaba reservado a un pequeño grupo de especialistas cuyos miembros se hacían llamar ‘escribas’ (...) Programar será muy pronto la forma a través de la cual interactuaremos con todos nuestros objetos, y un componente importante de cómo interactuaremos los unos con los otros” (Prensky, 2008, en línea)

La aproximación de Douglas Rushkoff es algo distinta. Este autor es conocido por ser el firmante de uno de los lemas principales del movimiento ‘code-literacy’: el aforismo “*Program or be programmed*” (“Programar o ser programado”) (Rushkoff, 2010). Así pues, su aproximación es sociocrítica y se enuncia en términos de cómo aprender a programar empodera al sujeto frente a un mundo lleno de objetos digitales programados por otros. Traducimos, a continuación, unas palabras demoledoras del autor al respecto (Rushkoff, 2012):

“Tal y como yo lo veo, la codigoalfabetización (Rushkoff ya utiliza literalmente el término ‘code-literacy’) es un requisito para poder participar en un mundo digital. Cuando en el pasado adquirimos el lenguaje, no aprendimos sólo a cómo escuchar, sino también a cómo hablar. Cuando adquirimos los textos (analógicos) no aprendimos sólo a leerlos, sino también a escribirlos. Ahora que tenemos ordenadores, estamos aprendiendo cómo usarlos, pero no cómo programarlos. Cuando uno no está codigoalfabetizado, debe aceptar los dispositivos y el software que utiliza con todas las limitaciones (e intenciones ocultas) con que sus creadores los han construido (...) Este hecho sitúa a nuestros niños, que vivirán en un mundo cada vez más digital, en una terrible desventaja. Estos niños pasan cada vez más tiempo en entornos digitales en los cuales las reglas han sido escritas por otros. Simplemente estar familiarizados con el funcionamiento del código podría ayudarles a manejar mejor este terreno digital, comprender sus limitaciones y decidir si sus límites vienen determinados por la tecnología en sí, o sólo porque alguna empresa ha establecido que así sea. Los niños codigoalfabetizados dejan de aceptar las aplicaciones y webs que utilizan en sus ‘valores por defecto’, y, en vez de eso, comienzan a involucrarse de manera crítica e intencionada con ellas” (Rushkoff, 2012, en línea)

Rushkoff también hace hincapié en los valores sociales, de colaboración, compartición y remezcla, asociados a la codigoalfabetización. Efectivamente, el ‘coding’ 3.0, lejos de caracterizarse como mero juego intelectual y solitario con un lenguaje formal (al estilo del estereotipo clásico del programador, aislado y absorto); se erige como práctica sociocultural de primer orden que se despliega en entornos en línea, con fuertes connotaciones comunicativas y transversal a géneros. Además, la inmensa mayoría de los recursos para el aprendizaje del ‘coding’⁹⁸ se encuentra ‘on-line’ e incluyen en su metodología la esencia interactiva y colaborativa de la Red: el código no sólo se lee y se escribe, también se comparte, se copia y se remezcla.

⁹⁸ Una taxonomía completa de recursos para el aprendizaje del ‘coding’ se expondrá en el epígrafe 2.4.1

Finalmente, visitamos a Mitch Resnick, director del grupo de investigación *Lifelong Kindergarten* en el MIT Media Lab. Resnick *et al.* (2009) también dudan de que los, mal llamados, ‘nativos digitales’ posean realmente habilidades digitales de orden superior (a éstas las engloban en la denominación ‘*digital fluency*’ o ‘fluidez digital’):

*“De hecho, muchos jóvenes se sienten muy cómodos enviando mensajes de texto, jugando a juegos ‘on-line’, y navegando por la Web. ¿Pero eso les convierte en sujetos ‘fluidos’ (≈ expertos) con las nuevas tecnologías? Aunque los jóvenes interactúan constantemente con los medios y objetos digitales, sólo unos pocos son capaces de crear sus propios juegos, animaciones o simulaciones. Es como si pudieran ‘leer’ pero no ‘escribir’ (...) Tal y como nosotros lo vemos, la ‘fluidez digital’ requiere no sólo la capacidad de chatear, navegar e interactuar en las redes; sino también la capacidad de diseñar, crear, e inventar con los nuevos medios digitales. Para hacerlo, se necesita aprender algún tipo de programación informática” (Resnick *et al.*, 2009, p. 62)*

Por tanto, Resnick *et al.* (2009) establecen la codigoalfabetización como requisito para alcanzar el estadio superior de habilidad digital, o ‘fluidez digital’. Pero no se detienen en dicha aseveración, sino que trabajan por ofrecer a los niños y jóvenes de todo el mundo, aun sin experiencia previa ninguna con el código, un entorno de programación accesible y motivador. Frutos de estos trabajos es Scratch⁹⁹ (Maloney *et al.*, 2010), una plataforma ‘on-line’ de aprendizaje de la programación, que reúne una serie de características que la convierten en un recurso para la codigoalfabetización de primer orden, adecuado para niños a partir de tan sólo 8 años:

- *‘Tinkerable’*: Scratch es un lenguaje de programación visual, compuesto por bloques de código que encajan unos con otros como si fueran piezas de un puzle. El joven *scratcher* simplemente tiene que pinchar, arrastrar y soltar (*‘drag-and-drop’*) los bloques de código para ir construyendo su programa. Ésta es una manera intuitiva de programar, que la hace accesible a los niños independientemente de su experiencia previa (baja las ‘barreras de entrada’ a la programación); y que evita que los aprendices se frustren innecesariamente por no poder dominar la compleja sintaxis de los lenguajes de programación textuales (Stefik & Siebert, 2013). Además, el programa escrito con Scratch se ejecuta en tiempo real en la misma pantalla (Figura 2.4), dando un *‘feed-back’* inmediato al *scratcher* sobre su funcionamiento. Por todo ello, Scratch se considera ideal para comenzar a ‘cacharrear’ (en inglés y en este contexto, *‘to tinker’*) con el código informático.
- *‘Meaningful’*: Scratch es un lenguaje de programación que permite construir y personalizar todo tipo de proyectos-objetos¹⁰⁰ digitales. Así, permite crear juegos, animaciones, simulaciones, narraciones interactivas, etc... que contengan un significado relevante para el joven programador. A esto contribuye que Scratch permite incorporar a sus proyectos todo tipo de imágenes o sonidos; tanto extraídos de la Red, como aportados y subidos por el propio *scratcher*. Y, más recientemente, la posibilidad de conectar el código programado con

⁹⁹ <https://scratch.mit.edu/>

¹⁰⁰ Una amplia muestra de los tipos de proyectos que pueden programarse con Scratch puede consultarse, y remezclarse si se desea, en: https://scratch.mit.edu/starter_projects/

Scratch al mundo físico¹⁰¹ mediante dispositivos como MaKeyMaKey¹⁰², LegoWeDo¹⁰³, PicoBoard¹⁰⁴, o Microsoft Kinect¹⁰⁵.

- ‘Social’: Scratch no es sólo un entorno y un lenguaje de programación; es también una comunidad de jóvenes programadores. Entre ellos colaboran para la realización de sus proyectos, intercambiando y remezclando su código.

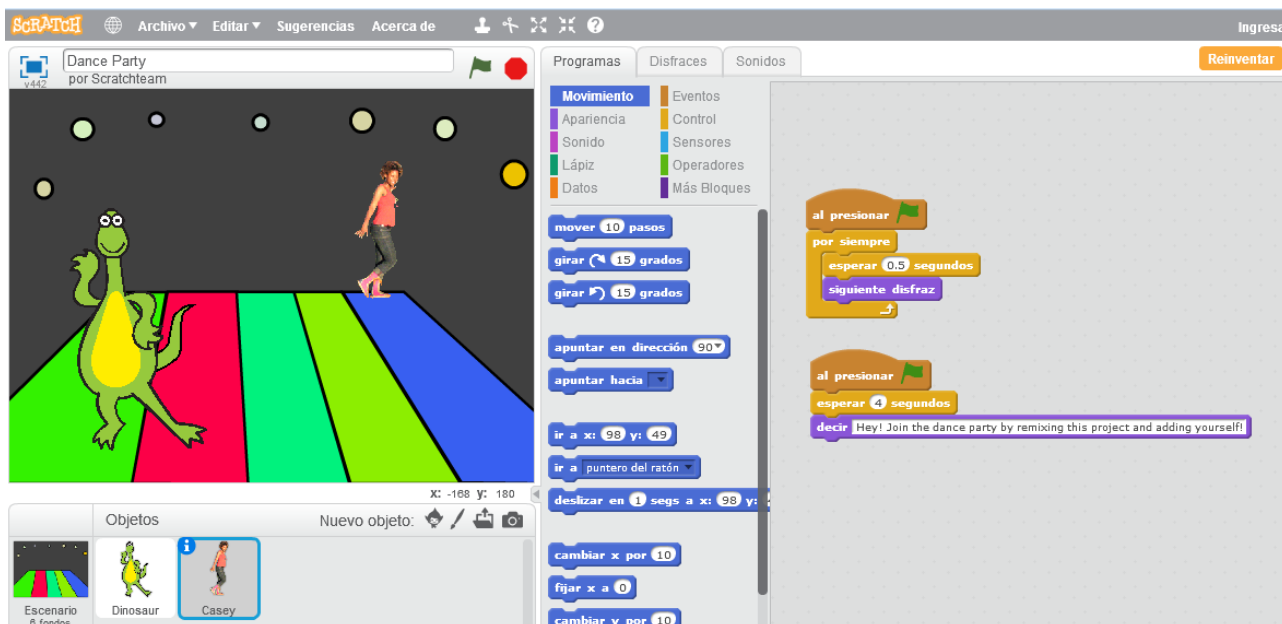


Figura 2.4. Entorno de programación Scratch: en el centro, los bloques de código disponibles, que son arrastrados y colocados en la parte derecha; a la izquierda se ejecuta el programa resultante en tiempo real [Ejemplo tomado de <https://scratch.mit.edu/projects/10128067/#editor>]

Además, Resnick *et al.* (2009), inciden en cómo la codigoalfabetización lleva consigo toda una serie de aprendizajes valiosos asociados:

“En la medida que los scratchers programan y comparten sus proyectos interactivos, también aprenden importantes conceptos matemáticos y computacionales, y aprenden cómo pensar creativamente, razonar de manera sistemática, y trabajar en equipo: todas ellas habilidades esenciales para el siglo XXI. De hecho, nuestro objetivo fundamental no es preparar a los niños para una carrera profesional como futuros programadores, sino ‘nutrir’ a una nueva generación de pensadores sistemáticos y creativos que se sienta cómoda utilizando la programación para expresar sus ideas” (Resnick *et al.*, 2009, p. 60)

Todas estas ideas han sido magistralmente expuestas por el propio Resnick en su charla “*Let’s teach kids to code*”¹⁰⁶ (“Enseñemos a los niños a programar/codificar”), dentro de la prestigiosa serie de conferencias TED Talks (Resnick, 2013a)

¹⁰¹ http://wiki.scratch.mit.edu/wiki/How_to_Connect_to_the_Physical_World

¹⁰² <http://www.makeymakey.com/>

¹⁰³ <https://education.lego.com/es-es/lesi/elementary/wedo-2>

¹⁰⁴ <http://www.picocricket.com/picoboard.html>

¹⁰⁵ <http://scratch.saorog.com/>

2.2.3.1. Un evento a escala mundial: ‘The Hour of Code’

‘La Hora del Código’¹⁰⁷ (*The Hour of Code*) es el mayor evento a nivel mundial para la promoción del aprendizaje de la programación informática (coloquialmente ‘coding’) en las escuelas (véase, por ejemplo, [Garcia et al., 2015](#); [Wilson, 2015a, 2015b](#); o en prensa¹⁰⁸, [BBC News, 2014](#); [Cadwalladr, 2014](#); [Dubinski, 2014](#); [Hawke, 2014](#); [Layton, 2014](#), [Richtel, 2014](#); [Shah, 2014](#); [Ward, 2014](#)). El evento está promovido desde la fundación estadounidense sin ánimo de lucro Code.org¹⁰⁹ cuyo lema es: “*Every student in every school should have the opportunity to learn Computer Science*” (“Cada estudiante, en cada escuela, debería tener la oportunidad de aprender Ciencias de la Computación”).

‘La Hora del Código’ se convoca anualmente coincidiendo con la ‘*Computer Science Education Week*’ (CSEW) de los Estados Unidos, durante una semana del mes de diciembre; aunque inicialmente se trató de un evento circunscrito al ámbito estadounidense, desde la edición de diciembre de 2013 adquirió un carácter global, convocándose en el resto de países. Actualmente, tal y como se describe en su página web ([Code.org, 2015c](#)):

“La Hora del Código es un movimiento global, que llega a decenas de millones de estudiantes en más de 180 países. Cualquier persona, en cualquier lugar del mundo puede organizar una Hora del Código. Tutoriales, de una hora de duración, están disponibles en más de 30 idiomas. No se necesita experiencia. Para edades entre 4 y 104 años”

En la Figura 2.5 se ilustran algunas cifras del evento relativas a su impacto acumulado en las 2 primeras ediciones celebradas a nivel mundial (diciembre de 2013 y de 2014; que son las ediciones objeto del estudio empírico que se presenta en el Capítulo 5): más de 100 millones de estudiantes de todo el planeta han probado ya ‘La Hora del Código’, lo que da cuenta de su enorme alcance.



Figura 2.5. Algunas cifras del evento ‘La Hora del Código’. Tomado de Code.org (2015c)

Podríamos sintetizar el objetivo general del evento en tres palabras, siguiendo sus propias etiquetas: ‘promover’ (*promote*), ‘motivar’ (*encourage*) y ‘desmitificar’ (*demystify*) el aprendizaje del

¹⁰⁶ Disponible en https://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code [Más de 1.2 millones de visitas a comienzos del año 2016]

¹⁰⁷ <https://hourofcode.com>

¹⁰⁸ Una impresionante colección de notas de prensa procedentes de medios de comunicación de todo el mundo puede recuperarse desde <https://csedweek.org/about/news> ó <https://code.org/about/news>

¹⁰⁹ <https://code.org/>

'coding'. Dicho en otras palabras, dar a conocer la programación informática entre la población escolar, y exponer a los estudiantes a una experiencia introductoria con el 'coding' que destierre el prejuicio de su alta dificultad y les motive para continuar con un aprendizaje más profundo y prolongado; al tiempo que sensibiliza a toda la comunidad educativa, incluidos padres y madres, profesores y equipos directivos, de la importancia del 'coding' para el futuro de sus alumnos.

En este sentido, el evento 'La Hora del Código' se convoca a partir de 2 premisas básicas, que funcionan como justificación del mismo: a) la constatación de una incipiente 'brecha laboral' entre el número de puestos de trabajo relacionados con tareas de programación que los estudios prospectivos prevén para 2020 (p.e. 1,4 millones sólo en Estados Unidos), y el insuficiente ritmo actual de estudiantes que se gradúan con competencias adecuadas para cubrir dichos puestos (Figura 2.6); y b) la afirmación de que el 'coding' es el emergente 'superpoder', dicho en términos del propio evento, del siglo XXI que permite a quien lo posee alcanzar las máximas cotas de participación y expresión en un mundo repleto de objetos digitales, en el que estamos inmersos. Estas premisas se comunican brillantemente en el vídeo "What Most Schools Don't Teach"¹¹⁰ (Code.org, 2013), pieza promocional del evento y sintetizadora del espíritu de la organización.

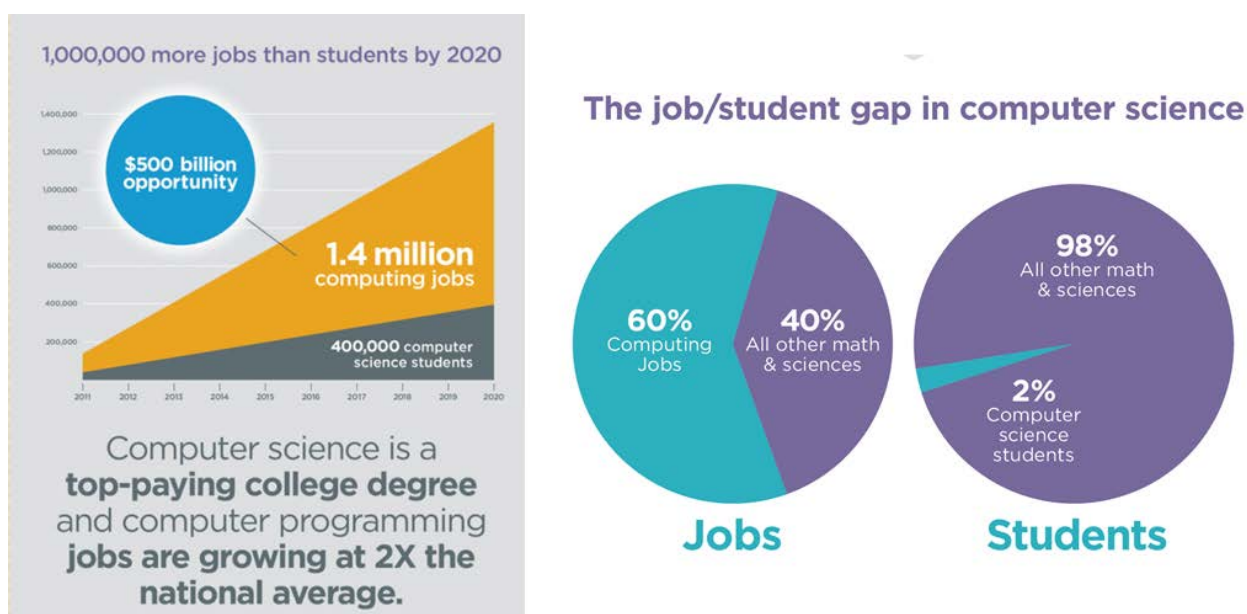


Figura 2.6. La 'brecha laboral' en el campo de la programación informática. Tomado de Code.org (2015c)

Abundando algo más en el mensaje que sirve de soporte al evento 'La Hora del Código', podemos revisar algunos de los lemas que se distribuyen como 'merchandising' asociado al mismo y con el que se anima a decorar las aulas durante su celebración (Figuras 2.7, 2.8 y 2.9):

- En la Figura 2.7 podemos leer la frase atribuida a Steve Jobs (fundador de Apple, inventor de ingenios tan valorados como el iPhone o el iPad que hoy en día nos rodean, y uno de los máximos exponentes de talento digital): "Creo que toda persona en este país debería aprender cómo programar un ordenador, porque ello te enseña a cómo pensar"; anticipando ya la perspectiva del 'pensamiento computacional' que se viene imponiendo en el discurso de aprendizaje del código, tal y como veremos en el Capítulo 3 de esta tesis.

¹¹⁰ <https://youtu.be/nKIu9yen5nc> [Más de 12 millones de visitas en YouTube a fecha 27 de octubre de 2015]

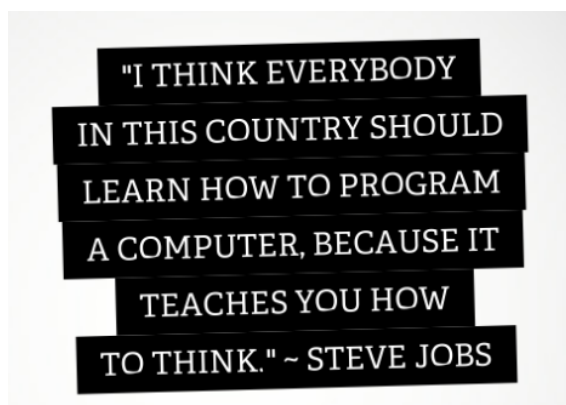


Figura 2.7. Cita de Steve Jobs utilizada como *leitmotiv* en ‘La Hora del Código’

- En la Figura 2.8, además de quedar patente el equilibrio de género que anima igualmente el evento ‘La Hora del Código’, varias figuras relevantes del entorno empresarial de Silicon Valley enuncian algunas ideas fuerza del movimiento de codigoalfabetización, a saber, de arriba abajo y de izquierda a derecha: que la lectoescritura con lenguajes de programación informáticos suponen un nuevo alfabetismo; la ubicuidad de la tecnología en el mundo digital actual, y la importancia de su creación (no sólo su consumo); la equiparación de las Ciencias de la Computación con el resto de asignaturas centrales del currículo; y la desmitificación de su extrema dificultad, especialmente para la población femenina.



Figura 2.8. Figuras relevantes de Silicon Valley (y Hollywood) promoviendo el evento. Tomado de Code.org (2015c)

- Finalmente, en la Figura 2.9, el propio presidente de los Estados Unidos, Barack Obama, anima a los estudiantes a “no sólo jugar con su teléfono móvil, sino también a programarlo”, situando en el foco de discurso (incluso diríamos a modo de prioridad nacional) la necesidad de elevar el nivel de alfabetización digital de los estudiantes hacia estándares que incluyan la creación, y no sólo el consumo, de objetos digitales a través de la programación; es decir, ascender a los niveles propios de la codigoalfabetización, o nivel 3.0.



Figura 2.9. El presidente Barack Obama anima a los estudiantes a programar sus teléfonos móviles (Code.org, 2015c)

Ya en un plano más operativo, ‘La Hora del Código’ pretende que, a lo largo de la semana en la que se extiende su celebración, en el mayor número posible de escuelas del mundo se lleve a cabo, al menos, una hora de introducción a la programación informática. Para ello, basta con que el centro educativo se registre en la web del evento¹¹¹, y elija según sus gustos e intereses de entre los múltiples tutoriales interactivos de introducción al ‘coding’ que se ofertan desde el portal de recursos¹¹²: muchas de las herramientas y recursos de aprendizaje del ‘coding’ que veremos en la taxonomía detallada del epígrafe 2.4.1, aparecen ahora aquí en una versión simplificada con el objetivo de captar la atención de los estudiantes a través de una actividad corta e introductoria de 1 hora, enganche que sirva como paso previo a un aprendizaje de mayor profundidad. Así, además de los tutoriales propios diseñados por Code.org¹¹³ (Figura 2.10), plataformas como Scratch¹¹⁴, Tynker¹¹⁵, Lightbot¹¹⁶ o Khan Academy¹¹⁷, entre muchas otras, tienen un tutorial reducido que se oferta durante ‘La Hora del Código’.

¹¹¹ <https://hourofcode.com>

¹¹² <https://code.org/learn>

¹¹³ <http://hourofcode.com/code>

¹¹⁴ <http://hourofcode.com/sc>

¹¹⁵ <http://hourofcode.com/ty>

¹¹⁶ <http://hourofcode.com/lb>

¹¹⁷ <http://hourofcode.com/kh>

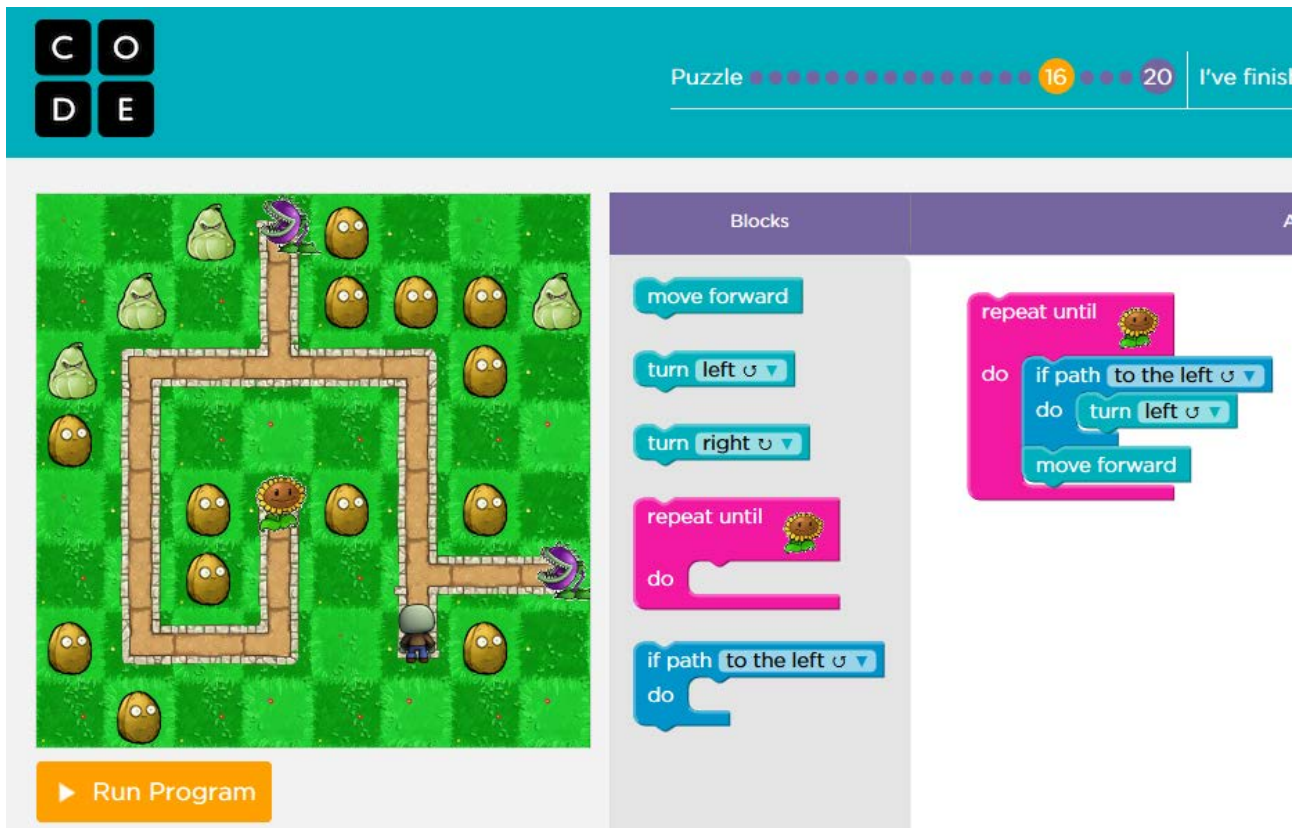


Figura 2.10. Captura de pantalla de 1 de los 20 retos de 'coding' que se proponen en el tutorial interactivo de 'La Hora del Código' diseñado por Code.org

Para finalizar con la descripción del evento, destacamos que, al final del mismo, el profesor puede generar fácilmente, de manera automatizada, certificados personalizados de participación para sus estudiantes, con el efecto motivacional que ello conlleva. Tras participar en 'La Hora del Código' el estudiante consigue así una recompensa que certifica simbólicamente que se ha iniciado en la disciplina del 'coding' (Figura 2.11), esto es, que ha comenzado su proceso de codigoalfabetización.



Figura 2.11. Plantilla personalizable para la certificación de participación el 'La Hora del Código' (Code.org, 2015c)

2.2.4. Delimitación final del problema de investigación

Tras los anteriores epígrafes, creemos que ha quedado suficientemente justificada la ampliación de nuestro foco de investigación: desde la población específica de sujetos de alta capacidad a la población escolar general; y desde la programación de aplicaciones para dispositivos móviles o ‘apps’, a la programación de cualquier objeto digital (animaciones, juegos, simulaciones, aplicaciones, etc...)

Es decir, nuestro foco de investigación ampliado se sitúa ahora sobre los procesos de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación en la población escolar general, y orientados a la creación de todo tipo de objetos digitales. Esto es, nuestro foco ampliado se sitúa sobre lo que hemos venido a denominar como ‘codigoalfabetización’.

En consecuencia, la formulación definitiva de nuestro problema general de investigación se enuncia así: ¿es posible y deseable incorporar la codigoalfabetización en el sistema educativo español? Su delimitación final sigue, de manera análoga, la estructura trazada inicialmente en el Capítulo 1 (Figura 1.1); pero ahora proyectada sobre un área de mayor alcance (Figura 2.12)

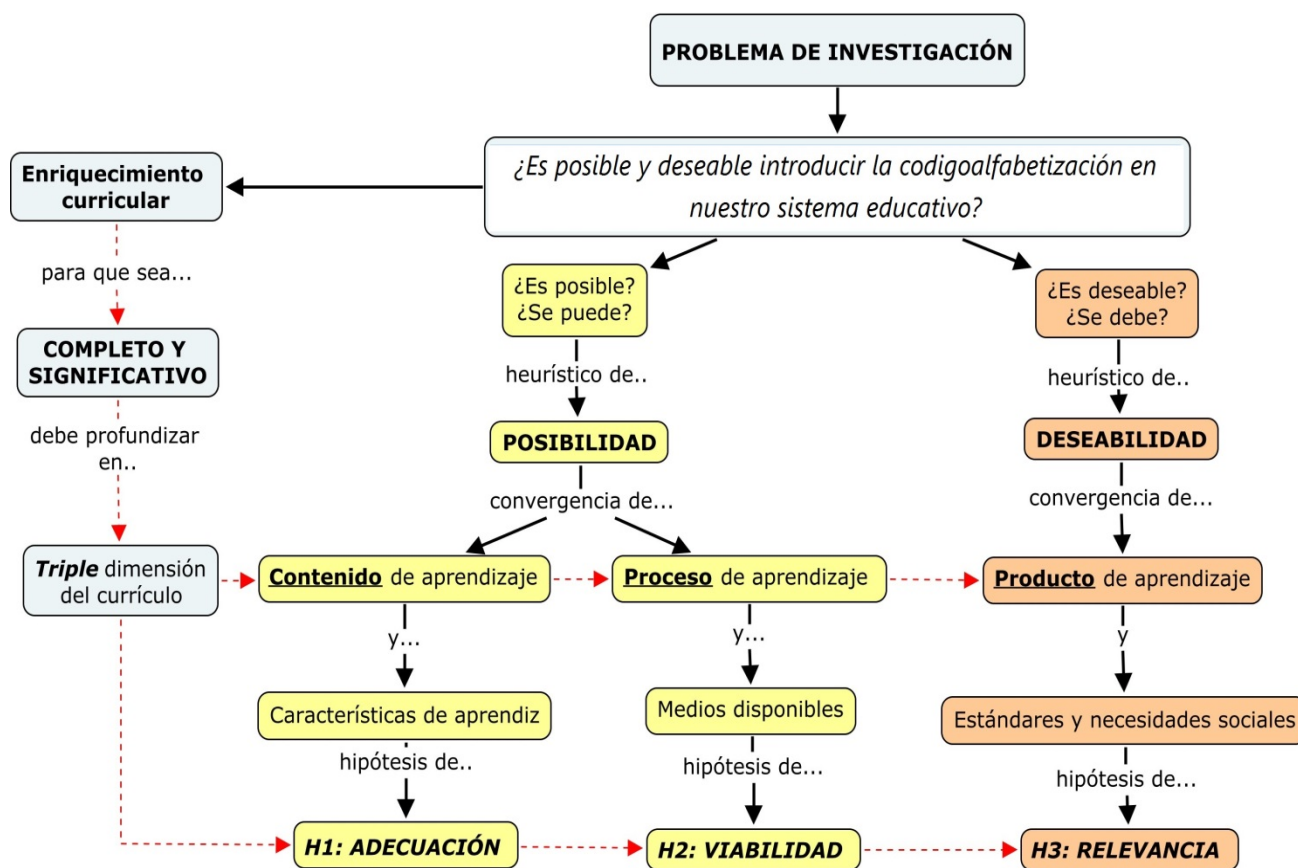


Figura 2.12. Delimitación final del problema de investigación

Pasaremos ahora, por tanto, a revisar nuestras hipótesis generales o directivas (*adecuación, viabilidad, y relevancia*) a la luz de nuestro definitivo problema de investigación, cuyo foco ha sido inevitablemente ampliado por la irrupción del código como el lenguaje maestro de nuestra sociedad digital.

2.3. Adecuación: el *porqué* de la codigoalfabetización

En primer lugar, abordamos la cuestión de si es *adecuado* incorporar la codigoalfabetización a nuestro sistema educativo; o el *porqué* de la codigoalfabetización. En otras palabras, si el *contenido* de aprendizaje que propone la codigoalfabetización, la lectoescritura con los lenguajes informáticos de programación, se ajusta a las características de los estudiantes y produce en ellos efectos positivos en distintos ámbitos.

Para ello revisaremos dos grupos de estudios bien diferenciados en el tiempo: un primer grupo de estudios *clásicos*, fechados en la década de los años 80 fundamentalmente, que se corresponden con una primera ‘edad de oro’ de investigación relativa a la introducción de la programación informática en las escuelas a través de lenguajes como BASIC¹¹⁸ o, principalmente, Logo (Papert, 1980); y un segundo grupo de estudios recientes, fechados a partir de 2008, que se corresponde con una nueva oleada de investigación a raíz de la eclosión de los lenguajes visuales de programación, como Scratch, y su progresiva introducción en las aulas.

2.3.1. Estudios clásicos sobre efectos del aprendizaje del ‘coding’

Aunque el término ‘codigoalfabetización’ (*code-literacy*) es de reciente cuño, el uso de la programación informática con fines educativos no es en absoluto nuevo. A finales de los años 60, Seymour Papert inventa el lenguaje informático de programación Logo, con el objetivo de introducir a los niños en el uso de las computadoras como instrumentos para el aprendizaje y la potenciación de la creatividad (Papert, 1980). Logo fue específicamente diseñado, no sólo como un lenguaje informático, sino principalmente como un entorno de aprendizaje:

“Logo ha sido diseñado para proporcionar un entorno en el que los aprendices determinen un problema a resolver, tomen decisiones, experimenten, prueben soluciones, y construyan lo que en realidad ya sabían” (Watt, 1982, p. 112)

Logo está basado en los presupuestos constructivistas de Piaget (1952), con quien Papert colaboró unos años en la Universidad de Ginebra (entre 1959 y 1963); es decir, que el aprendizaje consiste en un proceso de construcción activa de conocimiento por parte del sujeto, a partir de la interacción que éste establece entre el mundo y sus esquemas cognitivos previos. Pero Papert, a través del lenguaje Logo, va un paso más allá y propone pasar del *constructivismo* al *construccionismo*. La idea central del *construccionismo* es que la construcción de ideas en el ámbito simbólico-cognitivo que propone el *constructivismo*, debe acompañarse de manera simbiótica por la construcción de artefactos en el ámbito físico-real, a través de la manipulación activa del aprendiz. En el paso del *constructivismo* al *construccionismo* tuvieron un peso decisivo las aportaciones de Vygotsky relativas a cómo la actividad mental está mediada por las herramientas y lenguajes que utilizamos; o cómo el mundo externo perfila el funcionamiento del mundo interno, y viceversa (Vygotsky, 1978, 1986).

En términos más concretos, a través de Logo, los niños construyen-escriben programas que generan gráficos digitales, a través de la secuenciación de instrucciones textuales que dirigen el movimiento de una tortuga cibernética (*‘cybernetic turtle’*), representada por un pequeño puntero triangular en la

¹¹⁸ BASIC = *Beginner's All-purpose Symbolic Instruction Code*

pantalla (Figura 2.13). Según Clements (1986), podemos caracterizar a Logo como un entorno de aprendizaje interactivo (responde a las acciones del aprendiz), procedimental (está basado en secuencias de procedimiento), y extensible (no están predefinidos los programas que se pueden escribir, ni los gráficos que se generan en consecuencia).

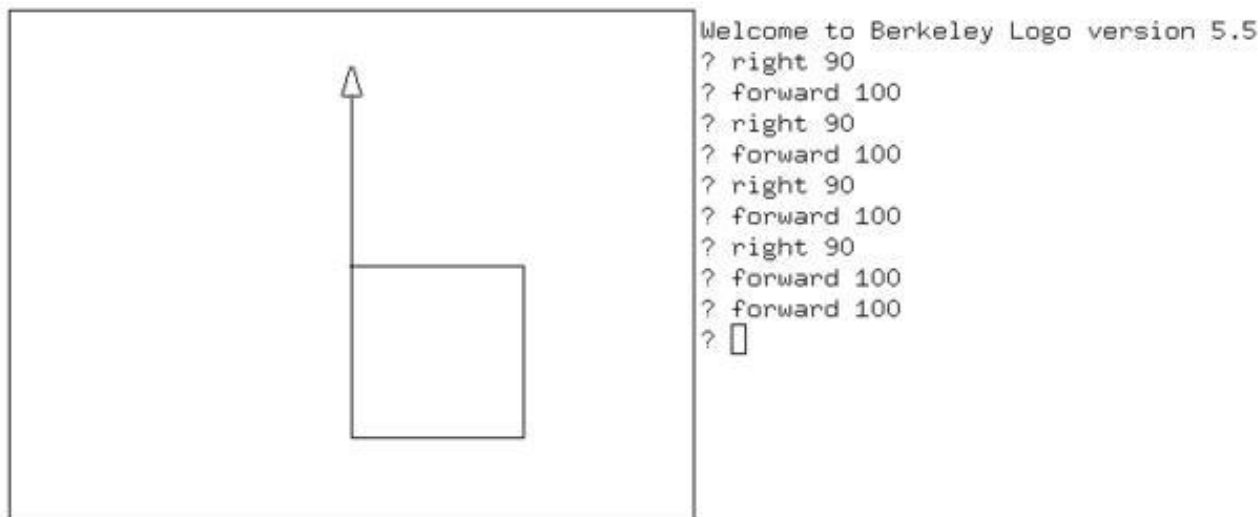


Figura 2.13. El lenguaje-entorno de programación informática Logo. A la derecha, el programa informático escrito por el aprendiz; a la izquierda, el gráfico resultante.

Papert (1980) defendía que, al programar con Logo, los niños deben reflexionar sobre qué tarea realizar, cómo hacerla y, en consecuencia, sobre sus mismos procesos de pensamiento. El proyecto Logo, enraizado igualmente en algunos presupuestos de la inteligencia artificial, otorga máxima importancia al hecho de que las personas aprendemos de nuestros errores, a través de un proceso de depuración (*'debugging'*) progresiva de los mismos. Este proceso de depuración de errores es central en la filosofía de Logo, que invita a los sujetos a aprender a través de sucesivas iteraciones de ensayo-error; en la asunción de que, al detectar sus errores, el sujeto realiza una reorganización de sus esquemas perceptuales y cognitivos.

En síntesis, el lenguaje Logo nació con la promesa de ser un entorno de aprendizaje efectivo para desarrollar habilidades cognitivas y metacognitivas (Clements, 1986). Esta promesa fue sometida a una intensa oleada de investigación empírica en los años sucesivos, cuyos resultados quedaron resumidos en algunas revisiones escritas a comienzos de los años 90 (Clements & Meredith, 1993; Clements & Sarama, 1997; Palumbo, 1990), y que se sintetizan en los próximos párrafos.

Así, en el ámbito de las matemáticas, existen estudios con niños de Primaria que muestran mejoras asociadas al uso de Logo en tareas como la clasificación de figuras (Battista & Clements, 1988), el pensamiento geométrico (Johnson-Gentile, Clements, & Battista, 1994), la conceptualización matemática de ángulos (Clements & Battista, 1989), o la medición de dimensiones lineales (Campbell, 1987). También encontramos otro grupo de estudios, de orientación piagetiana, que demuestra que el uso de Logo puede acelerar el paso desde el estadio *preoperacional* al *operacional* (Hines, 1983), apareciendo efectos positivos en tareas específicas de clasificación y seriación (Clements, 1986; Clements & Gullo, 1984). Sin embargo, parece que Logo sólo incrementa el rendimiento de los niños en el área matemática cuando las tareas que se evalúan comparten las

características geométricas y espaciales de dicho entorno de programación (Papert, Watt, diSessa, & Weir, 1979). Así, se encuentra igualmente investigación que no encuentra beneficios en el rendimiento matemático general después de un periodo de aprendizaje con Logo (Clements, 1986; Hamada, 1987; Olive, 1991).

Por otra parte, existe también cierta evidencia de que la programación con Logo puede incrementar la habilidad de resolución de problemas de los niños de Educación Primaria (Statz, 1974), específicamente en tareas que implican el aprendizaje y uso de reglas o algoritmos (Gorman & Bourne, 1983). En el otro sentido, cuando la habilidad de resolución de problemas es medida a través de tareas más generales (p.e. problemas de aritmética general extraídos de libros de texto), Logo parece no tener efectos (Clements, 1985).

Algunos de estos estudios demuestran además los efectos positivos de Logo sobre las habilidades metacognitivas de los niños (p.e., Clements, 1986; Clements & Gullo, 1984), especialmente en lo relativo a la automonitorización de la comprensión de sus tareas. Este hallazgo es consistente con la naturaleza del entorno Logo:

“En cierta manera, en el entorno de programación Logo los niños resuelven conscientemente problemas que se han auto-planteado, utilizando estrategias desconocidas. Deben comunicar su organización de la tarea y sus procesos de resolución a sí mismos, a los compañeros, al profesor, y a la máquina. Los niños deben analizar sus propios procesos de pensamiento a partir de los errores frecuentes que cometen con Logo, y dedicar tiempo a ello. Por lo tanto, parece probable que, programando con Logo, ocurran experiencias metacognitivas en la mente de los niños” (Clements, 1986, p. 310)

Pero las investigaciones sobre Logo también se focalizaron sobre otros ámbitos como el lingüístico. Así, se encuentran estudios que demuestran cómo Logo fomenta las interacciones conversacionales entre los niños (Genishi, McCollum, & Strand, 1985), incrementa las habilidades de lenguaje-perceptual (Lehrer & DeBernard, 1987), eleva las puntuaciones en vocabulario y comprensión oral en el SAT¹¹⁹ (Robinson, Gilley, & Uhlig, 1988), o mejora la capacidad de los niños para describir direcciones oralmente (Clements, 1986). Sin embargo, no se encuentran efectos de Logo sobre la habilidad lectora general (Clements, 1986), y se reconoce que a este cuerpo de investigación sobre el área lingüística le falta profundidad para ser concluyente (Clements & Sarama, 1997)

Finalmente, se sugieren mejoras en la creatividad de los alumnos a partir de su aprendizaje y uso continuado de Logo, lo cual implica una continua invención y reinención de formas geométricas. (Clements, 1986; Clements & Gullo, 1984).

Globalmente, el *corpus* de estudios clásicos sobre los efectos que tiene en los niños el aprendizaje de la programación a través de Logo, resulta algo decepcionante y no concluyente (Clements & Meredith, 1993; Clements & Sarama, 1997; Palumbo, 1990). Hay resultados contradictorios; y, cuando se encuentran efectos positivos, éstos son de un tamaño menor del esperado. Las mejoras más destacables se encuentran en tareas específicas de tipo espacial y geométrico similares a las que

¹¹⁹ SAT = Stanford Achievement Test

involucra el entorno Logo. Cuando se trata de transferir los aprendizajes de Logo a tareas de índole más general, esos efectos positivos desaparecen.

Así pues, la ambiciosa promesa implícita en Logo se vio truncada, y la programación informática desapareció de las aulas a principios de los años 90. Justo en ese momento, comienza el reinado de Microsoft con la eclosión del ordenador personal (que se introduce en el ámbito doméstico), los paquetes ofimáticos, como Microsoft Office, y la aparición de los primeros navegadores web (Netscape, Internet Explorer) o las herramientas de chateo ‘*on-line*’ (Messenger). Así, la desaparición de Logo a principios de los 90 dejó paso a una nueva concepción de la informática en las aulas vinculada al aprendizaje del uso de herramientas ofimáticas y de navegación web, orientadas a la comunicación y al manejo/gestión de información (el denominado ‘*ICT Curriculum*’ o ‘*Currículum TIC*’); que llega prácticamente hasta el momento presente.

Según Resnick *et al.* (2009), la desaparición de Logo, y otros lenguajes para niños de esa época como BASIC, pudo deberse fundamentalmente a 3 factores, adicionales a los resultados poco concluyentes del *corpus* de investigación empírica que hemos revisado:

- Dificultades de sintaxis: los lenguajes de programación como Logo o BASIC, aun estando diseñados para niños, estaban basados en comandos-órdenes de tipo textual, incluyendo algunas reglas de sintaxis que ciertos niños no llegaban a dominar; y que frustraban su aprendizaje.
- Actividades no significativas: muchas de las actividades de programación con Logo (p.e. escribir un programa que dibuja un determinado diseño geométrico), o con BASIC (p.e. escribir un programa que genera listas de números primos), estaban desvinculadas de los intereses y experiencias cotidianos de los niños.
- Programación en solitario: la programación con Logo o BASIC se producía frecuentemente en contextos solitarios, en los cuales el niño no podía obtener ayuda cuando las cosas iban mal, o felicitación y apoyo para profundizar más cuando las cosas iban bien.

Todas estas limitaciones han sido recientemente superadas por los lenguajes visuales de programación como Scratch (*visual + significativo + social*, tal y como dijimos algo más arriba), lanzado al mundo en mayo de 2007. Ello ha contribuido decisivamente al nacimiento del movimiento de *codigoalfabetización*, y ha generado una nueva oleada de investigación sobre los efectos positivos del ‘*coding*’ para los aprendices, que pasamos a describir en el próximo sub-epígrafe.

2.3.2. Estudios recientes sobre efectos del aprendizaje del ‘*coding*’

En este epígrafe revisamos una segunda oleada de estudios sobre los efectos del aprendizaje del ‘*coding*’. Son estudios recientes, fechados a partir del año 2008, coincidiendo con la eclosión de los nuevos lenguajes de programación para niños, que se caracterizan por ser: *visuales* (bajando las barreras de entrada a la programación que antaño suponían las sintaxis de lenguajes textuales como Logo y BASIC); *significativos* (permitiendo la creación de todo tipo de artefactos digitales, cercanos

a los intereses de niños y jóvenes); y *sociales* (porque los entornos de aprendizaje del ‘coding’ son ahora ‘on-line’, permitiendo el aprendizaje colaborativo del mismo).

Así pues, si Logo se inscribió en un marco de aprendizaje *construccionista*; los nuevos lenguajes, como Scratch, lo hacen en el emergente marco *conectivista* (Siemens, 2005, 2010). El *conectivismo* considera que las teorías previas del aprendizaje, como el constructivismo o el construccionismo, son insuficientes para explicar las nuevas formas de adquirir información y generar conocimiento que se dan en los entornos digitales ‘on line’. En síntesis, el *conectivismo* define el aprendizaje como un proceso de conexión que el sujeto establece entre diferentes fuentes de información y sus nodos, determinado por una variedad de puntos de vista y perspectivas, y dependiente de una pluralidad de opiniones. El mantenimiento y actualización de esta red de conexiones es la clave para el aprendizaje, desde el punto de vista *conectivista*.

Recientemente, Kafai y Burke (2014), en su influyente libro “*Connected Code: Why Children Need to Learn Programming*”, argumentan cómo la programación informática o ‘coding’ ha sufrido una transformación cualitativa a raíz de su aprendizaje en entornos visuales, significativos y conectados como Scratch; erigiéndose en una habilidad de primer orden para los niños digitales del Siglo XXI. En concreto, describen que el ‘coding’ se ha transformado en los siguientes ejes:

- ‘Desde el código en sí, hacia sus aplicaciones’ (*‘From code to applications’*): incidiendo en cómo los niños aprenden actualmente a programar, no como una disciplina abstracta-formal, sino como un medio para crear aplicaciones, animaciones o videojuegos.
- ‘Desde las herramientas de aprendizaje, hacia las comunidades de aprendizaje’ (*‘From tools to communities’*): destacando cómo el aprendizaje del ‘coding’ ha dejado de ser una tarea solitaria, dependiente de herramientas concretas; para pasar a ser una actividad social inscrita en comunidades de aprendizaje, que incluyen relaciones de pares y de mentoría entre iguales¹²⁰
- ‘Desde el cacharreo, hacia la remezcla’ (*‘From scratch to remix’*): haciendo notar las nuevas prácticas de programación a partir de la reutilización y remezcla de librerías de código escritas por terceros, y los desafíos éticos que ello conlleva.
- ‘Desde la pantalla, hacia los objetos tangibles’ (*‘From screens to tangibles’*): mostrando cómo el ‘coding’ se va extendiendo desde la construcción de artefactos digitales que se ejecutan en la pantalla, hacia su conexión con objetos tangibles como robots, impresoras 3D o *wearables*¹²¹ (Kafai & Vasudevan, 2015)

En este contexto, se está generando un nuevo *corpus* de investigación que indaga sobre el impacto educativo asociado al aprendizaje del ‘coding’ con los lenguajes visuales de programación, especialmente Scratch (Maloney *et al.*, 2010).

¹²⁰ No nos resistimos, en este punto, a referenciar una noticia de impacto que saltó a la prensa en los días en que se acababan de escribir estas líneas: “*Two Valley teen programmers have started a movement where kids teach other kids how to code*” (Bort, 2016)

¹²¹ ‘Wearables’ = ‘Dispositivos ponibles’, como los relojes inteligentes (*‘smartwatches’*) o las pulseras de actividad

Centrándonos en el ámbito matemático, y haciendo la revisión de la literatura de menor a mayor edad de los sujetos: Zavala, Gallardo y García-Ruiz (2013) muestran evidencia de que el aprendizaje del ‘coding’ mejoró el desempeño de niños de 3° de Primaria en tareas de identificación y comparación de números. Por su parte, Lewis y Shah (2012) informan de una correlación positiva, en niños de 5° de Primaria, entre el nivel de desempeño en retos o puzzles de programación (‘programming quizzes’) y sus calificaciones académicas en matemáticas. En niños de 6° de Primaria, se han demostrado mejoras significativas en pensamiento matemático tras la utilización de Scratch en el aula (Calao, Moreno-León, Correa, & Robles, 2015). En sujetos ‘middle school’ (equivalente a 6° Primaria, 1° y 2° ESO), se han encontrado actitudes significativamente más positivas hacia las matemáticas en las aulas donde se incorporó el ‘coding’ como parte de la asignatura (Ke, 2014). Finalmente, también en estudiantes ‘middle school’, Akpinar y Aslan (2015) informan de mejoras significativas en el aprendizaje de conceptos estadísticos y de probabilidad tras la introducción de Scratch en el aula como herramienta para la creación de videojuegos.

También encontramos investigación reciente sobre los efectos del aprendizaje del ‘coding’ en el ámbito lingüístico. Así, Sanjanaashree, Kumar y Soman (2014) describen cómo el uso de Scratch contribuyó a que los estudiantes aprendieran a construir frases sencillas en inglés (idioma extranjero para los sujetos del estudio). Burke y Kafai (2010) llevaron a cabo talleres extraescolares de escritura creativa con sujetos de secundaria, y encontraron que el aprendizaje de la escritura de código con Scratch tiene efectos positivos sobre las habilidades narrativas de los participantes, específicamente en lo que tiene que ver con la secuenciación, estructuración y claridad de sus narraciones. Finalmente, Burke (2012) concluye que el ‘coding’ con Scratch es un marco de aprendizaje que facilita y promueve las habilidades generales de composición digital de estudiantes ‘middle school’.

En el ámbito de las ciencias, se encuentran también estudios que informan de mejoras en el aprendizaje de las mismas a partir de la incorporación de Scratch en el aula, por ejemplo en 5° de Primaria (Lai & Lai, 2012). Por otro lado, nos encontramos con varios estudios que destacan las virtudes del ‘coding’ para la modelización y simulación de sistemas físicos o biológicos complejos; a través de los denominados ‘Multi-Agent-Based Computational Models’ (MABCM). Estos ‘Modelos Computacionales Multiagente’ son capaces de simular sistemas físicos o biológicos complejos a través de la programación-fijación de reglas sobre el conjunto de elementos o agentes que forman dicho sistema. Así, por ejemplo, Dickes y Sengupta (2013) informan de cómo estudiantes de 4° de Primaria fueron capaces de modelizar computacionalmente, a través del entorno de programación NetLogo¹²², el concepto biológico de ‘selección natural de especies’. También a través de NetLogo, Wilensky y Reisman (2006) reportan una experiencia con estudiantes ‘high school’ (3° ESO a 2° Bachillerato) que modelizaron un ecosistema biológico de depredación, específicamente de lobos frente a ovejas (Figura 2.14). En este mismo sentido, algunos investigadores de referencia de nuestro país (Olabe, Basogain, Olabe, Maíz, & Castaño, 2014) han llamado la atención de cómo la programación en entornos como NetLogo, Scratch o Snap¹²³, es especialmente indicada para potenciar en los estudiantes su capacidad de resolución de problemas Tipo-B (problemas complejos y no determinísticos, sin solución unívoca sino iterativa, como los relativos a sistemas físicos y

¹²² <https://ccl.northwestern.edu/netlogo/> [NetLogo es un entorno de programación específicamente diseñado para trabajar con ‘Modelos Computacionales Multiagente’]

¹²³ <http://snap.berkeley.edu/> [Snap es una versión avanzada de Scratch orientada a estudiantes ‘high school’ o incluso universitarios, que incluye bloques de código más complejos]

biológicos); según los autores, estos problemas Tipo-B serán progresivamente más relevantes en nuestra sociedad y deben incorporarse a los currículos educativos, que actualmente se limitan a abordar problemas Tipo-A (problemas sencillos y determinísticos, con solución unívoca, como los problemas de aritmética).

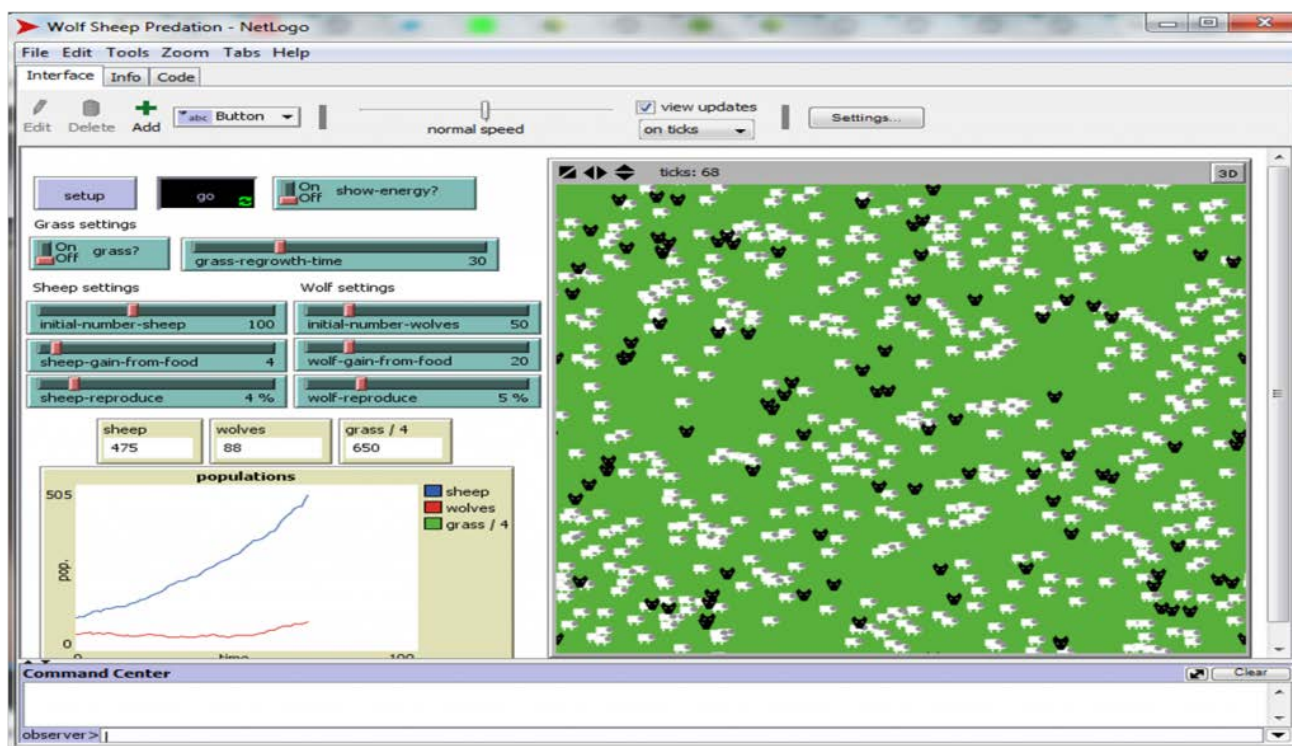


Figura 2.14. 'Modelo Computacional Multiagente' de depredación lobos-ovejas en NetLogo

Adicionalmente, se encuentra otro grupo de estudios recientes que se focaliza en cómo el aprendizaje del 'coding' contribuye a la adquisición de 'conceptos de programación' ('*programming concepts*'), o, en la terminología más reciente de Brennan & Resnick (2012), de 'conceptos computacionales' ('*computational concepts*'). Así, por ejemplo, Maloney *et al.* (2008) describen cómo sujetos de entre 8 y 18 años participando en un Club de Informática extraescolar, y utilizando Scratch como entorno de aprendizaje del 'coding', mejoraron en la adquisición de conceptos de programación como los bucles ('*loops*'), las sentencias condicionales ('*conditional statements*'), o el establecimiento de variables (en la Figura 2.15 se muestra un ejemplo de cómo estos conceptos se hayan inscritos en un programa escrito con Scratch); es de destacar que estas adquisiciones conceptuales se dieron sin intervención instructiva formal (los participantes en el Club aprendieron Scratch de manera informal a través de mentoría con pares más expertos). Otro estudio, en este caso centrado en sujetos '*middle school*' (6º Primaria a 2º ESO) y en el contexto de aula ordinaria, informa igualmente de la adquisición por parte de los estudiantes de conceptos de programación (bucles, condicionales, variables) a través del uso de Scratch (Meerbaum-Salant, Armoni, & Ben-Ari, 2010). Por su parte, Grover y Pea (2013b) informan sobre cómo introdujeron conceptos computacionales a sujetos '*middle school*' a través del aprendizaje de la programación de aplicaciones móviles o '*apps*' con App Inventor.¹²⁴

¹²⁴ <http://appinventor.mit.edu/explore/>

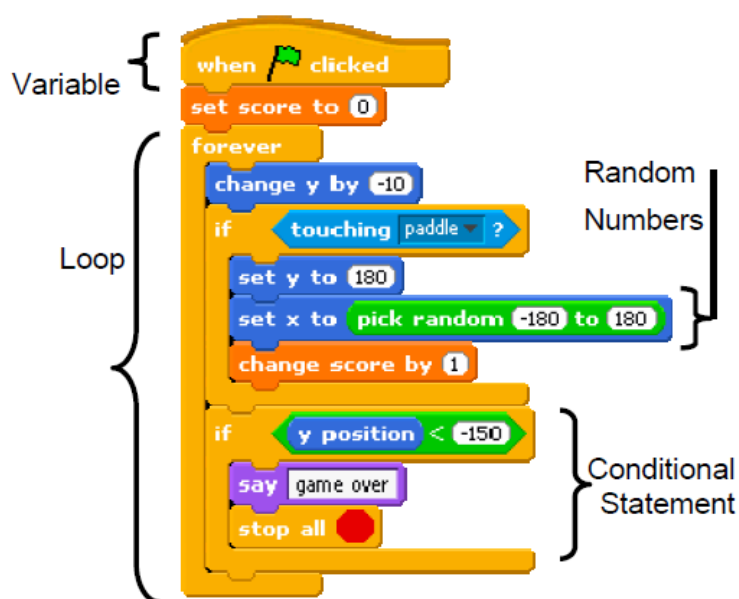


Figura 2.15. Conceptos de programación implícitos en un *script* de Scratch (Tomado de Maloney *et al.*, 2008)

Todavía más recientes, son los estudios que investigan en qué medida el aprendizaje del ‘coding’ contribuye a desarrollar en los estudiantes su ‘pensamiento computacional’ (PC), emergente constructo que se refiere esencialmente al proceso cognitivo de solución de problemas subyacente a las tareas de ‘coding’; y al que dedicamos el Capítulo 3 por completo. Una revisión profunda de este grupo de estudios puede encontrarse en Lye y Koh (2014), que detallaremos en el epígrafe 3.2.2.

En síntesis, podemos afirmar que el *corpus* reciente de estudios sobre los efectos del aprendizaje del ‘coding’ con los nuevos lenguajes visuales de programación muestra evidencia sobre mejoras en el ámbito matemático, en el ámbito lingüístico (en especial en las habilidades narrativas de secuenciación, estructuración y claridad expositiva), y en el ámbito de la modelización-simulación de sistemas físicos o biológicos complejos.

Sin embargo, aunque dichas investigaciones dibujan un panorama muy prometedor, muchos de los estudios son de tipo descriptivo y no siguen las recomendaciones básicas para la realización de investigación aplicada en educación (Cohen, Manion, & Morrison, 2007). Especialmente, parece necesario desarrollar más estudios empíricos de corte experimental o *cuasi-experimental*, que incluyan medidas *pretest* y *posttest* con grupos experimentales y de control, para ir consolidando algunas conclusiones en este campo. En esa línea, en el Capítulo 7, ya en la parte empírica de esta tesis doctoral, se presentará una evaluación del curso-programa ‘K-8 Intro to Computer Science’ (Code.org, 2015a) siguiendo un diseño *cuasi-experimental*.

2.4. Viabilidad: el cómo de la codigoalfabetización

Afrontamos ahora la cuestión de si es *viabile* incorporar la codigoalfabetización a nuestro sistema educativo; o el *cómo* de la codigoalfabetización. En otras palabras, si el *proceso* de aprendizaje que propone la codigoalfabetización, la lectoescritura con los lenguajes informáticos de programación, se puede abordar desde el conjunto de recursos, herramientas, sistemas y metodologías actualmente disponibles.

Para ello, dividiremos esta apartado en dos epígrafes. En el primero de ellos, se presenta una propuesta de taxonomía que trata de caracterizar y clasificar la gran cantidad existente de recursos para la enseñanza-aprendizaje del ‘coding’. En el segundo de ellos, con objeto de ilustrar la viabilidad real de la codigoalfabetización a nivel de los sistemas educativos formales, se presenta un análisis comparativo sobre el grado de incorporación del ‘coding’ a los currículos de distintos países de la Unión Europea; centrándonos especialmente en un ejemplo internacional, el currículo de Reino Unido, y en otro nacional, el currículo de la Comunidad de Madrid.

2.4.1. Taxonomía de recursos para la enseñanza-aprendizaje del ‘coding’

La taxonomía que detallamos a continuación es una versión mejorada y ampliada de un trabajo previo del doctorando presentado al *I Simposio Internacional de Mobile Learning*, celebrado en Córdoba en el mes de marzo de 2014 (Román-González, 2014c)

Según entendemos, los recursos de enseñanza-aprendizaje del ‘coding’ que podrían incorporarse a nuestro sistema educativo, pueden caracterizarse a través de los siguientes 4 criterios:

- a. **Tipo de lenguaje de programación (y edad de inicio de uso recomendada):** los recursos de enseñanza-aprendizaje del ‘coding’ están basados en alguno de los siguientes tres tipos de lenguajes de programación.
 - o Lenguajes visuales por flechas (*‘arrow-based programming languages’*): el aprendiz programa visualmente una secuencia de comandos a partir de flechas e iconos intuitivos similares. Estos lenguajes están diseñados para niños pre-lectores, y pueden comenzar a utilizarse a finales de la etapa de Educación Infantil o comienzos de la Educación Primaria. Algunos ejemplos de recursos que utilizan lenguajes visuales por flechas son Kodable¹²⁵ (Figura 2.16), Lightbot¹²⁶, Bee-Bot App¹²⁷, ScratchJr¹²⁸ o The Foos¹²⁹
 - o Lenguajes visuales por bloques (*‘block-based programming languages’*): el aprendiz programa visualmente una secuencia de comandos a través de bloques o piezas de código que encajan entre sí, y en las cuales puede fijar distintos parámetros. Estos lenguajes exigen ya habilidades lectoras, y suelen estar diseñados para edades intermedias (finales de Educación Primaria y comienzo de Educación Secundaria). Algunos ejemplos de recursos que utilizan lenguajes visuales por bloques son Scratch¹³⁰ (ya ilustrado anteriormente en la Figura 2.4), Tynker¹³¹ (Figura 2.17), Blockly¹³², Code.org¹³³ o Hopscotch¹³⁴

¹²⁵ <https://www.kodable.com/>

¹²⁶ <https://lightbot.com/>

¹²⁷ <https://itunes.apple.com/es/app/bee-bot/id500131639?mt=8>

¹²⁸ <http://www.scratchjr.org/>

¹²⁹ <http://thefoos.com/>

¹³⁰ <https://scratch.mit.edu/>

¹³¹ <https://www.tynker.com/>

¹³² <https://blockly-games.appspot.com/>

¹³³ <https://code.org/>

¹³⁴ <https://www.gethopscotch.com/>

- Lenguajes textuales (*'textual programming languages'*): El aprendiz debe teclear el código correspondiente al lenguaje de programación utilizado, respetando su sintaxis, para que se ejecute en pantalla (habitualmente en tiempo real). La edad recomendada para iniciarse con este tipo de lenguajes ronda finales de la Educación Secundaria y Bachillerato. Algunos ejemplos de recursos que utilizan lenguajes de programación textuales son Khan Academy¹³⁵ (ya ilustrado anteriormente en la Figura 1.2), Code Combat¹³⁶, Codecademy¹³⁷ (Figura 2.18), CodeHS¹³⁸ o Mozilla Webmaker¹³⁹

En las siguientes Figuras 2.16, 2.17, y 2.18 se ilustran los tres tipos de lenguajes:

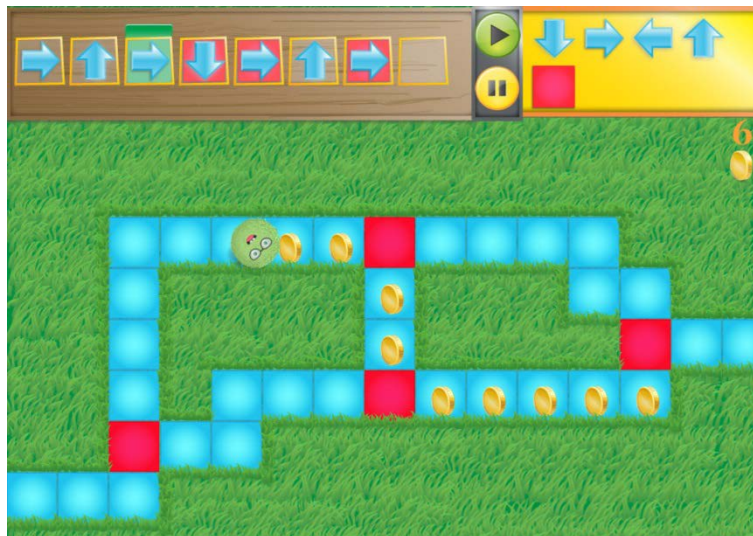


Figura 2.16. Ejemplo de lenguaje de programación visual por flechas (Tomado de Kodable)

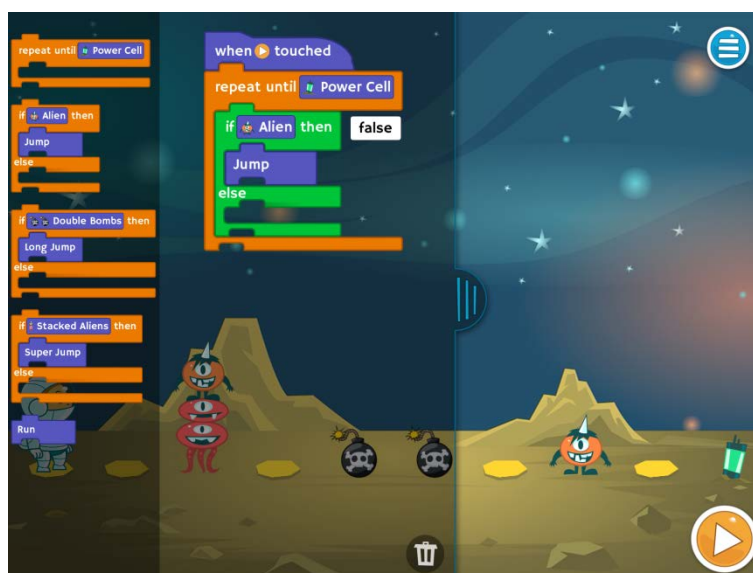


Figura 2.17. Ejemplo de lenguaje de programación visual por bloques (Tomado de Tynker)

¹³⁵ <https://www.khanacademy.org/computing/computer-programming>

¹³⁶ <https://codecombat.com/>

¹³⁷ <https://www.codecademy.com/>

¹³⁸ <https://codehs.com/>

¹³⁹ <https://teach.mozilla.org/tools/>

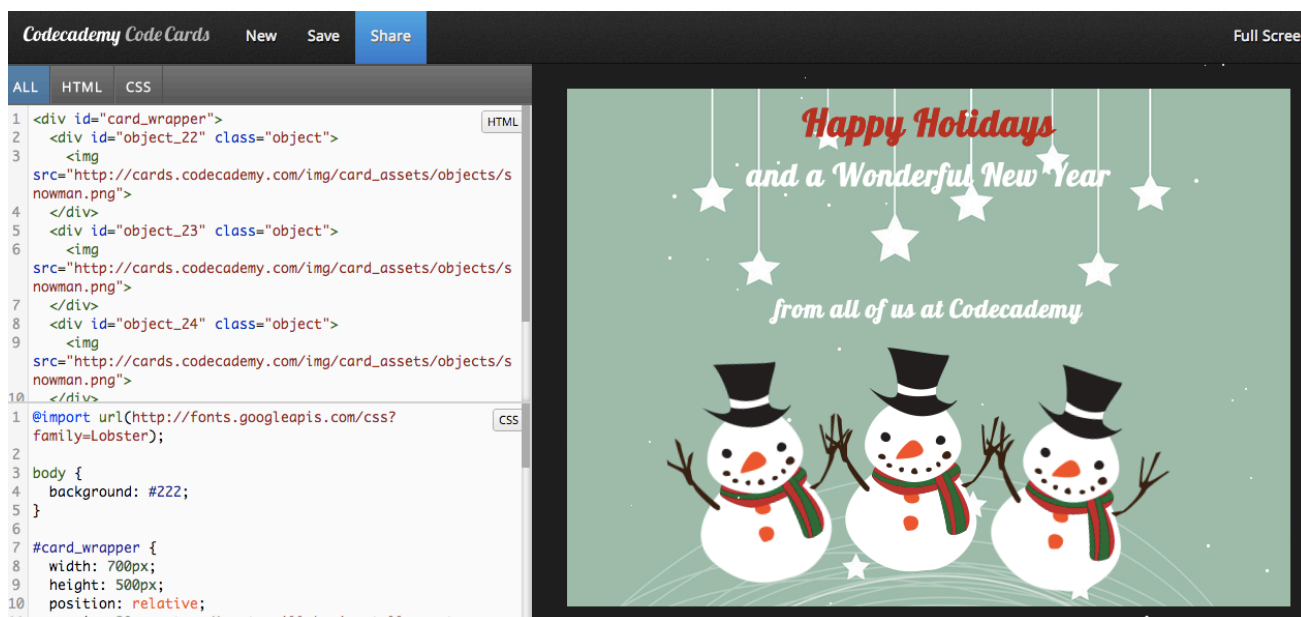


Figura 2.18. Ejemplo de lenguaje de programación textual (Tomado de Codecademy)

b. Orientación/Finalidad: podemos distinguir entre los distintos recursos de enseñanza-aprendizaje del ‘coding’, en función de si son entornos orientados a:

- Resolver problemas cerrados (‘close-ended’): el aprendiz escribe un programa que sirve para resolver un problema pre-definido. Estos entornos son también conocidos como retos o puzzles de programación (‘programming quizzes and puzzles’); y sus interfaces más habituales son del tipo ‘laberinto’ (en la que hay que programar el recorrido de una figura para llegar de un punto a otro del mismo) y tipo ‘lienzo o pizarra’ (en la que hay que programar un diseño geométrico previamente dado). En las Figuras 2.19 y 2.20 se ilustran ambos tipos de interfaces, tomadas de Blockly.

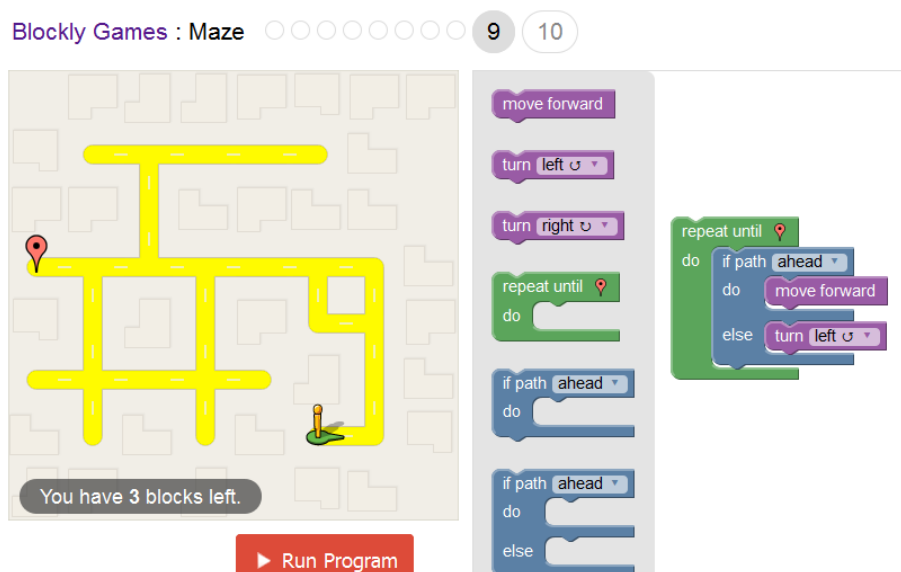


Figura 2.19. Ejemplo de una de las muchas interfaces tipo ‘laberinto’ existentes para el aprendizaje del ‘coding’ (tomado de <https://blockly-games.appspot.com/maze?lang=en&level=9>)

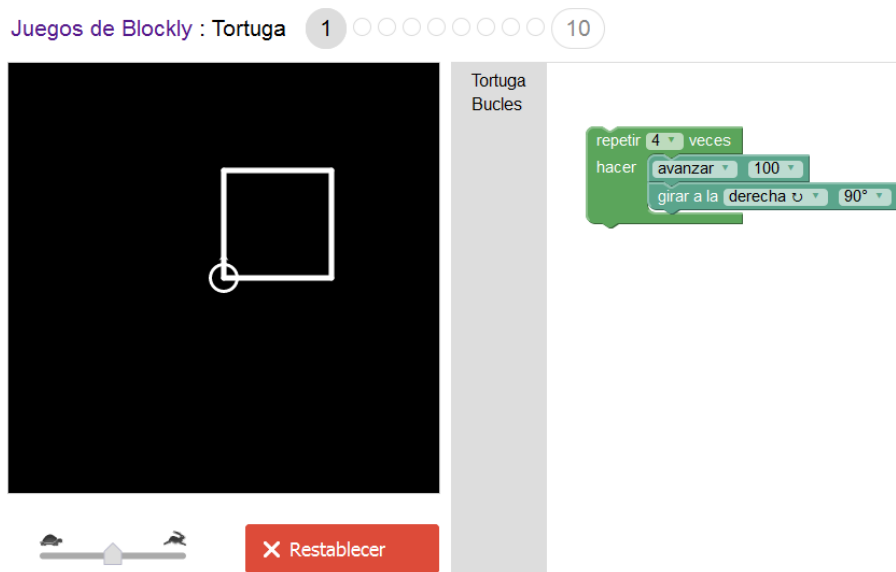


Figura 2.20. Ejemplo de una de las muchas interfaces tipo ‘lienzo o pizarra’ existentes para el aprendizaje del ‘coding’ (Tomado de <https://blockly-games.appspot.com/turtle?lang=es&level=1>)

- Crear/desarrollar proyectos abiertos (*‘open-ended’*): el ‘coding’ tiene como finalidad la creación de un nuevo objeto digital (bien sea escribiendo todo el programa desde cero, o remezclando partes de código de otros usuarios de la herramienta). Podemos distinguir algunos recursos en función de si los objetos digitales que permiten crear son: narraciones y animaciones digitales (p.e. a través de la herramienta Alice¹⁴⁰, ver Figura 2.21); simulaciones y modelizaciones computacionales (p.e. a través de NetLogo¹⁴¹, ya visto en la Figura 2.14, o Minecraft¹⁴²); aplicaciones (p.e. a través de App Inventor¹⁴³, ya ilustrado en la Figura 1.5, o App Lab¹⁴⁴); juegos (p.e. a través de Play my Code¹⁴⁵, Kodu¹⁴⁶, o Gamestar Mechanic¹⁴⁷); o diseños geométricos abiertos y sin definir previamente (p.e. a través de Move the Turtle¹⁴⁸). Un caso especial es Scratch (o su versión más avanzada Snap¹⁴⁹), que permite generar todos los tipos de objetos digitales citados anteriormente, razón por la cual se erige como un recurso especialmente versátil e indicado para el contexto escolar.
- Tener efectos sobre el mundo físico (*‘hardware-driven’*): el aprendiz escribe un programa que controla, y tiene efectos sobre, objetos del mundo físico. Dentro de esta categoría, podemos distinguir: juguetes-robots directamente programables por el aprendiz a través de un sistema de botones que incorpora el propio objeto, y que suelen estar diseñados para niños pre-lectores (p.e. el robot Bee-Bot¹⁵⁰, ver Figura

¹⁴⁰ <http://www.alice.org/index.php>

¹⁴¹ <https://ccl.northwestern.edu/netlogo/>

¹⁴² <https://minecraft.net/>

¹⁴³ <http://appinventor.mit.edu/explore/>

¹⁴⁴ <https://code.org/educate/applab>

¹⁴⁵ <http://www.playmycode.com/>

¹⁴⁶ <http://www.kodugamelab.com/>

¹⁴⁷ <https://gamestarmechanic.com/>

¹⁴⁸ <http://movetheturtle.com/>

¹⁴⁹ <http://snap.berkeley.edu/>

¹⁵⁰ <https://www.bee-bot.us/> [Ejemplo de uso de Bee-Bot en un aula de Infantil: <https://youtu.be/za6wHI50fJU>]

2.22, o el robot Cubetto¹⁵¹); robots o placas programables¹⁵² que son controlados desde el código escrito en un dispositivo digital, sea ordenador, tableta o *smartphone* (p.e. a través de Tickle¹⁵³ que se muestra en la Figura 2.23, S4A-Scratch for Arduino¹⁵⁴ que se muestra en la Figura 2.24, o Bitbloq¹⁵⁵); y entornos donde se escribe y comparte código que, a modo de planos de ingeniería, controla la impresión en 3D de objetos físicos (p.e. a través de la comunidad Thingiverse¹⁵⁶, que se ilustra en el Figura 2.25)

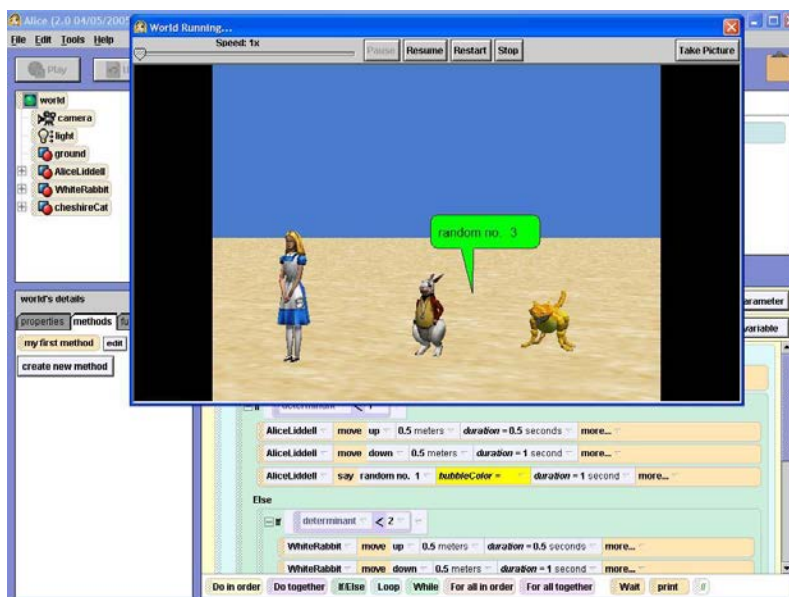


Figura 2.21. Ejemplo de 'coding' en un proyecto abierto de narración digital, a través de Alice



Figura 2.22. Ejemplo de 'coding' a través de la programación directa de robots Bee-Bots

¹⁵¹ <http://www.primotoys.com/cubetto>

¹⁵² Los ejemplos más claros de placas programables de hardware libre son actualmente Arduino (<https://www.arduino.cc/>) y Raspberry Pi (<https://www.raspberrypi.org/>)

¹⁵³ <https://tickleapp.com/>

¹⁵⁴ <http://s4a.cat/>

¹⁵⁵ <http://bitbloq.bq.com/>

¹⁵⁶ <http://www.thingiverse.com/>

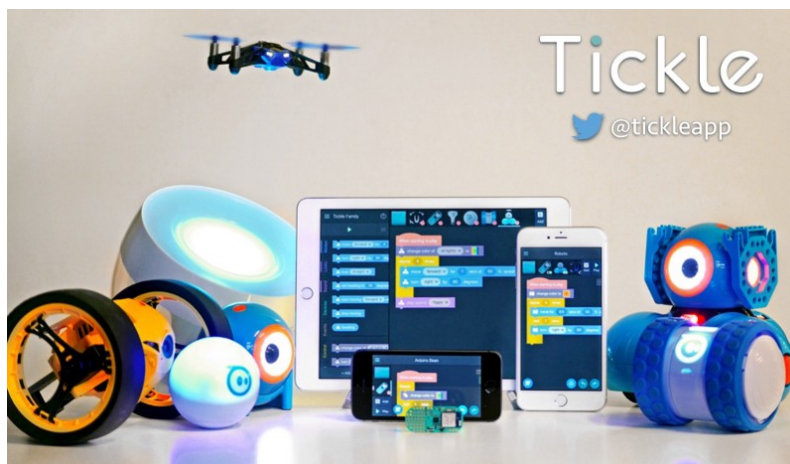


Figura 2.23. Ejemplo de 'coding' desde dispositivos móviles para controlar robots y drones, a través de Tickle

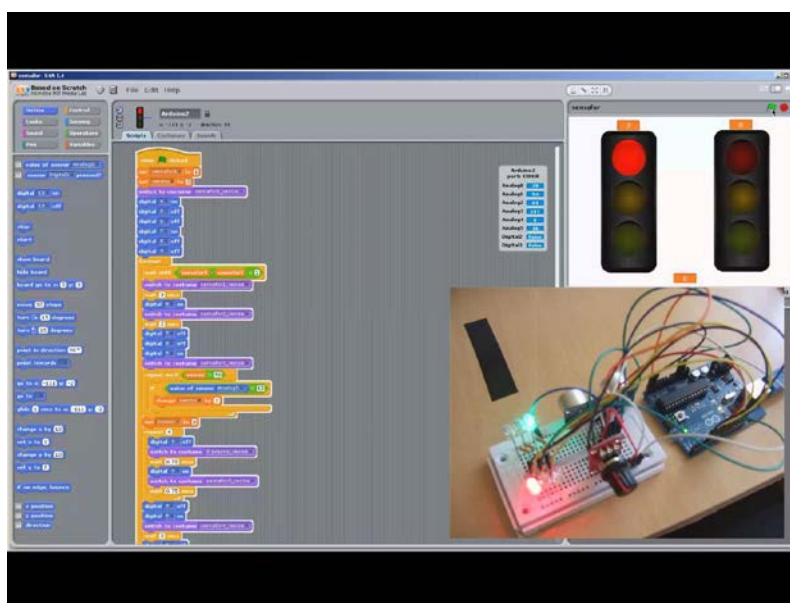


Figura 2.24. Ejemplo de 'coding' desde un ordenador para controlar una placa Arduino, a través de S4A

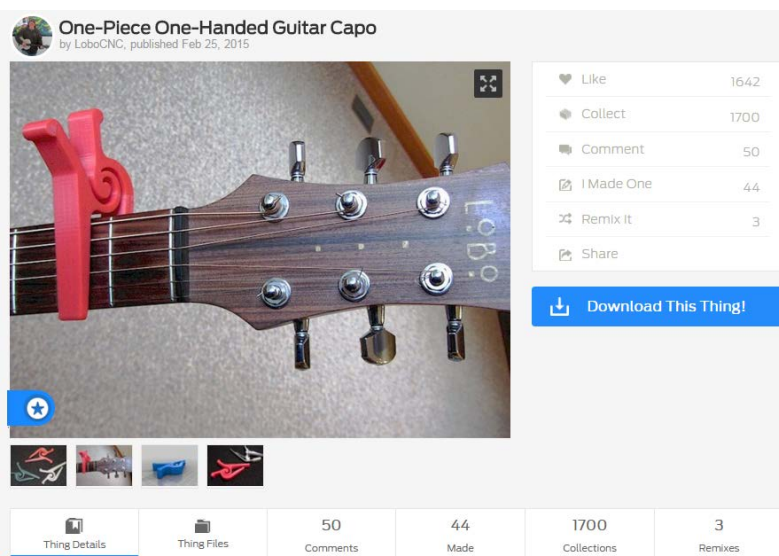


Figura 2.25. Cejilla para guitarra impresa en 3D, cuyos códigos se comparten y remezclan en Thingiverse

c. Plataforma: con arreglo a este criterio, podemos distinguir entre recursos para la enseñanza-aprendizaje del *'coding'* que:

- Deben ser descargados y ejecutados en local desde ordenadores tipo PC's. Este tipo de recursos es cada vez más escaso, dada su naturaleza *'off-line'*. Ejemplos de este tipo son los ya citados Alice, Kodu o NetLogo.
- Recursos *'web-based'* o *'browser-based'*, es decir, accesibles y ejecutables desde un navegador web con conexión a Internet. Aquí podríamos distinguir internamente entre recursos no navegables desde dispositivos móviles por estar basados en tecnología Flash Player, como Scratch; y otros sí navegables desde dispositivos móviles como, por ejemplo, los tutoriales de Code.org.
- Recursos tipo *'apps'* específicas para el aprendizaje del *'coding'*, pensadas y diseñadas para ser descargadas y ejecutadas desde dispositivos móviles, habitualmente tabletas. Ejemplos de este tipo son los ya citados Kodable o Hopscotch.

En cualquier caso, este criterio de clasificación es cada vez más difuso, y es muy habitual encontrar recursos que se ofrecen en distintos tipos de plataforma, retroalimentándose entre unas y otras. Por ejemplo, Tynker es un recurso originalmente del tipo *'browser-based'* que, ante la extensión y popularización de los dispositivos móviles, ha diseñado su propia *'app'*. A la inversa, la *'app'* Kodable también dispone de su propio sitio web en el que sus usuarios comparten manuales de uso de la misma, opiniones y mejoras.

d. Coste: encontramos desde recursos completamente gratuitos (habitualmente son recursos *'browser based'* como Khan Academy, Scratch o Code.org), a recursos de pago que ofrecen cursos completos de *'coding'* para ser implantados en el aula ofreciendo soporte técnico y pedagógico externo (p.e. Tynker), pasando por modelos *'freemium'* (habitualmente recursos tipo *'apps'* que se descargan inicialmente de manera gratuita para, posteriormente, exigir un pequeño pago por acceder a todas las funcionalidades de la aplicación, como por ejemplo la *'app'* de Lightbot)

En la siguiente Tabla 2.1 presentamos una recopilación de recursos para la enseñanza-aprendizaje del *'coding'*, clasificados con arreglo a los cuatro criterios que acabamos de explicitar. La lista que se aporta no pretende ser exhaustiva, dado el ingente volumen de recursos y herramientas para el *'coding'* que aparece prácticamente todas las semanas, pero sí representativa de los recursos que actualmente parecen tener más presencia e impacto en los ambientes educativos, tanto formales como no formales. Para su confección nos hemos guiado por artículos-listas influyentes de la blogosfera al respecto (Adam & Mowers, 2013¹⁵⁷; Crawley, 2014; Cronin, 2014; EdTech, 2014). Y, en cualquier caso, es un listado siempre provisional, susceptible de ser sometido a continua revisión, y que será sometido a investigación empírica en el Capítulo 5 de esta tesis.

¹⁵⁷ Para dar una idea al lector de la influencia de estas listas en la blogosfera, hacemos notar que, por ejemplo, ésta que referenciamos, "7 Apps for Teaching Children Coding Skills" (Adams & Mowers, 2013), publicada en el blog Edutopia, ha sido compartida en su comunidad de educadores más de 25.000 veces.

Tabla 2.1. Listado de recursos de enseñanza-aprendizaje del ‘coding’ clasificados según nuestra propuesta taxonómica

Nombre del recurso (por orden alfabético)	Tipo de Lenguaje de programación	Orientación / Finalidad	Plataforma	Coste
<i>Alice</i>	Visual por bloques	‘Open-ended’: Narrativa digital	Descarga en PC’s local	Gratuito
<i>App Inventor</i>	Visual por bloques	‘Open-ended’: Diseño de ‘apps’	‘Web-based’ (Navegable desde dispositivo móvil)	Gratuito
<i>App Lab</i>	Visual por bloques	‘Open-ended’: Diseño de ‘apps’	‘Web-based’ (Navegable desde dispositivo móvil)	Gratuito
<i>Bee-Bot App</i>	Visual por flechas	‘Close-ended’: Laberintos	‘App’ (iOS)	Gratuito
<i>Bee-bot Robots</i>	Visual por flechas	‘Hardware-driven’: Robots directamente programables	-	89,95 \$ por unidad
<i>Bitbloq</i>	Visual por bloques	‘Hardware-driven’: Programación de robots o placas desde pantalla	Descarga en PC’s local	Gratuito
<i>Blockly</i>	Visual por bloques	‘Close-ended’: Laberintos y lienzos	‘Web-based’ (Navegable desde dispositivo móvil)	Gratuito
<i>Cargo-Bot</i> ¹⁵⁸	Visual por flechas	‘Close-ended’: Puzles de programación	‘App’ (iOS)	Gratuito
<i>Code Combat</i>	Textual	‘Close-ended’: Puzles de programación en forma de aventura gráfica	‘Web-based’ (Navegable con dificultad desde dispositivo móvil)	Gratuito
<i>Code.org</i>	Visual por bloques	‘Close ended’ ¹⁵⁹ : Laberintos y lienzos	‘Web-based’ (Navegable desde dispositivo móvil)	Gratuito
<i>Codecademy</i>	Textual	Itinerarios de aprendizaje que combinan tareas ‘close-ended’ y ‘open-ended’	‘Web-based’ (Navegable con dificultad desde dispositivo móvil)	Gratuito
<i>CodeHS</i>	Textual	Itinerarios de aprendizaje que combinan tareas ‘close-ended’ y ‘open-ended’	‘Web-based’ (Navegable con dificultad desde dispositivo móvil)	Gratuito
<i>Cubetto</i>	Visual por flechas	‘Hardware-driven’: Robots directamente programables	-	£170 por unidad
<i>Daisy the Dinosaur</i> ¹⁶⁰	Visual por bloques	‘Open-ended’: Animaciones digitales	‘App’ (iOS)	Gratuito
<i>Hopscotch</i>	Visual por bloques	‘Open-ended’: Animaciones digitales	‘App’ (iOS)	Gratuito
<i>Khan Academy</i>	Textual	Itinerarios de aprendizaje que combinan tareas ‘close-ended’ y ‘open-ended’	‘Web-based’ (Navegable con dificultad desde dispositivo móvil)	Gratuito

¹⁵⁸ <http://twolivesleft.com/CargoBot/>

¹⁵⁹ Code.org está incorporando progresivamente a su plataforma entornos relativamente ‘open-ended’ como PlayLab: <https://code.org/playlab>

¹⁶⁰ <https://itunes.apple.com/es/app/daisy-the-dinosaur/id490514278?mt=8>

Nombre del recurso (por orden alfabético)	Tipo de Lenguaje de programación	Orientación / Finalidad	Plataforma	Coste
<i>Kodable</i>	Visual por flechas	'Close-ended': Laberintos	'App' (iOS)	5,99 € (versión completa)
<i>Kodu</i>	Visual por bloques	'Open-ended': Videojuegos	Descarga en PC's local	Gratuito
<i>Lego Mindstorms (Fix the Factory)</i> ¹⁶¹	Visual por flechas	'Close-ended': Laberintos	'App' (iOS y Android)	Gratuito
<i>Lightbot</i>	Visual por flechas	'Close-ended': Laberintos	'Web-based' y 'App' (iOS y Android)	2,99 € (versión completa)
<i>Move the Turtle</i>	Visual por bloques	'Open-ended': Diseño geométrico	'App' (iOS)	2,99 €
<i>Mozilla Webmaker</i>	Textual	Itinerarios de aprendizaje que combinan tareas 'close-ended' y 'open-ended'	'Web-based' (Navegable con dificultad desde dispositivo móvil)	Gratuito
<i>My Robot Friend</i> ¹⁶²	Visual por flechas	'Close-ended': Puzzles de programación	'App' (iOS)	3,59 €
<i>NetLogo</i>	Textual	'Open-ended': Simulaciones y modelización	Descarga en PC's local	Gratuito
<i>Play my Code</i>	Textual	'Open-ended': Videojuegos	'Web-based' (Navegable con dificultad desde dispositivo móvil)	Gratuito
<i>S4A – Scratch for Arduino</i>	Visual por bloques	'Hardware-driven': Programación de robots o placas desde pantalla	'Web-based' (No navegable desde dispositivo móvil)	Gratuito
<i>Scratch</i>	Visual por bloques	'Open-ended': Animaciones, narraciones, simulaciones, videojuegos	'Web-based' (No navegable desde dispositivo móvil)	Gratuito
<i>ScratchJr</i>	Visual por flechas	'Open-ended': Animaciones digitales	'App' (iOS y Android)	Gratuito
<i>Snap</i>	Visual por bloques	'Open-ended': Animaciones, narraciones, simulaciones, videojuegos	'Web-based' (No navegable desde dispositivo móvil)	Gratuito
<i>The Foos</i>	Visual por flechas	'Close-ended': Puzzles de programación en forma de aventura gráfica	'Web-based' y 'App' (iOS y Android)	Gratuito
<i>Tickle</i>	Visual por bloques	'Hardware-driven': Programación de robots o placas desde pantalla	'App' (iOS)	Gratuito
<i>Tynker</i>	Visual por bloques	Itinerarios de aprendizaje que combinan tareas 'close-ended' y 'open-ended'	Web-based' y 'App' (iOS y Android)	Freemium

Fuera de la anterior taxonomía podemos situar algunos otros recursos peculiares para el aprendizaje del 'coding' como los juegos de mesa Robot Turtles¹⁶³ y Code Monkey Island¹⁶⁴.

¹⁶¹ <https://itunes.apple.com/es/app/lego-mindstorms-fix-factory/id671493323?mt=8>

¹⁶² <https://www.commonsemmedia.org/app-reviews/my-robot-friend>

2.4.2. Currículos educativos para la enseñanza-aprendizaje del ‘coding’

En este epígrafe realizaremos un breve análisis comparativo sobre la situación del ‘coding’ en los currículos de los sistemas educativos de la Unión Europea. Para ello, nos apoyaremos fundamentalmente en el reciente informe de la Comisión Europea (octubre de 2015), denominado “*Computing our future. Computer programming and coding*¹⁶⁵: *Priorities, School curricula and initiatives across Europe*” (European Schoolnet, 2015). En la exposición de motivos de dicho informe, se dice:

“Las competencias y habilidades digitales son una de las principales condiciones para que la transformación digital de Europa sea un éxito, así como para su crecimiento y el bienestar de sus ciudadanos y sociedades (...) El reto para el sector educativo es elevar el nivel de dichas habilidades digitales en la futura fuerza de trabajo; pero, aún más importante, empoderar a la gente joven con competencias que les permitan dominar y crear sus propias tecnologías digitales, y prosperar en la sociedad actual. Creemos que la enseñanza-aprendizaje del ‘coding’, tanto en contextos formales como no formales, jugará un papel fundamental en este proceso (...) Muchos educadores, así como padres, economistas y políticos en Europa y el resto del mundo, comienzan a pensar que los estudiantes necesitan unas mínimas habilidades de computación y programación. Una razón es la escasez de profesionales empleables con habilidades digitales de orden superior. Para el año 2020 se calcula que Europa podría sufrir un déficit de alrededor de 800.000 profesionales capacitados en computación y programación. Otra razón importante es que el ‘coding’ ayuda a entender la sociedad digitalizada actual y fomenta habilidades fundamentales del siglo XXI como la resolución de problemas, la creatividad, y el pensamiento lógico” (European Schoolnet, 2015, p. 4-6)

Los resultados que presenta el informe se basan en una encuesta realizada a 21 Ministerios de Educación (correspondientes a 20 países europeos, más Israel), en la cual se les interroga sobre la situación del ‘coding’ en su respectivos sistemas educativos. Los países participantes en el estudio son, por orden alfabético: Austria (AT), Bélgica-Flandes (BE-NL), Bélgica-Valonia (BE-FR), Bulgaria (BG), Dinamarca (DK), Eslovaquia (SK), España (ES), Estonia (EE), Finlandia (FI), Francia (FR), Hungría (HU), Irlanda (IE), Israel (IL), Lituania (LT), Malta (MT), Noruega (NO), Países Bajos (NL), Polonia (PL), Portugal (PT), Reino Unido (UK) y República Checa (CZ).

A la pregunta, ¿qué países incluyen actual y formalmente el ‘coding’ en su currículum?, los resultados principales son:

- 16 países incluyen el ‘coding’ en su currículum, ya sea a nivel nacional, regional o local: Austria, Bulgaria, Dinamarca, Eslovaquia, España, Estonia, Francia, Hungría, Irlanda, Israel, Lituania, Malta, Polonia, Portugal, Reino Unido y República Checa.

¹⁶³ <http://www.robotturtles.com/>

¹⁶⁴ <http://codemonkeyplanet.com/>

¹⁶⁵ En dicho informe se dice explícitamente que los términos ‘computer programming’ y ‘coding’ se utilizan de manera indistinta.

- Bélgica-Flandes y Finlandia tienen planes de incluir el *'coding'* en su currículum en un futuro próximo (p.e. Finlandia ha incluido el *'coding'* entre los contenidos nucleares del nuevo currículum que entra en vigor en el curso 2016/2017)
- Bélgica-Valonia, Noruega y los Países Bajos no han integrado aún el *'coding'* en su currículum ni tienen planes para ello en un futuro próximo.

En la Tabla 2.2 se muestran los 16 países que ya incorporan, en alguna medida, el *'coding'* en su currículum; más los 2 países que tienen previsto hacerlo en un futuro inmediato. En la tabla se indica, además, en qué nivel del sistema (nacional, regional o local) se haya incluido el *'coding'*, y cuál fue el año en el que se inició su implantación curricular real.

Tabla 2.2. Relación de países europeos (más Israel) que incluyen el *'coding'* en su currículum actual o inmediato

	Nivel nacional	Nivel regional	Nivel local/escolar	Año de implantación
Austria	*			N/A ¹⁶⁶
Bélgica-Flandes		*		N/A
Bulgaria	*			N/A
Dinamarca	*			2014
Eslovaquia	*		*	1990
España	*	*		2015
Estonia	*		*	N/A
Finlandia	*	*	*	2016
Francia	*			2015
Hungría	*			1995
Irlanda	*		*	2014
Israel	*			1976
Lituania	*		*	1986
Malta	*			1997
Polonia	*			1985
Portugal	*			2012
Reino Unido	*			2014
República Checa			*	N/A

Tal y como puede observarse en la tabla anterior, Israel fue el primer país en incorporar la programación informática a su currículum, en el año 1976. Los antiguos países comunistas del este de Europa, fueron los siguientes en dicha incorporación: Lituania y Polonia en los años 80, y Eslovaquia y Hungría en los años 90. Por su parte, países como Dinamarca, Irlanda y Reino Unido, tienen una larga tradición en la integración de las TIC en su currículum¹⁶⁷, pero no ha sido hasta el año 2014 cuando han reformulado y conseguido implantar nuevos planes educativos en materia de

¹⁶⁶ N/A: *Not available* (No disponible)

¹⁶⁷ Recordamos que estos currículos TIC 'clásicos' tienen que ver con usos más pasivos de la tecnología como el aprendizaje de herramientas ofimáticas y la navegación web.

tecnología digital con orientación decidida hacia el *'coding'*. Por su parte, España y Francia, por primera vez y en el año 2015, han incluido el *'coding'* en su currículum educativo en alguna medida.

El caso de España es un tanto particular (así se hace notar en el propio informe de la Comisión Europea). En nuestro país existe una regulación a nivel nacional, que luego puede ser reformulada y concretada en cada Comunidad Autónoma de manera propia. En síntesis, la situación actual del *'coding'* en nuestro sistema educativo es la siguiente:

- A nivel nacional, el *'coding'* forma parte de la asignatura optativa “Tecnologías de la Información y la Comunicación I” de la Etapa de Bachillerato.
- A nivel autonómico, nos encontramos con 3 Comunidades Autónomas que incluyen carga adicional en su currículum en materia de *'coding'*:
 - Cataluña: en segundo ciclo de la ESO, el *'coding'* se oferta como parte central de una asignatura optativa de “*Informática*”, muy focalizada al diseño de aplicaciones para dispositivos móviles¹⁶⁸ o *'apps'*
 - Comunidad de Madrid: en 1º, 2º y 3º de la ESO, el *'coding'* forma parte explícitamente de la nueva asignatura obligatoria “*Tecnología, Programación y Robótica*”. En Educación Primaria se permite, y anima, a los colegios a ofertar una asignatura optativa de programación, que se cursaría fuera del horario escolar.
 - Comunidad Foral de Navarra: en 4º y 5º de Primaria se han incluido contenidos obligatorios de *'coding'*, no como asignatura separada sino transversalmente al área de matemáticas.
- Finalmente, hay que señalar el caso de la Comunidad Valenciana que, aunque no incluye formalmente contenidos de *'coding'* en el currículum de su asignatura optativa de “*Informática*” de primer ciclo de la ESO; tiene un cuerpo de profesorado con titulación específica en dicha disciplina (al menos 1 profesor de este tipo por centro), lo que viene promoviendo que, *de facto*, en la citada asignatura se estén impartiendo algunos contenidos de *'coding'*. Ello se hará evidente en la parte empírica de la tesis.

Siguiendo con el informe de la Comisión Europea, a la pregunta, ¿por qué incorporar el *'coding'* en el currículum?, los resultados principales son:

- La mayoría de los países tienen como objetivo desarrollar en los estudiantes las habilidades relativas al pensamiento lógico y la resolución de problemas (14 países). En más de la mitad de los países, concretamente en 11 de ellos, se focalizan explícitamente en el desarrollo de la habilidad de programar. También 11 países declaran su intención de atraer más sujetos hacia el campo de las Ciencias de la Computación a la hora de cursar estudios universitarios. Y finalmente 8 países manifiestan abiertamente su objetivo de fomentar la empleabilidad de sus estudiantes en el sector de la economía digital. Se detalla en la Tabla 2.3

¹⁶⁸ A este hecho ha contribuido decisivamente que en Barcelona se venga celebrando cada año el mayor congreso mundial sobre tecnologías móviles (*'Mobile World Congress'*), tal y como aparece en prensa (p.e. [Vallespín, 2014](#))

Tabla 2.3. Razones por las cuales incorporar el ‘coding’ en el currículo, por países

	Fomentar el pensamiento lógico	Fomentar la resolución de problemas	Atraer hacia estudios universitarios en Informática	Fomentar habilidades de programación	Fomentar la empleabilidad en el sector digital	Fomentar otras competencias clave
Austria	*	*	*	*	*	*
Bélgica-Flandes			*		*	*
Bulgaria	*	*	*	*		
Dinamarca	*	*				*
Eslovaquia	*	*				
España	*	*		*		*
Estonia	*	*	*			*
Finlandia	*	*		*		
Francia			*		*	*
Hungría	*	*				
Irlanda	*	*	*	*		*
Israel	*	*	*	*	*	*
Lituania	*			*		
Malta			*	*		
Polonia	*	*	*	*	*	*
Portugal	*	*			*	*
Reino Unido	*	*	*	*	*	
República Checa	*	*	*	*	*	*

Con respecto a la pregunta, ¿en qué etapa/s del sistema educativo se incorpora el ‘coding’ al currículo?; en síntesis, el ‘coding’ está incorporado principalmente en la etapa de Educación Secundaria (13 países), aunque se está incrementando el número de países que lo incorporan también en la etapa de Primaria (10 países). Destacan los casos de Estonia, Israel y Eslovaquia, que ya incorporan el ‘coding’ a lo largo de todas las etapas escolares (pre-universitarias) de su sistema educativo formal; y el caso de Polonia, que hará lo propio a partir de septiembre de 2016, al dar entrada a su nuevo currículum actualizado en Ciencias de la Computación.

Con respecto a si el ‘coding’ se imparte como asignatura separada (12 países) o como parte de una asignatura más amplia de Informática/Tecnología/TIC (13 países), se encuentra una gran variabilidad al respecto en función de países, pero también internamente en cada uno de ellos en función de regiones y de la etapa educativa a la que nos refiramos. Destaca un creciente enfoque de integración transversal (‘cross-curricular’) del ‘coding’ en otras asignaturas, principalmente en matemáticas. Este enfoque está presente, por ejemplo, en Dinamarca, Eslovaquia, España (caso de Navarra en las matemáticas de 4º y 5º de Primaria), Estonia, Finlandia (que será el primer país, en 2016, en aplicar un enfoque ‘cross-curricular’ puro al ‘coding’ en todas sus etapas y asignaturas), y Francia.

Finalmente, por su especial relevancia (Valverde, Fernández, & Garrido, 2015), analizaremos con algo más de detalle 2 currículos en particular: uno a modo de ejemplo internacional, el currículo de Reino Unido; y otro como ejemplo nacional, el currículo de la Comunidad de Madrid.

2.4.2.1. *Un ejemplo internacional: el currículo de Reino Unido*

El currículo oficial del Reino Unido ha introducido recientemente, en septiembre de 2014, una nueva asignatura denominada “*Computing*” ([UK Department of Education, 2013](#)), que ha venido a sustituir a la anterior asignatura “*ICT: Information and Communication Technology*”. La asignatura “*Computing*” es de carácter obligatorio, y se imparte desde los 5 a los 16 años. Su implantación ha supuesto un giro copernicano en la forma de aproximarse a la tecnología digital desde el sistema educativo británico; un cambio de perspectiva que ha trascendido el mero debate pedagógico especializado, y que se ha convertido en prioridad nacional, llenando ríos de tinta en los medios de comunicación de las islas ([Cuthbertson, 2014](#); [Dredge, 2014](#); [Morrison, 2014](#); [Pozniak, 2014](#)). La nueva asignatura “*Computing*” dice en su exposición de motivos:

“Una educación en computación de alta calidad debe preparar a los alumnos para usar el pensamiento computacional y la creatividad, con objeto de comprender y cambiar el mundo. La computación tiene profundos vínculos con las matemáticas, las ciencias, el diseño y la tecnología, y proporciona visión y conocimiento tanto de sistemas naturales como artificiales. El núcleo de “Computing” son las Ciencias de las Computación, a través de las cuales los alumnos aprenden los principios de la información y la computación, cómo funcionan los sistemas digitales y cómo aplicar este conocimiento a través de la programación. A partir de este conocimiento computacional, los alumnos estarán preparados para utilizar la tecnología para crear programas, sistemas y todo tipo de contenido. “Computing” también asegura que los alumnos lleguen a estar digitalmente alfabetizados, capaces de usar las tecnologías digitales y de expresarse y desarrollar sus ideas a través de las mismas; a un nivel adecuado para las futuras demandas de los puestos de trabajo y para ser participantes activos de un mundo digital” ([UK Department of Education, 2013, en línea](#))

La asignatura “*Computing*” tiene como objetivo general garantizar que los estudiantes:

- ✓ Puedan comprender y aplicar los principios y conceptos fundamentales de las Ciencias de la Computación; incluyendo la abstracción, la lógica, los algoritmos y la representación de los datos-información.
- ✓ Puedan analizar los problemas en términos computacionales, y tengan experiencia práctica en lectoescritura de programas informáticos con objeto de resolver dichos problemas.
- ✓ Puedan evaluar y aplicar las tecnologías digitales de la información, incluidas tecnologías emergentes (nuevas o desconocidas), analíticamente para resolver problemas.
- ✓ Sean usuarios responsables, competentes, seguros y creativos de las tecnologías digitales de la información y la comunicación.

Así pues, la nueva asignatura “*Computing*” gira esencialmente sobre la formulación y resolución de problemas desde un enfoque computacional, en términos de los principios y conceptos fundamentales de la computación (esto es, lo que se ha venido en denominar ‘Pensamiento Computacional’ al que dedicamos por completo el Capítulo 3); siendo la lectoescritura de programas informáticos o ‘*coding*’ su principal práctica asociada.

En la siguiente Tabla 2.4, se detallan los objetivos específicos de la asignatura “*Computing*” en función del ciclo educativo (‘*key stage*’ en terminología británica):

Tabla 2.4. Objetivos específicos de la asignatura “*Computing*” del Reino Unido en función del ciclo educativo

Ciclo Educativo (Key Stage)	Objetivos específicos
Key Stage – 1 (de 5 a 7 años)	<p>Los estudiantes deben ser capaces de:</p> <ul style="list-style-type: none"> ✓ Comprender qué son los algoritmos, y cómo son implementados como programas en los dispositivos digitales; y que los programas se ejecutan siguiendo instrucciones precisas e inequívocas. ✓ Crear y depurar programas informáticos simples. ✓ Utilizar el razonamiento lógico para predecir el comportamiento de programas simples. ✓ Utilizar la tecnología de manera propositiva, para crear, organizar, almacenar, manipular, y recuperar contenidos digitales. ✓ Reconocer usos comunes de las tecnologías digitales de la información más allá de la escuela ✓ Utilizar la tecnología de manera segura y respetuosa, manteniendo la información personal en privado; identificar a dónde acudir en busca de ayuda y apoyo cuando tengan preocupaciones o inquietudes sobre el contenido o contactos establecidos en Internet (u otras tecnologías ‘<i>on-line</i>’)
Key Stage – 2 (de 7 a 11 años)	<p>Los estudiantes deben ser capaces de:</p> <ul style="list-style-type: none"> ✓ Diseñar, escribir y depurar programas que acometan metas específicas, incluyendo el control o simulación de sistemas físicos; resolver problemas a través de su descomposición en partes más pequeñas. ✓ Utilizar la secuenciación, la selección (condicional), y la repetición en los programas; trabajar con variables y con varias formas/formatos de <i>input</i> y <i>output</i>. ✓ Utilizar el razonamiento lógico para explicar cómo funcionan algunos algoritmos sencillos, y detectar y corregir errores en algoritmos y programas. ✓ Comprender el funcionamiento de las redes informáticas, incluida Internet; cómo pueden proporcionar múltiples servicios, como por ejemplo la <i>World Wide Web</i>, y las oportunidades que ofrecen para la comunicación y la colaboración. ✓ Utilizar de manera efectiva las tecnologías de búsqueda, apreciar cómo se seleccionan y ordenan los resultados, y tener capacidad de discernimiento para evaluar contenido digital. ✓ Seleccionar, utilizar y combinar una amplia variedad de software (incluidos servicios de Internet) sobre un amplio abanico de dispositivos digitales, para diseñar y crear una amplia variedad de programas, sistemas y contenidos que acometan determinados objetivos; incluyendo la recolección, análisis, evaluación y presentación de la información. ✓ Utilizar la tecnología de manera segura, respetuosa y responsable; reconocer el comportamiento digital aceptable o inaceptable; identificar un abanico de formas para informar sobre preocupaciones o inquietudes relativas al contenido o contactos establecidos en la Red.

Ciclo Educativo (Key Stage)	Objetivos específicos
<p>Key Stage – 3 (de 11 a 14 años)</p>	<p>Los estudiantes deben ser capaces de:</p> <ul style="list-style-type: none"> ✓ Diseñar, usar y evaluar abstracciones computacionales que modelen el estado y comportamiento de problemas del mundo real y de sistemas físicos. ✓ Comprender diversos algoritmos clave que reflejen el pensamiento computacional (por ejemplo, algoritmos de clasificación y búsqueda); utilizar el razonamiento lógico para comparar la eficacia/eficiencia de algoritmos alternativos para un mismo problema. ✓ Usar dos o más lenguajes de programación, al menos uno de los cuales sea textual, para resolver una variedad de problemas computacionales; hacer un uso apropiado de las estructuras de datos (por ejemplo, listas, tablas, matrices); diseñar y desarrollar programas modulares que usen procedimientos o funciones. ✓ Comprender la lógica Booleana (por ejemplo los conectores ‘and’, ‘or’ y ‘not’) y algunos de sus usos en electrónica y programación; comprender cómo los números pueden ser representados en código binario y ser capaz de llevar a cabo operaciones simples sobre números binarios (por ejemplo, suma binaria y conversión entre sistema binario y decimal). ✓ Comprender los componentes de hardware y software que constituyen los sistemas informáticos, y cómo se comunican entre ellos y con otros sistemas. ✓ Comprender cómo las instrucciones se almacenan y ejecutan dentro de un sistema informático; comprender cómo datos de diverso tipo (incluyendo texto, sonidos e imágenes) pueden ser representados y manipulados digitalmente, en forma de dígitos binarios. ✓ Empezar proyectos creativos que impliquen selección, uso y combinación de múltiples aplicaciones, preferiblemente a través de un conjunto de dispositivos, para alcanzar metas desafiantes, que incluyan recolección y análisis de datos y satisfagan necesidades de usuarios conocidos. ✓ Crear, reutilizar y revisar artefactos digitales para una audiencia dada, con atención a su integridad, diseño y usabilidad. ✓ Comprender un conjunto de formas de uso seguro, respetuoso y responsable de la tecnología, que incluya la protección de su identidad y privacidad ‘on-line’; reconocimiento de contenidos, conductas y contactos inapropiados y saber cómo informar de problemas.
<p>Key Stage – 4 (de 14 a 16 años)</p>	<p>En el ciclo superior de Educación Secundaria, todos los alumnos deben tener la oportunidad de estudiar aspectos de las tecnologías digitales de la información y de las Ciencias de la Computación con suficiente profundidad como para permitirles progresar hacia estudios de nivel superior o a una carrera profesional. Los estudiantes deben ser capaces de:</p> <ul style="list-style-type: none"> ✓ Desarrollar su capacidad, creatividad y conocimiento en Ciencias de la Computación, medios digitales y tecnologías de la información. ✓ Desarrollar y aplicar sus habilidades analíticas, de resolución de problemas, de diseño y de pensamiento computacional. ✓ Comprender cómo los cambios en la tecnología afectan a la seguridad, incluyendo nuevas formas para proteger su privacidad e identidad ‘on-line’, y cómo informar de problemas.

2.4.2.2. *Un ejemplo nacional: el currículo de la Comunidad de Madrid*

Siguiendo la estela del nuevo currículo británico, la Consejería de Educación de la Comunidad de Madrid ha incluido, a través de un decreto de mayo de 2015 ([DECRETO 48/2015](#)), una nueva

asignatura de libre configuración autonómica¹⁶⁹ denominada “*Tecnología, Programación y Robótica*”, que deben cursar de manera obligatoria todos los alumnos madrileños a lo largo de 1º, 2º y 3º de la Educación Secundaria Obligatoria. Para cada uno de los cursos, se establece una carga lectiva de la asignatura de 2 horas semanales (de un total de 30 horas de clase a la semana). La asignatura se articula en torno a cinco ejes:

- i. Programación y pensamiento computacional
- ii. Robótica y la conexión con el mundo real
- iii. Tecnología y el desarrollo del aprendizaje basado en proyectos
- iv. Internet y su uso seguro y responsable
- v. Técnicas de diseño e impresión 3D

En la siguiente Tabla 2.5, se recogen los contenidos establecidos por la Comunidad de Madrid para cada uno de los tres cursos en los que se aborda la asignatura. Como se puede observar, los contenidos fijados no son exclusivos de programación y pensamiento computacional, sino que se mantienen algunos contenidos de la anterior asignatura de “*Tecnología*” (p.e. electricidad, electrónica, proyecto tecnológico). Se espera que, como resultados de aprendizaje, los alumnos, al acabar el itinerario formativo, sepan: crear una página web y una aplicación para dispositivos móviles; diseñar un videojuego; manejar la impresión en 3D; y tener conocimientos de robótica. Para la implantación de la asignatura se promueve la utilización de software ‘*open source*’¹⁷⁰ (p.e. Scratch para la programación de videojuegos a través de un lenguaje visual por bloques; o App Inventor para el desarrollo de aplicaciones móviles). Y para la parte de robótica se promueve el uso de hardware libre (p.e. placas Arduino).

Tabla 2.5. Contenidos de la asignatura “*Tecnología, Programación y Robótica*” de la Comunidad de Madrid

1º ESO	2º ESO	3º ESO
1. Internet: páginas Web, aplicaciones que intercambian datos.	1. Análisis y resolución de problemas mediante algoritmos.	1. Formulación de un proyecto tecnológico. Identificación del problema. Análisis de su naturaleza.
2. Privacidad y responsabilidad digital.	2. Internet: arquitectura y protocolos.	2. Innovación y creatividad para la búsqueda de soluciones tecnológicas.
3. Herramientas de programación visuales por bloques	3. Seguridad en Internet.	3. Diseño y representación gráfica de los elementos de un proyecto tecnológico
4. Aplicaciones para dispositivos móviles.	4. Aplicaciones y servicios para internet y nuevas tendencias en la red.	4. Documentación de un proyecto para la elaboración de un prototipo tecnológico.
5. Proyectos tecnológicos	5. Páginas Web. Gestores de contenidos (CMS) y herramientas de publicación.	5. Divulgación de la evolución de un proyecto tecnológico a través de la Web.
6. Materiales de uso tecnológico	6. Estructuras y mecanismos.	6. Diseño y fabricación de los elementos mecánicos de un proyecto tecnológico mediante impresión 3D.
7. Electricidad y circuitos eléctricos en continua.	7. Diseño e impresión 3D.	7. Diseño, montaje y medida de los circuitos electrónicos de un proyecto tecnológico.
	8. Conceptos básicos de señales y sistemas de comunicaciones.	8. Programación de los circuitos electrónicos de un proyecto tecnológico.
	9. Sistemas electrónicos analógicos y digitales.	9. Documentación de un prototipo desarrollado a través de un proyecto tecnológico.
	10. Programación de sistemas electrónicos (robótica).	

¹⁶⁹ Un tipo de asignatura establecido en la LOMCE (2013), que permite a las administraciones educativas determinar sus contenidos, fijar horarios y definir los estándares de aprendizaje evaluables.

¹⁷⁰ Software de código abierto

El currículo oficial de la Comunidad de Madrid ([DECRETO 48/2015](#)) también fija los criterios de evaluación y los estándares de aprendizaje evaluables. Por ejemplo, los criterios respecto del bloque de Programación son:

- Mantener y optimizar las funciones principales de un ordenador, tableta o teléfono móvil en los aspectos referidos a su uso, su seguridad y a las funciones del sistema operativo.
- Analizar los diferentes niveles de lenguajes de programación.
- Utilizar con destreza un entorno de programación gráfica por bloques.
- Desarrollar y programar aplicaciones móviles sencillas en entornos de programación por bloques.
- Desarrollar una página Web sobre un gestor de contenidos (CMS)
- Analizar el proceso de programación de páginas Web en un lenguaje estándar.
- Desarrollar programas en un lenguaje de programación textual (Lenguajes de programación textuales pueden ser, por ejemplo, Python, PHP, Processing, Ruby, JavaScript, etc.)

Aunque resulta evidente que tanto el nuevo currículo de la Comunidad de Madrid como, especialmente, el de Reino Unido suponen un paso adelante en la incorporación del ámbito tecnológico e informático al sistema educativo formal desde una perspectiva códigoalfabetizadora y computacional; ya comienzan a aparecer algunas críticas respecto de su planteamiento. Por ejemplo, Valverde, Fernández y Garrido ([2015](#)) afirman que ambas propuestas curriculares:

“Se enmarcan dentro de propuestas curriculares [excesivamente] prescriptivas, organizadas en torno a disciplinas académicas, para grupos homogéneos de estudiantes y con un grado de innovación educativa [excesivamente] dependiente, en gran medida, de la opción metodológica del profesorado en cada contexto específico” (Valverde, Fernández, & Garrido, 2015, p. 8)

En cualquier caso, ambos currículos están recién implantados y necesitan de abundante investigación empírica que evalúe sus virtudes y defectos.

En síntesis, y para cerrar este apartado, podemos afirmar que se encuentra evidencia suficiente para enunciar de manera fundamentada la hipótesis de *viabilidad* de la códigoalfabetización en nuestro sistema educativo, dada la gran variedad de recursos y herramientas existentes para la enseñanza-aprendizaje del ‘coding’, y la emergente apertura de los currículos educativos a la inclusión del ‘coding’ en su seno.

2.5. Relevancia: el para qué de la códigoalfabetización

Afrontamos, por último, la cuestión de si es *relevante* incorporar la códigoalfabetización a nuestro sistema educativo; o el *para qué* de la códigoalfabetización. En otras palabras, si los *productos* de aprendizaje que se derivan de la códigoalfabetización son deseables y valiosos, en sentido amplio, en relación al contexto social actual.

Los *productos* de la códigoalfabetización pueden ser entendidos en un doble sentido: por un lado, son productos los objetos digitales realizados a través de la lectoescritura con código informático (‘apps’, animaciones digitales, videojuegos, webs, etc...); por otro lado, podemos considerar al

propio sujeto que ha aprendido a leer y escribir con los lenguajes informáticos de programación como un *producto* de la codigoalfabetización. Abordaremos estas dos vertientes por separado en los siguientes epígrafes.

2.5.1. Productos digitales derivados del ‘coding’

Ya vimos en el apartado 1.3 del Capítulo 1, cómo las aplicaciones para dispositivos móviles o ‘apps’ vienen convirtiéndose en los últimos años, por la proliferación de *smartphones* y la extensión de la conectividad ‘en cualquier parte y en cualquier momento’, en objetos digitales de creciente uso, utilidad y valoración social; que potencialmente pueden ser puestos a disposición de inmensas audiencias reales y globales.

Hacíamos entonces una breve referencia a los ‘I Premios Apps de la Fundación Telefónica’¹⁷¹, un concurso destinado a chicos y chicas de entre 18 y 25 años para crear aplicaciones móviles tras un periodo formativo con Firefox OS, el sistema operativo de código abierto cuyo lanzamiento ha sido liderado por Telefónica I+D en España. Los premios de esta primera edición fueron fallados en el mes de enero de 2014. Al revisar los proyectos de ‘apps’ ganadores, se hacen evidentes algunas características de estos productos de la codigoalfabetización: son *versátiles* (tratan sobre una amplia variedad de temas), son *útiles* (sirven para resolver algún problema o cubrir alguna necesidad), y son *usables-accesibles* (el público general puede llegar a descargarlas y disfrutarlas en su dispositivo móvil personal).

Así, por ejemplo, en la ‘categoría social’ de los premios, los ganadores fueron Jorge Prudencio Fernández con la ‘app *LifeRescuer*’¹⁷² para rescates en la montaña a través de GPS en sitios de poca cobertura; Manuel Ballarín Naya, creador de ‘*3WTasks*’¹⁷³, una red social para la oferta y demanda de trabajo; Antonio José Martínez Sánchez con ‘*iSpeak*’¹⁷⁴, una ‘app’ social que ayuda a personas con discapacidad a hablar con facilidad; y Adrián de la Rosa Martín, con el proyecto ‘*Setties*’¹⁷⁵, una guía turística que funciona por geolocalización y da información del sitio a través de artículos de Wikipedia.

En la ‘categoría educativa’, los premios correspondieron a Asier Aguado Corman con el proyecto ‘*Manzanas Mágicas*’¹⁷⁶, una ‘app’ que refuerza el aprendizaje de las matemáticas en niños pequeños a partir de operaciones básicas; Manuel Casas Barrado con la aplicación ‘*Educacity*’¹⁷⁷, una guía cultural de la ciudad de Sevilla; Francisco Antonio Rampas Vázquez con ‘*Daily Expression*’¹⁷⁸, una ‘app’ que permite aprender a diario frases hechas en inglés; y Eva Fernández Rodríguez con la ‘App - *Tu huerto desde cero*’¹⁷⁹, una guía para cultivar tu propia cosecha.

¹⁷¹ <http://www.fundaciontelefonica.com/premios-apps/> [Los 8 ganadores de dicho concurso, 7 chicos y 1 chica, forman parte del grupo de expertos consultado para la validación de contenido de nuestro Test de Pensamiento Computacional (TPC), tal y como se describirá en el apartado 6.3 de esta tesis doctoral.]

¹⁷² Disponible en <https://marketplace.firefox.com/app/liferescuer-para-monta%C3%B1a/>

¹⁷³ https://youtu.be/HZRJ_OQcSZE

¹⁷⁴ Disponible en <https://itunes.apple.com/es/app/ispeak-spanish/id303491384?mt=8>

¹⁷⁵ <https://youtu.be/pIglj62TvM>

¹⁷⁶ Disponible en <http://www.magicapples.es/>

¹⁷⁷ https://youtu.be/h8_9VSdSKGI

¹⁷⁸ <http://dailyexpression.hol.es/>

¹⁷⁹ <https://youtu.be/n9Gmo9NfK14>

Veamos si podemos replicar y extender este mismo análisis desde edades universitarias hacia edades escolares (Primaria y Secundaria); y desde la creación específica de ‘apps’ a la creación de artefactos digitales de todo tipo. Para ello, acudimos a la reciente celebración de la primera edición del ‘Concurso Dr. Scratch’¹⁸⁰. Este concurso está promovido por la Fundación Española para la Ciencia y la Tecnología (FECYT), Google y la Universidad Rey Juan Carlos. El ‘Concurso Dr. Scratch’ está dirigido a escolares españoles de Educación Primaria y Secundaria; a los que reta a crear-escribir un proyecto en el lenguaje visual por bloques Scratch¹⁸¹ sobre una temática determinada. En esta primera edición, la temática elegida para el concurso fue ‘La Ciencia’. De alguna manera, el ‘Concurso Dr. Scratch’ supone una actualización de los antiguos (y analógicos) concursos de redacción escrita, sólo que ahora lo que se escribe ya no es texto plano sino programas informáticos a través de su código.

Casi un centenar de escolares de toda España participaron en el concurso, cuyos premios fueron fallados el pasado mes de diciembre de 2015, tras la valoración ciega de los proyectos Scratch por un grupo de 16 expertos (en su mayoría profesores de Informática de Primaria y Secundaria); trascendiendo a los medios, especialmente, el sexo femenino de los ganadores, tanto en la categoría de Primaria como de Secundaria (Yáñez, 2015). Veamos con algo más de detalle el contenido de 3 de los proyectos Scratch presentados, que obtuvieron una valoración excelente.

El proyecto “*Flotar o no flotar: Esa es la cuestión*”¹⁸² es obra de un estudiante de Secundaria. Está compuesto internamente por un total de 61 programas y 13 objetos, que conforman una simulación física interactiva sobre la flotabilidad de distintos cuerpos-materiales en función del líquido sobre el que se sumergen. Así, el ‘lector’ del proyecto puede ir probando a sumergir distintos materiales (hierro, aluminio, oro, madera de roble...) en distintos fluidos (agua, aceite de oliva, mercurio líquido), tratando de anticipar si el cuerpo flotará o no; obteniendo puntos por ello (Figura 2.26)

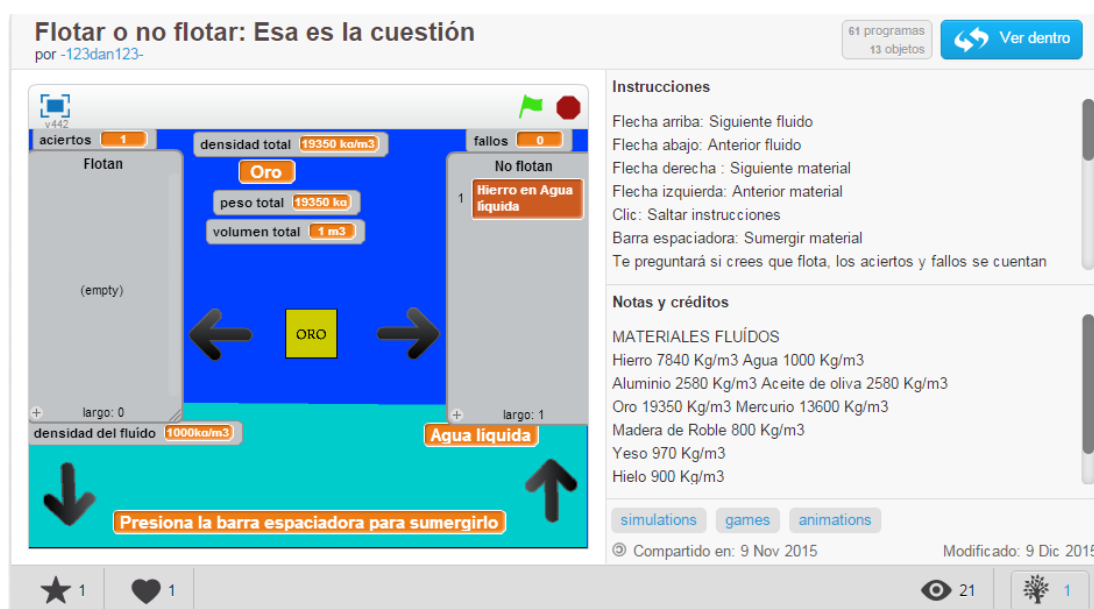


Figura 2.26. Proyecto Scratch “*Flotar o no flotar: Esa es la cuestión*” [<https://scratch.mit.edu/projects/85723084/>]

¹⁸⁰ <http://drscratch.org/contest> [Dr. Scratch es una herramienta ‘on-line’ de análisis automático de proyectos-programas informáticos escritos con Scratch, al que dedicaremos un sub-epígrafe específico en el Capítulo 3]

¹⁸¹ <https://scratch.mit.edu/>

¹⁸² Disponible (y remezclable pinchando sobre ‘Ver dentro’) en: <https://scratch.mit.edu/projects/85723084/>

El proyecto “*La aventura de Mark*”¹⁸³ (1º Premio de la categoría de Primaria) es una narración interactiva sobre las peripecias del perro Mark, que se ha quedado solo durante las vacaciones en el laboratorio donde trabaja su dueña (una famosa química). El proyecto está internamente compuesto por 178 programas y 37 objetos. Tal y como puede verse en la parte inferior derecha de la siguiente Figura 2.27, este proyecto ha sido visto, acaso diríamos ‘leído’, por 45 personas; y ha sido remezclado en 2 ocasiones.

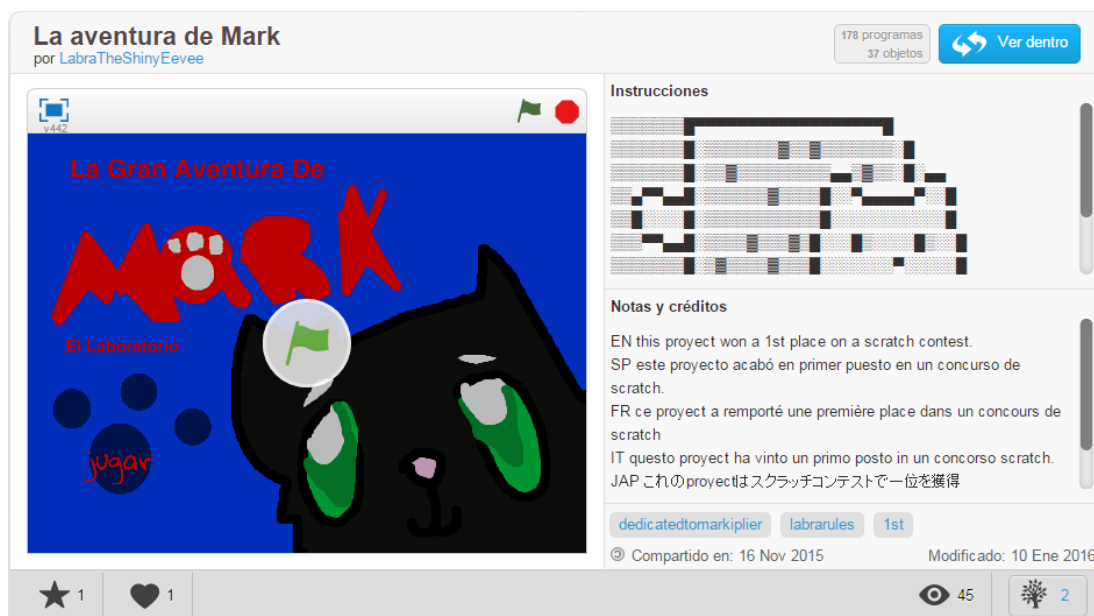


Figura 2.27. Proyecto Scratch “*La aventura de Mark*” [<https://scratch.mit.edu/projects/86270038/>]

Finalmente, el proyecto “*Concurs*”¹⁸⁴, escrito en lengua valenciana por una alumna de 4º de Primaria, consiste en un videojuego interactivo de preguntas y respuestas sobre ciencia (Figura 2.28)



Figura 2.28. Proyecto Scratch “*Concurs*” [<https://scratch.mit.edu/projects/88301574/>]

¹⁸³ Disponible (y remezclable pinchando sobre ‘Ver dentro’) en: <https://scratch.mit.edu/projects/86270038/>

¹⁸⁴ Disponible (y remezclable pinchando sobre ‘Ver dentro’) en: <https://scratch.mit.edu/projects/88301574/>

Si pinchamos en ‘Ver dentro’, accedemos a la composición interna del proyecto (cuyo código podemos remezclar o, dicho de otra manera, re-escribir). En la parte inferior izquierda de la siguiente Figura 2.29, podemos ver los 10 objetos que componen el proyecto “*Concurs*”, y a la derecha los programas que controlan uno de dichos objetos.

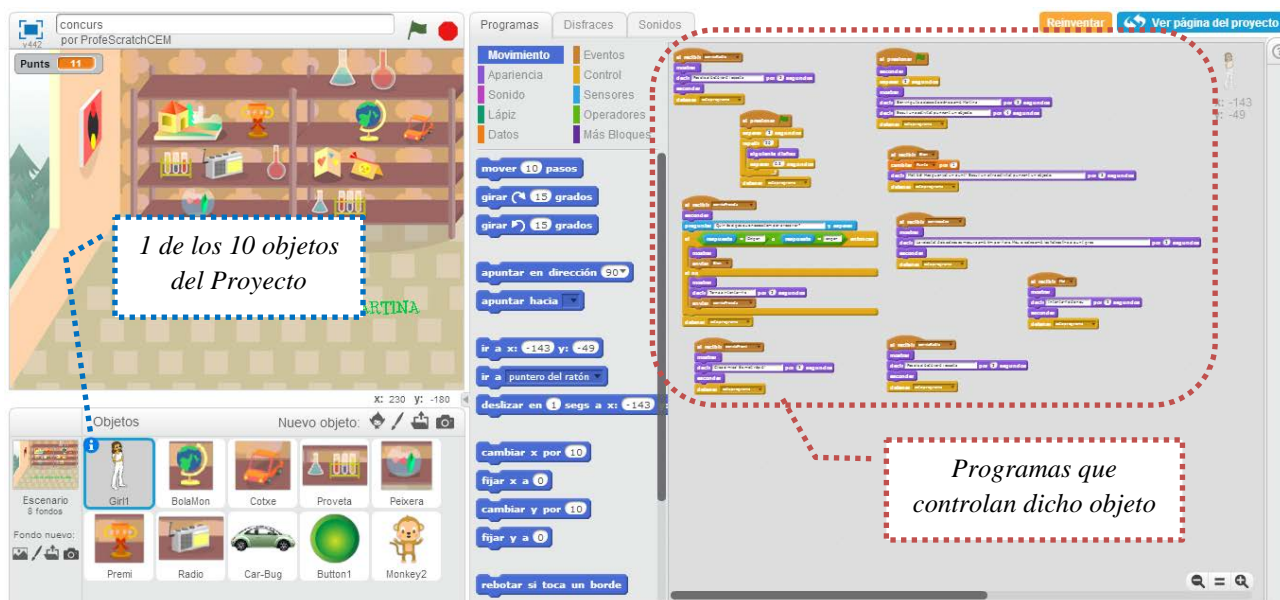


Figura 2.29. Composición interna, editable, del proyecto “*Concurs*” [<https://scratch.mit.edu/projects/88301574/#editor>]

A su vez, si acercamos nuestra observación hacia uno de los programas concretos que controlan uno de los objetos, encontramos, por ejemplo, el *script* (programa corto) que se muestra en la Figura 2.30. Tal y como puede intuirse, este programa se ejecuta cuando el lector-jugador clicla sobre la probeta; entonces se le pregunta sobre el gas que necesitamos los humanos para respirar y, en función de su respuesta, se le muestra un mensaje de felicitación o de ánimo para probar de nuevo.



Figura 2.30. *Script* que controla uno de los objetos del proyecto “*Concurs*”

Así pues, estos otros productos, fruto de la ampliación de foco que supone la codigoalfabetización, presentan también las características señaladas anteriormente para el caso concreto de las ‘apps’. Desde luego, son *versátiles* (pueden proyectarse sobre cualquier temática y en distintas formas, como simulaciones, narraciones, videojuegos...); son *útiles* (por ejemplo, el proyecto “*Flotar o no flotar*” sirve para el aprendizaje y estudio del concepto físico de densidad); y son *usables-accesibles* (basta con disponer de la URL del proyecto y un dispositivo electrónico con conexión a Internet). Pero, además, presentan algunas características adicionales: son *sociales* (los proyectos son comentados y valorados en la plataforma por otros programadores, generándose comunidades de aprendizaje); son *remezclables* (los proyectos no sólo se leen-consumen sino que se pueden re-escribir y reinventar mediante sucesivas remezclas del código); y son *cognitivamente complejos* (para la creación de estos proyectos deben ponerse en juego habilidades cognitivas de orden superior, como el establecimiento de estructuras condicionales que se ejemplifica en la anterior Figura 2.30. Estas habilidades cognitivas subyacente a la lectoescritura de código es lo que ha venido a denominarse ‘pensamiento computacional’, al que dedicamos por completo el Capítulo 3.

En síntesis, encontramos evidencia sobre el valor y la deseabilidad de los productos digitales derivados de la codigoalfabetización; y por tanto, hallamos fundamento para enunciar nuestra hipótesis directiva de *relevancia*. Pero, análogamente, el sujeto codigoalfabetizado, capaz de leer y escribir con los lenguajes informáticos de programación, adquiere progresivo valor en nuestra sociedad digital. A su análisis crítico dedicamos el siguiente epígrafe.

2.5.2. Análisis crítico del sujeto codigoalfabetizado: ‘hackers’ y ‘makers’

El sujeto codigoalfabetizado está adquiriendo creciente valor económico en la sociedad digital. Por un lado, es un sujeto capaz de leer y escribir software, el ‘meta-lenguaje’ que media toda la comunicación digital actual; y en consecuencia, un sujeto que puede aportar valor añadido al contexto profesional en el cual se inserte. Por otro lado, existe una creciente brecha entre la oferta de puestos de trabajo que requieren competencias en programación informática y el número de sujetos actualmente codigoalfabetizados; por ejemplo, sólo en Europa se calcula que, para 2020, habrá 800.000 puestos de trabajo en el área de la programación y la computación que quedarán vacantes por falta de profesionales capacitados ([European Schoolnet, 2015](#)). Así pues, el sujeto codigoalfabetizado es, por el momento, una *rara avis* muy apreciada en el mercado laboral digital.

Así las cosas, no es de extrañar que el ‘programador informático’, y otras emergentes profesiones afines como el ‘científico de datos’¹⁸⁵ (*‘data scientist’*), se halle entre los perfiles profesionales mejor pagados. Incluso proliferan en la blogosfera artículos que ordenan, a modo de rankings, los distintos lenguajes informáticos de programación en función de los sueldos asociados a su dominio ([Allan, 2014](#); [Jee, 2015](#)); sueldos que se mueven en el entorno de los 100.000 dólares anuales, las ansiadas ‘*six figures*’, tal y como puede verse en la siguiente Figura 2.31¹⁸⁶.

¹⁸⁵ El ‘científico de datos’ es un emergente perfil profesional en la intersección de la estadística y las ciencias de la computación; cuya principal función es la extracción de conocimiento a partir del procesamiento de enormes cantidades de datos digitales (incluyendo la programación de algoritmos que sirvan para su interpretación)

¹⁸⁶ Nótese que uno de los lenguajes incluidos en esta figura es *R* [<https://www.r-project.org/>], un lenguaje específico para el análisis estadístico desde un punto de vista computacional y muy usado por los ‘científicos de datos’

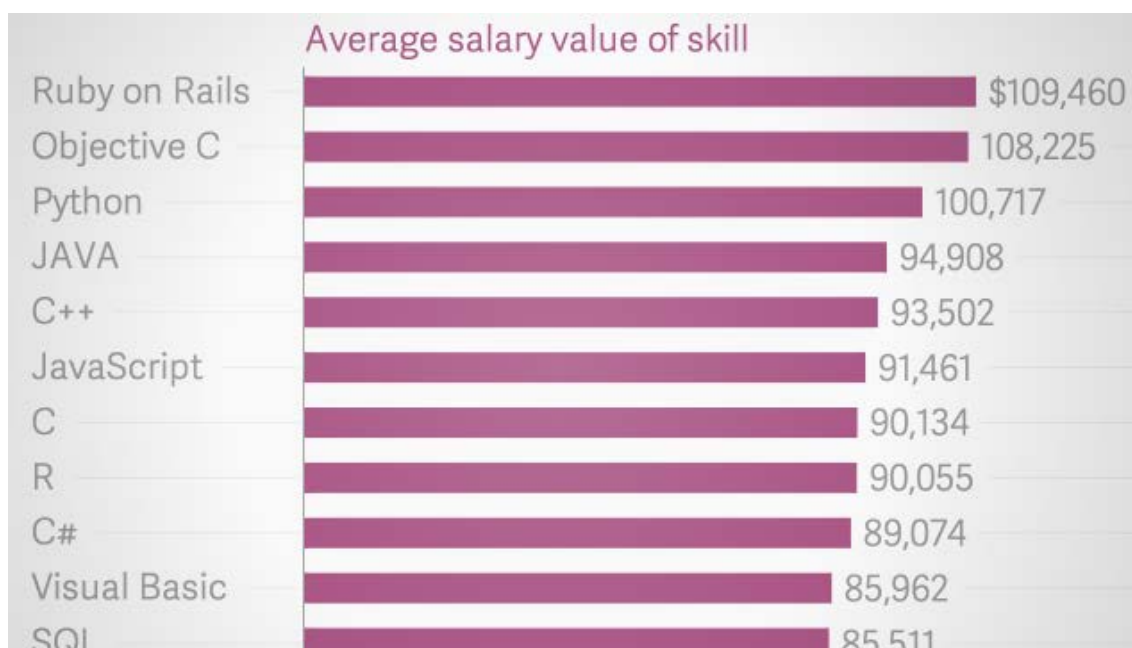


Figura 2.31. Ranking de sueldos para un programador informático en función del lenguaje que domine (Allan, 2014)

Pero más allá del valor económico del sujeto codigoalfabetizado, existe una dimensión ética, que ha venido a denominarse ‘la ética del *hacker*’ (Himanen, 2004; Levy, 1984; Raymond, 2001, 2003). Lo primero que hace dicha ética es rescatar las acepciones originales, no peyorativas, del término ‘*hacker*’ (del verbo inglés ‘*to hack*’ \approx ‘cortar con un hacha de manera precisa, y luego montar los trozos’). El ‘*hacker*’ no es el pirata o vándalo informático que pone sus conocimientos técnicos de programación al servicio de la delincuencia (el término correcto que describe al delincuente informático es ‘*cracker*’). Pero entonces, ¿qué es un ‘*hacker*’?; Pekka Himanen lo define así al comienzo de su manifiesto “*La ética del hacker y el espíritu de la era de la información*”:

“En el centro de nuestra era tecnológica se hallan unas personas que se autodenominan hackers. Se definen a sí mismos como personas que se dedican a programar de manera apasionada y creen que es un deber para ellos compartir la información y elaborar software gratuito. No hay que confundirlos con los crackers, los usuarios destructivos cuyo objetivo es el de crear virus e introducirse en otros sistemas: un hacker es un experto o un entusiasta de cualquier tipo que puede dedicarse o no a la informática. En este sentido, la ética hacker es una nueva moral que desafía la ética protestante del trabajo, tal como la expuso hace casi un siglo Max Weber en su obra clásica “La ética protestante y el espíritu del capitalismo”, y que está fundada en la laboriosidad diligente, la aceptación de la rutina, el valor del dinero y la preocupación por la cuenta de resultados. Frente a la moral presentada por Weber, la ética del trabajo para el hacker se funda en el valor de la creatividad, y consiste en combinar la pasión con la libertad. El dinero deja de ser un valor en sí mismo y el beneficio se cifra en metas como el valor social y el libre acceso, la transparencia y la franqueza” (Himanen, 2004, p. 2)

En esta misma línea, Raymond (2001) expresa en unas pocas sentencias cómo convertirse en ‘*hacker*’, una mezcla de actitud-ética y competencia técnica; centrándonos en lo primero:

“La actitud del hacker. Los hackers resuelven problemas y construyen cosas, y creen en la libertad y la ayuda voluntaria mutua (...) si quieres ser un hacker, repite lo que sigue hasta que te creas lo que estás diciendo (...)

1. El mundo está lleno de problemas fascinantes que esperan ser resueltos: *Es muy divertido ser un hacker, pero es la clase de diversión que requiere mucho esfuerzo. El esfuerzo requiere motivación. Los atletas triunfadores obtienen su motivación a partir de un tipo de placer físico que surge de trabajar su cuerpo, al forzarse a sí mismos más allá de sus propios límites físicos. De manera similar, para ser un hacker deberás sentir un estremecimiento de tipo primitivo cuando resuelves problemas, afinas tus habilidades y ejercitas tu inteligencia (...)*

2. Ningún problema tendría que resolverse dos veces: *Los cerebros creativos son un recurso valioso y limitado. No deben desperdiciarse reinventando la rueda cuando hay tantos y tan fascinantes problemas nuevos esperando por allí. Para comportarte como un hacker, debes creer que el tiempo para pensar que emplean otros hackers es precioso —tanto, que es casi una obligación moral para ti el compartir la información, resolver los problemas y luego exponer la solución de manera que los otros hackers puedan resolver nuevos problemas, en lugar de enfrentarse perpetuamente con los viejos (...)*

3. El aburrimiento y el trabajo rutinario son perniciosos: *Los hackers (y las personas creativas en general) nunca debieran ser sometidas a trabajos estúpidamente repetitivos, porque cuando esto sucede significa que no están haciendo lo único que son capaces de hacer: resolver nuevos problemas. Este desperdicio de talento daña a todo el mundo. Por ello, las tareas rutinarias, repetitivas y aburridas no sólo son desagradables, sino intrínsecamente perversas (...) Para comportarte como un hacker, debes creer en esto lo suficiente como para automatizar las tareas rutinarias todo lo que se pueda, no solamente por ti mismo, sino para beneficio de todos los demás (especialmente otros hackers).*

4. La libertad es buena: *Los hackers son antiautoritarios por naturaleza. Cualquiera que pueda darte órdenes, puede obligarte a dejar de resolver ese problema que te está fascinando —y, dada la manera como trabajan las mentes autoritarias, encontrarán alguna razón espantosamente estúpida para hacerlo. Por eso, la actitud autoritaria debe ser combatida donde sea que se la encuentre, pues si se la deja te asfixiará, tanto a ti como a otros hackers.*

5. La actitud no es sustituto para la competencia: *Para ser un hacker, deberás desarrollar algunas de esas actitudes. Pero tener solo la actitud no te transformará en hacker, como tampoco te puede transformar en un atleta campeón o en estrella del rock. Para transformarte en hacker necesitarás inteligencia, práctica, dedicación y trabajo duro” (Raymond, 2001, en línea)*

En cualquier caso, aunque la ética *hacker* nace originalmente en el contexto de las cibercomunidades de programadores que desarrollan software libre de código abierto, es una ética que puede predicarse de cualquier sujeto u organización que siga estos valores (Himanen, 2004): pasión, libertad, conciencia social, verdad, anti-corrupción, lucha contra la alienación del hombre, libre acceso a la información, valor social, accesibilidad, actividad, preocupación responsable, curiosidad, creatividad

o interés. En ese sentido, recientemente se vienen acuñando términos como *'hackschooling'*¹⁸⁷ (TEDx Talks, 2013) o *'eduhackers'* para designar prácticas educativas que siguen este marco axiológico.

En paralelo al movimiento *'hacker'* se ha venido produciendo, con una ética que podríamos considerar análoga, el movimiento o cultura *'maker'* (traducido en ocasiones al español como *'cultura hacedora'*). El movimiento *'maker'* puede considerarse una extensión del movimiento *'hacker'* desde el mundo digital hacia el mundo físico. En su manifiesto (Hatch, 2014), afirma:

“Hacer ('make'¹⁸⁸) es algo inherente a la condición humana. Tenemos que hacer, crear, y expresarnos para sentirnos plenos. Hay algo único acerca de hacer [construir, fabricar] objetos físicos. Estos objetos son como pequeños pedazos de nosotros mismos que encierran parte de nuestro alma” (Hatch, 2014, p. 1)

Detrás del movimiento *'maker'* hay también una cultura del *'do it yourself (DIY)'* (*'hazlo por ti mismo'*); que aboga por la fabricación o reparación de cosas por uno mismo, de modo que la persona que práctica el DIY ahorra dinero, aprende y se entretiene al mismo tiempo. Por tanto, el DIY es una cultura de la autoproducción, que se lanza a la fabricación de objetos físicos propios y desde una visión-convicción propia; sin esperar ni depender del consumo de objetos fabricados por terceros desde sus particulares intereses. La ética DIY introduce un elemento nuevo de corte anticonsumista y anticapitalista, por lo que en ocasiones se la sitúa cerca de otras manifestaciones contraculturales (como el *'punk'*). Sea como fuere, el DIY también ha comenzado a calar en la esfera educativa, y se empieza a hablar de *'DIY education'*: un notable ejemplo de esta nueva aproximación educativa es la comunidad *'on-line'* DIY.org (Figura 2.32) en la que niños de todo el mundo se enseñan mutuamente *'que saben hacer por ellos mismos'*:

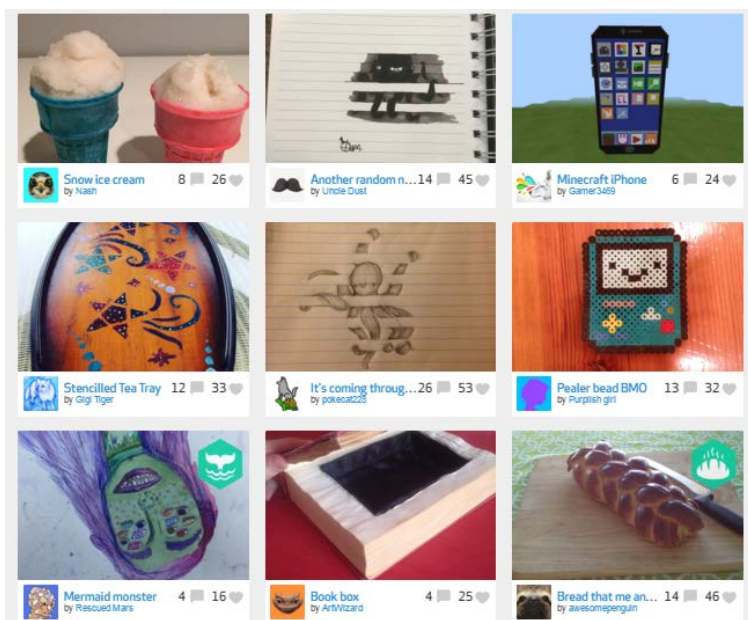


Figura 2.32. Captura de pantalla de la comunidad *'on-line'* DIY.org [<https://diy.org/>]

¹⁸⁷ Aparece por primera vez en la charla TED *“Hackschooling makes me happy”* [Disponible en <https://www.youtube.com/watch?v=h11u3vtcpaY>]

¹⁸⁸ Nótese que la traducción del verbo inglés *'to make'* se correspondería a un *'hacer'* en sentido fuerte y físico (similar a *'confeccionar'*, *'construir'* o *'fabricar'*)

En los últimos años, el movimiento ‘*hacker*’ y el movimiento ‘*maker*’ han fortalecido sus conexiones mutuas a través de la proliferación del hardware libre (Arduino o Raspberry Pi), la robótica y la impresión 3D. En suma, el mundo digital y el mundo físico se están viendo indisolublemente conectados a través de la lectoescritura de software que opera sobre hardware libre y dispositivos de fabricación, antaño enormes y accesibles únicamente para las grandes empresas, que hoy sólo cuestan unas decenas o cientos de euros y que ocupan el espacio de una caja de mudanzas. Por lo tanto, el sujeto plenamente codigoalfabetizado será un sujeto que entienda, y extienda, lo digital sobre lo físico.

Desde una aproximación alternativa y en un tono más filosófico, el sujeto codigoalfabetizado puede suponer una réplica a las críticas que, desde distintos ámbitos, se están realizando a la omnipresencia de lo digital y sus efectos alienantes en el ser humano. Haremos el siguiente juego dialéctico: presentaremos un destacado autor y su crítica a lo digital; y veremos si el sujeto codigoalfabetizado puede suponer un alivio a la misma:

- **#1 Jaron Lanier:** critica cómo las decisiones de diseño, algunas casuales o arbitrarias, sobre ciertas tecnologías digitales vienen determinando nuestra percepción del mundo. Por ejemplo, cómo la tecnología MIDI (*‘Musical Instrument Digital Interface’*), estandarizada y protocolizada en los años 80, ha ‘anclado’ todo el desarrollo posterior de creación y consumo de música en formato digital. En ese sentido, Lanier, en su conocido ensayo “*Contra el rebaño digital*” (Lanier, 2011) nos invita a cuestionar y superar estos ‘anclajes’ a los que nos somete el software; entre dichos ‘anclajes’, destaca cómo el concepto de persona, la identidad misma, está quedando ‘anclado’ en las decisiones de diseño de las redes sociales¹⁸⁹. Lanier nos reta con estas palabras:

“Crea un sitio web que exprese algo sobre ti que no encaje en el molde disponible de una red social. Cuelga de vez en cuando un vídeo cuya creación te haya exigido cien veces más tiempo que el necesario para verlo. Escribe una entrada de blog que te haya exigido semanas de reflexión hasta que has oído la vocecilla interior que necesitaba salir” (Lanier, 2011, p. 37)

El sujeto codigoalfabetizado no da el software por sentado. Tiene capacidad de ‘desanclarse’ del mismo, escribiendo el suyo propio.

- **#2 Byung-Chul Han:** filósofo coreano asentado en Alemania. En su ensayo “*En el enjambre*” (Han, 2014) analiza el comportamiento de las nuevas masas digitales, reunidas alrededor de las redes sociales ‘*on-line*’. Según Han, las redes sociales, lejos de *congregar* a un conjunto de personas en torno a una idea u objetivo (rasgo propio de la masa clásica, que genera un grito coordinado y unísono); más bien *agregan* a una sucesión volátil de individualidades narcisistas, enamoradas de sí mismas, que sólo generan ruido (como el zumbido de un enjambre) por el reenvío y rebote constante de informaciones redundantes y autocomplacientes. La promesa de la web 2.0, la web social y participativa, ha degenerado en

¹⁸⁹ En los días que se escriben estas líneas es noticia de máxima actualidad cómo la inclusión de nuevos botones de interacción en Facebook (una decisión de diseño de software, aparentemente arbitraria), puede implicar cambios sustanciales en la forma de relacionarnos digitalmente (Dans, 2015b)

una espiral aquiescente de información; conocimiento en ningún caso. Posteriormente, Han (2015) critica como las pantallas, *pulidas* y *lisas*, excesivas en positividad, impiden la distancia contemplativa suficiente para que en lo digital se dé auténtica apreciación estética de *lo bello* (a lo sumo, en una pantalla vislumbraremos *lo agradable*)

El sujeto códigoalfabetizado, con su conocimiento del software que subyace a la pantalla, gana en profundidad de apreciación de lo digital. Se sitúa más allá de la pantalla y más fácilmente se sustrae a la espiral que allí acaece.

- **#3 Nicholas Carr:** su libro “*Superficiales: ¿Qué está haciendo Internet con nuestras mentes?*” (Carr, 2011) es una feroz crítica a cómo las tecnologías digitales, en particular Internet, están alterando nuestra capacidad de atención y concentración. Las múltiples pestañas abiertas en el navegador, las notificaciones constantes de nuestras aplicaciones, los continuos enlaces de los hipertextos, etc.; generan un ecosistema multitarea cuya víctima es la atención sostenida y la reflexión profunda.

El sujeto códigoalfabetizado, en su tarea de lectura-escritura de software está obligado a detenerse, a centrarse. Cuando uno escribe código por un momento se deja en suspenso la potencia y la velocidad de la propia máquina (igual que un coche se detiene en manos de un mecánico que debe revisar sus tripas)

- **#4 Lev Manovich:** al que ya citamos en la introducción de esta tesis doctoral. En su ensayo “*El Software toma el mando*” (Manovich, 2013) argumenta cómo el software se ha convertido en el motor de las sociedades contemporáneas y, sin embargo, la poca atención relativa que ha recibido como objeto de estudio cultural (y, en paralelo, como objeto de enseñanza-aprendizaje en los sistemas educativos):

“El colegio y el hospital, la base militar y el laboratorio científico, el aeropuerto y la ciudad (todos los sistemas sociales, económicos y culturales de la sociedad moderna) funcionan con ayuda del software. El software es la cola invisible que mantiene unidos todos los componentes. Si bien cada sistema de la sociedad moderna habla su lenguaje y persigue sus propias metas, todos comparten la sintaxis del software: sentencias condicionales de “si... entonces” o “mientras... hacer” (...) el software es, asimismo, el motor de la globalización, en tanto en cuanto permite a las empresas repartir nodos de gestión, centros de producción y puntos de almacenado y consumo por todo el mundo (...) si fijamos nuestro punto de mira en cualquiera de estos nuevos elementos de la vida contemporánea, nos daremos cuenta de que todos son posibles gracias a la existencia de software. Ha llegado, pues, el momento de prestarle toda nuestra atención” (Manovich, 2013, p. 24-26)

El sujeto códigoalfabetizado se sitúa a la altura que fija el papel predominante del software en la sociedad actual. Por analogía, cualquier alfabetización digital será incompleta si no incluye explícitamente la códigoalfabetización.

- **#5 Sherry Turkle:** prestigiosa psicóloga social del Massachusetts Institute of Technology (MIT). En sus dos últimos libros, *“Alone Together: Why We Expect More from Technology and Less from Each Other”* (Turkle, 2011) y *“Reclaiming Conversation: The Power of Talk in a Digital Age”* (Turkle, 2015); advierte del peligro que supone sacrificar las relaciones sociales auténticas por el mero ‘estar conectado’. Turkle constata cómo progresivamente esperamos más de la tecnología y menos de las personas, y cómo nos vamos deslizado hacia relaciones sociales digitales que podemos controlar a golpe de ‘clic’. Esta hiperconexión e hipersocialización banal se constituye en una huida hacia delante para escapar de una continua sensación de soledad y de aislamiento. Frente a ello, Turkle reclama la importancia de volver a establecer conversaciones auténticas, cara a cara (o también digitales).

El sujeto códigoalfabetizado vuelve a situar a la persona en el centro del sistema. No le importan las herramientas digitales por sí mismas, sino el proceso, cognitivo y social, de lectoescritura de dichas herramientas a través del código. Dicho de otra manera, el sujeto códigoalfabetizado es consciente de que no puede esperar nada del software por sí mismo, pero sí de otro sujeto. Adicionalmente, las relaciones que se establecen al escribir código colaborativamente se parecen más a conversaciones significativas para los interlocutores, que el mero pinchar ‘me gusta’ al pie de foto de un contacto en Instagram.

- **#6 Éric Sadin:** autor de *“La vie algorithmique: Critique de la raison numérique”* (Sadin, 2015), aún no disponible en español y que podríamos traducir como *“La vida algorítmica: Crítica de la razón digital”*. Sadin expone cómo cada vez más decisiones son basadas en datos (*‘data-driven’*). Ello es posible, por un lado, por la enorme cantidad de datos que emitimos continuamente, pero también por la existencia de algoritmos, reglas de decisión escritas como programas informáticos, que alimentados de dicho datos establecen prioridades y decisiones; en la mayoría de las ocasiones, dichos algoritmos están ocultos y no son conocidos¹⁹⁰. Sadin advierte del peligro de caer en un sistema de ‘gobernanza algorítmica’, una ‘dictadura dataísta’ en la cual los algoritmos se constituyen en autoridad última. Encontramos abundantes ejemplos de cómo la vida está siendo progresivamente mediada por algoritmos: seguros de vida que son o no concedidos en función del riesgo de enfermedad calculado por un algoritmo secreto alimentado por los datos de tu tarjeta de crédito (que igual revelan tu compra excesiva de alcohol y tabaco), o por los datos de tus contactos de Facebook (que igual delatan tu afición por los deportes de riesgo); sueldos de conductores que varían en función de los datos de telemetría emitidos por su móvil o GPS y sometidos a un algoritmo que estima la percepción subjetiva de confort del pasajero (Dans, 2016); y un largo etcétera.

El sujeto códigoalfabetizado es capaz de leer y escribir programas informáticos, esto es, capaz de pensar de forma algorítmica. Lejos de aceptar sin más que de un conjunto determinado de datos (*input*) se derive una decisión unívoca (*output*), es conocedor de que la clave se halla en el algoritmo que media entre ambos.

¹⁹⁰ Por ejemplo, el algoritmo que ordena los resultados de búsqueda de Google se encuentra protegido bajo patente y su composición no es públicamente conocida (acaso diríamos que es una versión digital de la fórmula secreta de la Coca-Cola)

Así pues, podría decirse que la capacidad de leer y escribir con los lenguajes informáticos de programación puede servir de antídoto a los aspectos más alienantes de la sociedad digital. Estar codigoalfabetizado, de alguna manera, nos vacuna frente a ciertos vicios asociados a lo digital. En esta línea se manifiesta Douglas Rushkoff, gurú del movimiento ‘code-literacy’ y uno de los inspiradores de esta tesis doctoral, es su ensayo “*Program or be programmed*” (Rushkoff, 2010). Dejemos hablar al autor con su propia voz:

“La tecnología digital está programada. Ello hace que esté sesgada hacia aquéllos con la capacidad de escribir el código. En la era digital, debemos aprender cómo escribir el software, o corremos el riesgo de volvernos como el software mismo (asimilarnos a él). No es demasiado difícil, y no es demasiado tarde para, aprender el código que hay detrás de los objetos digitales que utilizamos; o, al menos, para comprender que de hecho hay código detrás de las pantallas (interfaces). De otra manera, estamos a merced de aquéllos que programan, de la gente que les paga, o incluso de la tecnología misma (...) La programación, el código en sí mismo, es el lugar desde donde emergen las innovaciones más significativas de la era digital (...) Programar es el ‘punto dulce’, el máximo nivel de la sociedad digital. Si no aprendemos a programar, corremos el riesgo de ser programados” (Rushkoff, 2010, p. 134-139)

Pero no queremos caer en una absoluta complacencia respecto del sujeto codigoalfabetizado. También hay algunas voces que alertan de la agenda oculta tras el movimiento ‘code-literacy’. En esa línea, Williamson (2015) afirma que, tras la exhortación lanzada a los niños de todo el mundo de ‘¡aprende a programar!’ (‘learn to code!’), hay un currículum oculto cuyo objetivo no confeso es que las nuevas generaciones interioricen que, para cualquier problema (incluidos los problemas políticos y sociales), puede encontrarse una solución algorítmica que puede ser programada y automatizada, solución que será óptima si el algoritmo es alimentado con la cantidad suficiente de datos. De esta manera, las nuevas generaciones asumirán, como parte de sus deberes ciudadanos, proveer de datos a estos sistemas de ‘gobernanza algorítmica’; algunos de los cuales ya no residen en los estados mismos, sino en corporaciones privadas. Morozov (2015) también ha advertido sobre lo que él llama la ‘locura del solucionismo’, una premisa que se va inoculando en la sociedad actual y que sostiene que pueden encontrarse soluciones óptimas para cualquier problema dado, si éste se formula en términos algorítmicos. Nosotros nos distanciamos de ambos autores; creemos que el sujeto que ha sido codigoalfabetizado de manera crítica, lejos de ser un fanático de los algoritmos, sabrá distinguir cuándo, cómo, por qué y para qué utilizarlos.

En síntesis, diremos que el sujeto plenamente codigoalfabetizado debe integrar armónicamente todas las dimensiones apuntadas: una dimensión laboral, profesional, y económica; una dimensión ética; y una dimensión sociocrítica y empoderadora. En consonancia, los procesos educativos de codigoalfabetización deberán contemplar todas ellas.

2.5.2.1. Breve aproximación desde la perspectiva de género

No queremos cerrar nuestro análisis del sujeto codigoalfabetizado sin hacer algunas breves consideraciones desde la perspectiva de género.

Desde los comienzos de la informática, siempre ha existido una preocupación persistente por el bajo porcentaje relativo de mujeres en los estudios universitarios relacionados con la computación, y por la escasa presencia femenina en los contextos profesionales punteros donde se diseña y escribe el software (Frenkel, 1990). Todavía hoy, inspirar, ‘reclutar’ y retener a las mujeres en las carreras profesionales relacionadas con la programación y la computación, sigue siendo un reto pendiente (Klawe, Whitney, & Simard, 2009). Por ejemplo, el reciente documental “*CODE: Debugging the Gender Gap*”¹⁹¹ (Hauser-Reynolds, 2015) cifra el desequilibrio de género en compañías como Apple, Google, Twitter o Facebook en el entorno de 85% masculino vs. 15% femenino.

Una primera crítica que se ha realizado a este estado de cosas desde la perspectiva de género es que la enseñanza-aprendizaje de la programación se ha venido tradicionalmente realizando, ya desde la época de Logo en los años 80, en entornos de solución de problemas muy vinculados a aptitudes de razonamiento lógico, y sesgados hacia una aproximación viso-espacial de la programación, con ejercicios que consistían en muchos casos en codificar un diseño de tipo geométrico (Burnett *et al.*, 2011). Hay abundante evidencia desde la psicología y la pedagogía diferencial de que estas aptitudes espaciales y de razonamiento lógico tienden a estar más desarrolladas entre los chicos (p.e. Caselles Pérez, 1997; Linn & Petersen, 1985; Voyer, Voyer, & Bryden, 1995); por tanto, entornos de programación de este tipo han venido favoreciendo al perfil aptitudinal masculino y, a la postre, han devenido en la asunción de que programar ‘se les da mejor a los chicos que a las chicas’.

Desde la perspectiva de género, se ha venido aportando una nueva oleada de investigación que balancea las anteriores premisas. Así, cuando la codigoalfabetización se realiza en entornos en los cuales hay que programar una narración (*‘story-telling’*) o un juego interactivo, que implican estructuras lingüísticas y verbales, se encuentra evidencia de que las chicas escriben secuencias de código de mayor complejidad (Howland & Good, 2015; Robertson, 2012). Ello igualmente tiene sentido desde la psicología y la pedagogía diferencial cuyo *corpus* clásico de investigación afirma la superioridad femenina en la aptitud verbal (Caselles Pérez, 1997; Hyde & Linn, 1988)

Por tanto, parece que la lectoescritura de código no tiene en sí misma un sesgo masculino o femenino; más bien, el sesgo aparece en función del contexto sobre el que se proyecte dicha lectoescritura. En la medida que los contextos de codigoalfabetización se han ido diversificando y permitiendo la programación de artefactos digitales más variados (ya no sólo diseños geométricos y ‘programas puramente lógicos’; sino también historias y animaciones digitales), igualmente ha ido emergiendo un talento computacional más diverso y transversal a ambos géneros.

Un claro ejemplo de entorno de codigoalfabetización especialmente indicado para las chicas, es la plataforma ‘*Made with Code*’¹⁹², promovida desde Google¹⁹³. En la plataforma ‘*Made with Code*’ las chicas pueden aprender la lectoescritura de código sobre contextos-proyectos más significativos y más cercanos a sus intereses habituales; por ejemplo, escritura de código para: crear música, diseños de patrones de moda, creación de avatares, composición de coreografías, etc. Además de estos contextos femeninos, la plataforma se completa con el apoyo de mentoras expertas que sirven simultáneamente de ayuda y de modelo para las niñas programadoras (Figura 2.33)


¹⁹¹ Tráiler disponible en <https://vimeo.com/136884902>

¹⁹² <https://www.madewithcode.com/>

¹⁹³ Google ha promovido la realización del documental “*CodeGirl*” (Chilcott, 2015) [<http://www.codegirlmovie.com/>]

PROJECTS

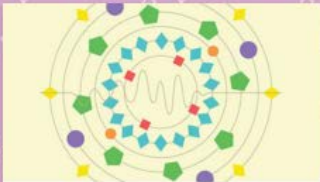
Check out some of the amazing things you can do with code.



INSIDE HQ

Help Riley from Disney Pixar's Inside Out solve some of life's little problems with the help of her Emotions.


[START SOLVING](#)



MUSIC MIXER

EDM, country, or hip hop? Grab some code and mix away.


[MIX IT UP](#)



ACCESSORIZER

Take your selfies to the next level with this project.

[ACCESSORIZE NOW](#)




LED DRESS

Design a ZAC Zac Posen dress that turns heads and lights up a room.

[GET STYLING](#)

MENTORS


Meet the dreamers, creators, and hackers whose code is changing the world.



SEWN W/ CODE

MADDY MAXEY


The future of fashion is bright. Literally.



DANCED W/ CODE

MIRAL KOTSB

The intersection of dance and code is on Broadway.



HEALED W/ CODE

BRITTANY WENGER

Head in the cloud and saving lives.

Figura 2.33. Proyectos y Mentoras de la plataforma 'Made with Code' de Google [<https://www.madewithcode.com/>]

Precisamente, una segunda crítica desde la perspectiva de género ha sido la ausencia de modelos femeninos de excelencia en el campo de la programación; y los abundantes estereotipos masculinos asociados a la figura del programador (hombre joven, caucásico, con gran capacidad lógica y pocas habilidades sociales; a veces considerado peyorativamente como *'nerd'*). Dicha combinación ha venido impidiendo que las chicas, incluso las más dotadas e interesadas, hayan desarrollado carreras e identidades profesionales sostenidas en programación y computación (Soe & Yakura, 2008)

Como réplica, desde la perspectiva de género se ha potenciado cierta investigación en entornos *'single-sex'*, en los que grupos exclusivamente formados por chicas aprenden a programar sin que éstas vean comprometida su imagen femenina por los estereotipos comentados anteriormente. El resultado de este reciente *corpus* de investigación va en el sentido de afirmar la bondad de los entornos *'single-sex'* para la codigoalfabetización de chicas, especialmente en edades *'middle-school'* (6º Primaria, 1º y 2º ESO), y trabajando sobre proyectos del tipo narraciones digitales y juegos interactivos (Denner, Werner, & Ortiz, 2012; Kelleher, Pausch, & Kiesler, 2007)

En la línea de ir modificando los estereotipos asociados a la figura del programador, para que ésta incluya una dimensión femenina, y de proveer modelos femeninos de excelencia en el área tecnológica en general, e informática en particular; vienen conformándose en los últimos años grupos como *'Girls who code'*¹⁹⁴, *'Black girls code'*¹⁹⁵, *'Women who code'*¹⁹⁶ o *'Women of Silicon Valley'*¹⁹⁷. Por ejemplo, *'Girls who code'* se declara como una organización sin ánimo de lucro cuyo objetivo es trabajar por el cierre de la brecha de género en el sector tecnológico en general, y en el de las ciencias de la computación en particular; para ello, *'Girls who code'* promueve acciones para educar, inspirar y equipar a chicas *'high-school'* (3º ESO a 2º Bachillerato) con las habilidades y recursos necesarios para perseguir sus sueños en el campo de la programación (Figura 2.34)



Figura 2.34. Ejemplo de cartel promocional¹⁹⁸ de *'Girls who code'* (con apoyo de Microsoft)

¹⁹⁴ <http://girlswhocode.com/>

¹⁹⁵ <http://www.blackgirlscode.com/>

¹⁹⁶ <https://www.womenwhocode.com/>

¹⁹⁷ <https://medium.com/@WomenOfSiliconValley>

¹⁹⁸ Se puede consultar una colección de 14 carteles de *'Girls who code'* en http://www.huffingtonpost.com/entry/girls-who-code_us_55ad5925e4b065dfe89f1b41

A través de los medios de comunicación hemos tenido noticia de algunos artefactos digitales programados enteramente por chicas, que han sido empoderadas por los anteriores colectivos. Así, Andrea Gonzales y Sophie Houser, dos chicas neoyorquinas de 17 años, tras participar en un programa estival de codigoalfabetización promovido desde ‘*Girls who code*’, han creado un videojuego llamado ‘*Tampon Run*’¹⁹⁹ con la idea de combatir el tabú de la menstruación femenina (Madrid, 2015). Cinco chicas de 14 años de Bangalore (India) han diseñado una ‘*app*’ para facilitar el reciclaje de residuos en su comunidad (Elder, 2015). Un último ejemplo proveniente de Brasil, en el que un grupo de jóvenes programadoras de entre 18 y 22 años han creado una aplicación a través de la cual las mujeres valoran el nivel de seguridad de distintas avenidas y calles de las ciudades brasileñas (Zeilinger, 2015).

En síntesis, una codigoalfabetización diversa e inclusiva debe garantizar la equidad de género en los contextos de aprendizaje de la programación. Ello redundará en un acervo de artefactos digitales que serán expresión y respuesta a necesidades tanto de mujeres como de hombres del siglo XXI.

2.6. Enunciación final de hipótesis directivas

Así pues, tras lo dicho en los apartados previos de este capítulo, consideramos que nuestras hipótesis directivas han quedado suficientemente fundamentadas, y pueden ser enunciadas finalmente:

- ✓ **Hipótesis de Adecuación (H_A):** introducir la codigoalfabetización en nuestro sistema educativo es *adecuado*, pues implica incorporar un *contenido* de aprendizaje que desarrolla aptitudes, habilidades y actitudes valiosas.
- ✓ **Hipótesis de Viabilidad (H_V):** introducir la codigoalfabetización en nuestro sistema educativo es *viable*, pues implica un *proceso* de aprendizaje actualmente abordable a través de metodologías y múltiples recursos didácticos soportados tecnológicamente en línea, como MOOCs o ‘*apps*’ específicos para el aprendizaje de la programación; todo ello en un contexto político favorable a nivel de la UE para la introducción del ‘*coding*’ en los currículos.
- ✓ **Hipótesis de Relevancia (H_R):** introducir la codigoalfabetización en nuestro sistema educativo es *relevante*, pues implica generar *productos* de aprendizaje de creciente uso, utilidad y valoración en nuestra sociedad (‘*apps*’, animaciones digitales, videojuegos, webs, etc...), que pueden ser compartidos con audiencias reales y globales, y ser remezclados por éstas; además el resultado final, el sujeto codigoalfabetizado, es un ciudadano y profesional relevante en la sociedad digital.

2.7. Síntesis: del ‘*Learn to Code*’ al ‘*Code to Learn*’

A modo de epílogo de este capítulo, y como nexo de transición con el siguiente, rescatamos un aforismo que ha ido ganando predicamento en el devenir reciente del movimiento de codigoalfabetización. Se trata del lema ‘*Learn to code, code to learn*’ (Resnick, 2013b), así expresado en un famoso artículo en la blogosfera por Mitch Resnick, director del grupo Lifelong Kindergarten en el MIT Media Lab, y artífice del lenguaje Scratch. Dicho lema sintetiza el nuevo paradigma bajo el cual lo principal no es ‘aprender a programar’ sino ‘programar para aprender’;

¹⁹⁹ Disponible en la App Store: <https://itunes.apple.com/app/tampon-run/id948757853>

dicho de otra manera, se desplaza el foco desde el código como objeto de aprendizaje en sí mismo, hacia el código como medio para aprender y desarrollar otras habilidades. Citando a Resnick (2013b, en línea):

“¿Es importante que todos los niños aprendan a escribir? Después de todo, muy pocos de esos niños llegarán a ser periodistas, novelistas, o escritores profesionales. Entonces, ¿por qué deberían todos los niños aprender a escribir? Naturalmente, estas preguntas nos parecen tontas. Las personas utilizamos la escritura en muchos ámbitos de nuestra vida (...) El acto de escribir también involucra distintas formas de pensamiento. En la medida que escribimos, aprendemos a organizar, refinar, y reflejar nuestras ideas. Claramente, ésas son poderosas razones para que todos aprendamos a escribir (...)

Considero la programación informática ('coding') como una extensión de la escritura. La capacidad de programar te permite 'escribir' nuevos tipos de artefactos (historias interactivas, juegos, animaciones, y simulaciones). Al igual que con la escritura tradicional, hay poderosas razones para que todos los niños aprendan a programar (...)

El reciente surgimiento del interés masivo por 'aprender a programar' se ha focalizado especialmente en las oportunidades profesionales (...) pero considero que hay razones más profundas para ello. En el proceso de aprender a programar, las personas aprenden muchas otras cosas. No sólo están 'aprendiendo a programar', sino que están 'programando para aprender' (...)

Programando para aprender no sólo conceptos matemáticos y computacionales (como variables y condicionales), sino también estrategias de resolución de problemas, diseño de proyectos, y comunicación de ideas. Todas estas habilidades son útiles no sólo para los científicos de la computación sino para cualquier persona de la sociedad digital, sin importar su edad, su profesión o sus intereses” (Resnick, 2013b, en línea)

En síntesis, el nuevo enfoque pone el acento no sólo en la lectoescritura de código por sí misma, cuyas sintaxis particulares cambian con la evolución misma de los lenguajes informáticos; sino en los procesos de pensamiento subyacentes: lo que se ha venido a denominar 'pensamiento computacional' (PC), al que dedicamos el próximo capítulo.

Haciendo un símil, si el mundo globalizado implicó la transición del 'aprender inglés' (como objeto separado de estudio) al 'inglés para aprender' (como lengua vehicular del resto de asignaturas); el mundo digitalizado ha hecho lo propio con el código, desde objeto estático de aprendizaje a meta-lenguaje dinámico que vehicula la construcción y expresión de todo tipo de artefactos digitales (Figura 2.35).

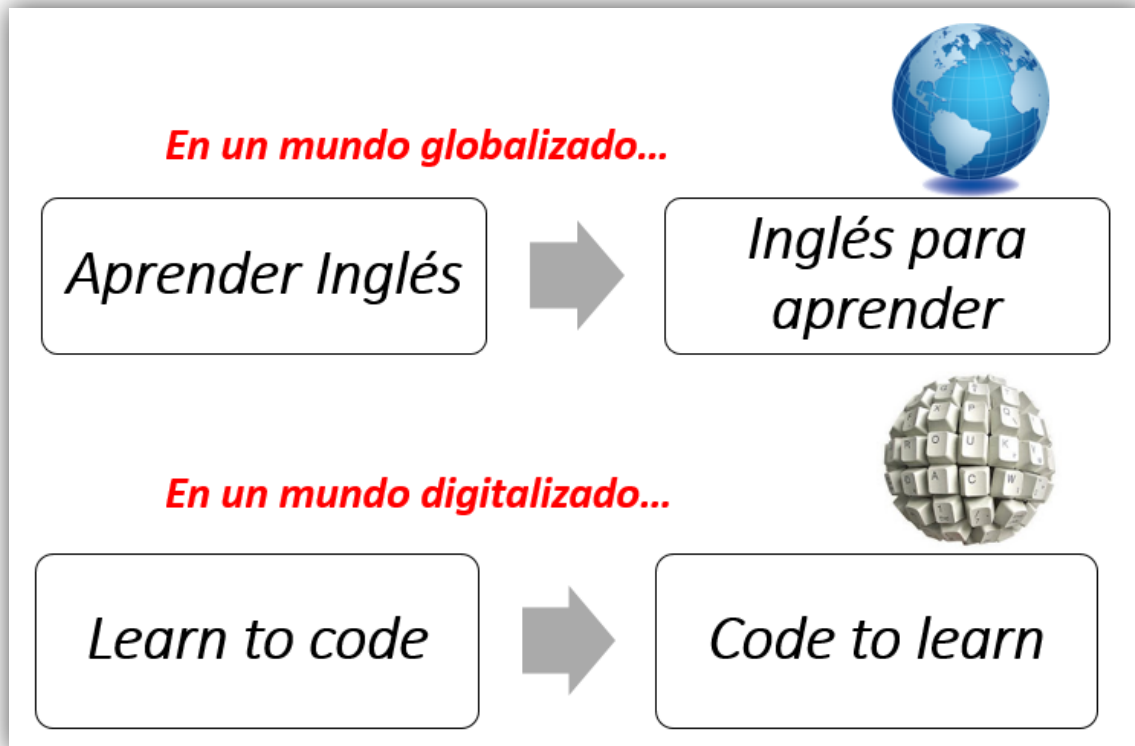


Figura 2.35. Símil entre el mundo globalizado y el mundo digitalizado

CAPÍTULO 3

PENSAMIENTO COMPUTACIONAL

En la introducción de esta tesis doctoral ya definimos la codigoalfabetización (del inglés ‘*code-literacy*’) como el proceso de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación. Así, se considera que una persona está codigoalfabetizada cuando es capaz de leer y escribir en el lenguaje de los ordenadores y otras máquinas, y de pensar computacionalmente (Román-González, 2014a). Por tanto, si la codigoalfabetización alude en última instancia a una nueva práctica de lectoescritura, el pensamiento computacional (PC) se refiere al proceso cognitivo subyacente de resolución de problemas que le da soporte: son dos caras de una misma moneda (Figura 3.1). Si el Capítulo 2 ha sido dedicado íntegramente a la codigoalfabetización, como un nuevo alfabetismo crítico en la actual sociedad digital; este Capítulo 3 se focaliza en el pensamiento computacional, como emergente constructo psicológico susceptible de ser definido, desarrollado, medido y evaluado.

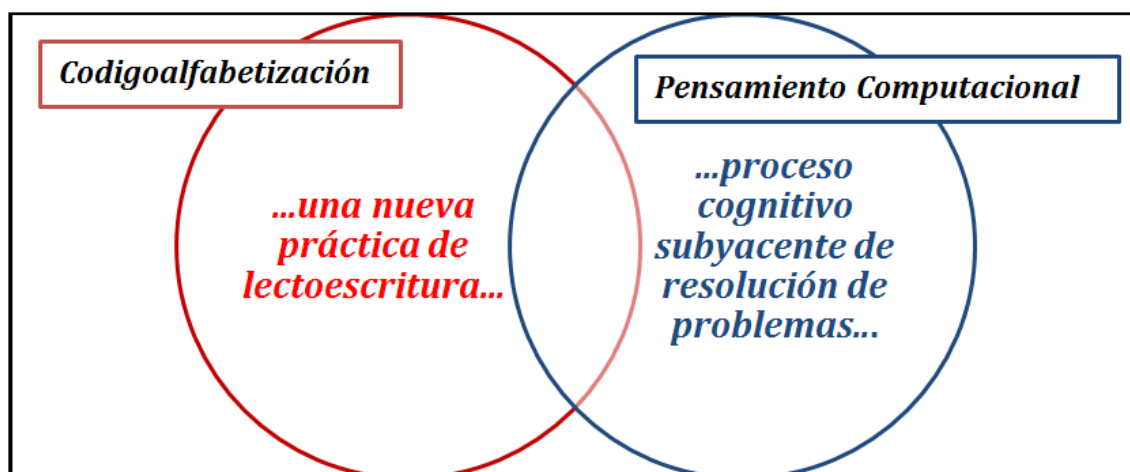


Figura 3.1. Codigoalfabetización y Pensamiento Computacional: dos caras de una misma moneda

En la nueva realidad que vivimos, invadida por lo digital, no es sorprendente que haya un interés renovado en muchos países por introducir el PC como un conjunto de habilidades de solución de problemas que debe ser adquirido por la nueva generación de estudiantes; aún más, el PC se está empezando a considerar como un elemento central de todas las disciplinas STEM: acrónimo de ‘*science, technology, engineering & mathematics*’, es decir, es decir ‘ciencia, tecnología, ingeniería y matemáticas’ (Henderson, Cortina & Wing, 2007). Aunque aprender a pensar computacionalmente viene siendo reconocido como relevante desde hace largo tiempo (p.e. Papert, 1980, cuando utiliza el término ‘*procedural thinking*’), en la medida que la programación y la computación se han convertido en omnipresentes, acciones que sustentan la comunicación, la ciencia, la cultura y los negocios en nuestra sociedad digital (Howland & Good, 2015); el PC es progresivamente visto como una habilidad esencial que nos posibilita crear, en vez de sólo consumir, tecnología (Resnick *et al.*, 2009, Zapata-Ros, 2015)

Sin embargo, todavía hay poco consenso sobre una definición formal del PC, y discrepancias sobre cómo debería ser integrado en los currículos educativos (Gouws, Bradshaw & Wentworth, 2013a). A pesar de todas las definiciones formuladas desde 2006, año en que se publica el artículo fundacional de la disciplina (Wing, 2006), se reconoce que aún no existe una idea clara sobre cómo incorporar el PC a los sistemas educativos en sus distintas etapas; existiendo una enorme variedad y heterogeneidad de intervenciones educativas al respecto (Lye & Koh, 2014). Igualmente, hay un enorme vacío sobre cómo medir y evaluar el PC, hecho que debe ser abordado. Y es que sin la atención suficiente sobre la medida y evaluación del PC, será muy difícil que éste se abra camino exitosamente en cualquier currículum. Aún más, para poder juzgar la efectividad de cualquier currículum que incorpore el PC, será necesario validar previamente instrumentos de medida que permitan a los educadores evaluar en qué grado los estudiantes han desarrollado esa habilidad de pensamiento (Grover & Pea, 2013a). Este vacío será el que anime el diseño y validación del ‘Test de Pensamiento Computacional’ (TPC), que se presentará en el Capítulo 6, ya en la parte empírica.

Por el momento, la presente revisión teórica se organizará en tres apartados: uno dedicado al concepto de ‘pensamiento computacional’ y las distintas definiciones que del mismo se han ido formulando; otro apartado focalizado en modelos, experiencias y recursos para el desarrollo del pensamiento computacional; y un último dedicado a su medida y evaluación.

3.1. Concepto

Para abordar el concepto de ‘pensamiento computacional’ procederemos a revisar las distintas definiciones que se han ido ofreciendo del mismo. Diferenciaremos en los siguientes epígrafes entre definiciones genéricas; operativas u operacionales; psicológico-cognitivas; y educativo-curriculares.

3.1.1. Definiciones genéricas

Hace una década, Jeannette M. Wing (2006)²⁰⁰ publicó el artículo fundacional de la disciplina, en el cual define que el *“pensamiento computacional implica la resolución de problemas, el diseño de sistemas, y la comprensión de la conducta humana, haciendo uso de los conceptos fundamentales de la informática”* (Wing, 2006, p. 33). Así, la esencia del PC sería pensar como un científico de la computación cuando uno se enfrenta a un problema.

En dicho artículo, Wing afirma igualmente que el pensamiento computacional *“representa un conjunto de habilidades y actitudes, aplicable universalmente; que toda persona (no sólo los científicos de la computación) debería estar ansiosa por adquirir y usar”* (Wing, 2006, p. 33). Así, la autora define el PC como una habilidad básica y fundamental para cualquier sujeto inmerso en la actual realidad digital, llegando incluso a afirmar que el PC debería añadirse a la ‘lectura, escritura, aritmética’ como paquete básico en el desarrollo de las habilidades analítico-instrumentales de cualquier niño.

Sintetizamos a continuación qué es, y qué no es, pensamiento computacional en esta aproximación pionera de Wing (2006):

²⁰⁰ Jeannette M. Wing es Jefa del Departamento de Ciencias de la Computación en la Universidad Carnegie Mellon (Pittsburgh, PA)

- *‘Conceptualizing, not programming’* (‘conceptualizar, no programar’): pensar como un científico de la computación, es decir, pensar computacionalmente, significa más que ser capaz de programar un ordenador. Requiere pensar en múltiples niveles de abstracción.
- *‘Fundamental, not rote skill’* (‘habilidad básica, no puramente mecánica’): una habilidad básica es aquella que cualquier persona debe dominar para funcionar en la sociedad actual.
- *‘A way that humans, not computers, think’* (‘una manera en que los humanos piensan, no las computadoras’): el pensamiento computacional es una manera en que los seres humanos solucionan problemas; no es tratar de que los humanos piensen como las computadoras. Las computadoras son previsibles y aburridas; los humanos, inteligentes e imaginativos. Somos los humanos los que hacemos a las computadoras interesantes. Los humanos, equipados con la potencia de las computadoras, usamos nuestra inteligencia para abordar y resolver problemas que no seríamos capaces de atacar en la era previa a la computación.
- *‘Complements and combines mathematical and engineering thinking’* (‘se complementa y se combina con el pensamiento matemático e ingenieril’): las ciencias de la computación y, por tanto, el pensamiento computacional, descansan sobre el pensamiento matemático dado que, como todas las ciencias, sus bases formales surgen de las matemáticas. Además, las ciencias de la computación descansan inherentemente en el pensamiento ingenieril dado que trata de construir sistemas informáticos que interactúan con el mundo real-físico.
- *‘Ideas, not artifacts’* (‘ideas, no artefactos-objetos’): el pensamiento computacional no tiene sólo que ver con los objetos de hardware y software que producimos, y que están presentes físicamente a diario en nuestras vidas; tiene que ver sobre todo con los conceptos computacionales que usamos para aproximarnos a los problemas y solucionarlos, para gestionar nuestra vida, y comunicarnos e interactuar con otras personas de manera efectiva.
- *‘For everyone, everywhere’* (‘para cualquiera, en cualquier parte’): el pensamiento computacional será una realidad cuando sea tan inherente a las tareas y esfuerzos humanos, que desaparezca como término y filosofía explícitos.

Esta primera definición genérica, casi metafísica, del pensamiento computacional, viene siendo revisada y especificada en intentos sucesivos a lo largo de los últimos años, sin llegar aún a un acuerdo generalizado sobre la misma. Veámoslo.

La propia Wing revisita su definición algo más tarde (2008), clarificando que “*el pensamiento computacional incluye los procesos de pensamiento implicados en la formulación de problemas y de sus soluciones, de tal modo que éstos estén representados de una manera que pueda ser abordada efectivamente por un agente-procesador de información*” (Wing, 2008, p. 3718), como un ordenador. Siguiendo con este grupo de definiciones, de corte genérico, cuatro años más tarde Aho (2012, p. 832) declara que el PC es “*el proceso de pensamiento involucrado en la formulación de problemas de tal manera que sus soluciones puedan ser representadas como pasos computacionales discretos y algoritmos*”. En ese mismo año, la Royal Society (2012) también aporta una definición sucinta que trata de capturar la esencia del PC:

“El pensamiento computacional es el proceso de reconocimiento de los aspectos computables en el mundo que nos rodea, y de aplicar las herramientas y técnicas de las Ciencias de la Computación para comprender y razonar sobre sistemas y procesos, tanto naturales como artificiales” (Royal Society, 2012, p. 29)

3.1.2. Definiciones operativas

De este grupo de definiciones generales, se va avanzando progresivamente a definiciones más operativas u operacionales, que tratan de enumerar los elementos constitutivos del pensamiento computacional. Así, por ejemplo, la Fundación Nacional para la Ciencia de los Estados Unidos (*‘National Science Foundation’*) en el marco de su curso *‘CS Principles’*²⁰¹, cuyo objetivo es fijar y transmitir las bases de las Ciencias de la Computación al alumnado de Bachillerato y primeros años de universidad, define las siguientes siete ideas como esenciales del pensamiento computacional ([National Science Foundation, 2015, en línea](#)):

- i. *“El pensamiento computacional es una actividad humana creativa.*
- ii. *La abstracción (uno de los elementos constitutivos, sino el central, del pensamiento computacional) reduce-elimina la información y detalles irrelevantes para focalizarse en los conceptos relevantes a la hora de entender y resolver un problema.*
- iii. *Los datos y la información facilitan la creación de conocimiento.*
- iv. *Los algoritmos son herramientas para desarrollar y expresar soluciones a problemas computacionales.*
- v. *Programar es un proceso creativo que produce artefactos-objetos computacionales.*
- vi. *Los dispositivos y sistemas digitales, y las redes que los interconectan, posibilitan y potencian una aproximación computacional a la resolución de problemas.*
- vii. *El pensamiento computacional permite la innovación en otros campos, incluyendo las ciencias naturales, ciencias sociales, humanidades, artes, medicina, ingeniería, y negocios.”*

Esta línea de especificar los elementos que constituyen el pensamiento computacional culmina con la *‘Operational Definition of Computational Thinking for K-12 Education’*: una definición operativa del PC que sirve de marco de trabajo y vocabulario compartido para los profesores de informática (*‘Computer Science Teachers’*) en las etapas de Educación Secundaria y preuniversitaria estadounidense. Esta definición operativa fue desarrollada inicialmente en 2011 por la *‘Computer Science Teachers Association (CSTA)’* y la *‘International Society for Technology in Education (ISTE)’* de los Estados Unidos; y sigue plenamente vigente en la actualidad. Así, la CSTA & ISTE ([2015, en línea](#)) dicen que *“el pensamiento computacional es un proceso de solución de problemas que incluye (aunque no está limitado a) las siguientes características:*

- *Formular problemas de un modo que se haga posible utilizar un ordenador y otras máquinas en su resolución.*
- *Organizar lógicamente y analizar datos*
- *Representar datos a través de abstracciones tales como modelos y simulaciones*
- *Automatizar soluciones a través del pensamiento algorítmico (una serie de pasos discretos y ordenados)*

²⁰¹ <http://apcsprinciples.org/>

- *Identificar, analizar e implementar posibles soluciones con el objetivo de lograr la combinación más efectiva y eficiente de pasos y recursos*
- *Generalizar y transferir este proceso de solución de problemas a una amplia variedad de situaciones.”*

La novedad que aporta esta definición operativa (CSTA & ISTE, 2015, en línea) es añadir una dimensión actitudinal al pensamiento computacional, pues continua diciendo: “*estas habilidades son apoyadas y reforzadas por una serie de disposiciones o actitudes que son también dimensiones esenciales del PC. Estas disposiciones y actitudes incluyen:*

- *Confianza al manejarse con la complejidad*
- *Persistencia al trabajar con problemas difíciles*
- *Tolerancia a la ambigüedad*
- *Capacidad de hacer frente a problemas abiertos (sin una solución concreta y evidente)*
- *Capacidad de comunicarse y trabajar con otros para llegar a una meta-solución común”*

En el marco de estas definiciones operativas, que suelen enunciarse como listados de elementos que componen el pensamiento computacional, ha surgido el debate entre los investigadores, académicos y expertos, sobre cuál es el elemento central (si lo hay), de mayor jerarquía, del PC. Siguiendo la revisión de Grover & Pea (2013a), la ‘abstracción’ sería la clave de bóveda del pensamiento computacional. Abstracción es “*definir patrones, generalizar a partir de casos concretos; y es la clave para lidiar con la complejidad*” (Grover & Pea, 2013a, p. 39).

En cualquier caso, y como síntesis de las distintas definiciones operativas que han ido apareciendo estos últimos años, Grover & Pea (2013a, p. 39-40) recogen “*los elementos que han conseguido un cierto consenso, y amplia aceptación, para formar parte del PC; y que deberían estar en la base de cualquier currículum que pretenda su desarrollo:*

- *Abstracción y generalización de patrones (incluyendo modelos y simulaciones)*
- *Procesamiento sistemático de la información*
- *Sistemas de símbolos y representación*
- *Noción algorítmica de control de flujo*
- *Descomposición estructurada de problemas (modularización)*
- *Pensamiento iterativo, recursivo y paralelo*
- *Lógica condicional*
- *Limitadores de eficiencia y rendimiento*
- *Depuración y detección sistemática de errores”*

3.1.3. Definiciones psicológico-cognitivas

Desde otro punto de vista, el grupo de investigación educativa de Google o ‘*Google for Education*’ ofrece una definición del pensamiento computacional como un ciclo de cuatro pasos-procesos cognitivos. Más concretamente, dicen que el PC es “*un conjunto de habilidades y técnicas de solución de problemas que los ingenieros de software usan para escribir los programas informáticos que subyacen a las aplicaciones que usamos a diario*” (Google for Education, 2015, en línea). Sin

embargo, continua algo más adelante la definición, “*el pensamiento computacional es aplicable a prácticamente cualquier ámbito (...) e incluye las siguientes 4 fases específicas* (Figura 3.2):

- I. *Descomposición de un problema o tarea en pasos discretos*
- II. *Reconocimiento de patrones (regularidades)*
- III. *Generalización de dichos patrones y abstracción (descubrir las leyes o principios que causan dichos patrones)*
- IV. *Diseño algorítmico (desarrollar instrucciones precisas para resolver el problema y sus análogos)”* (Google for Education, 2015, en línea)

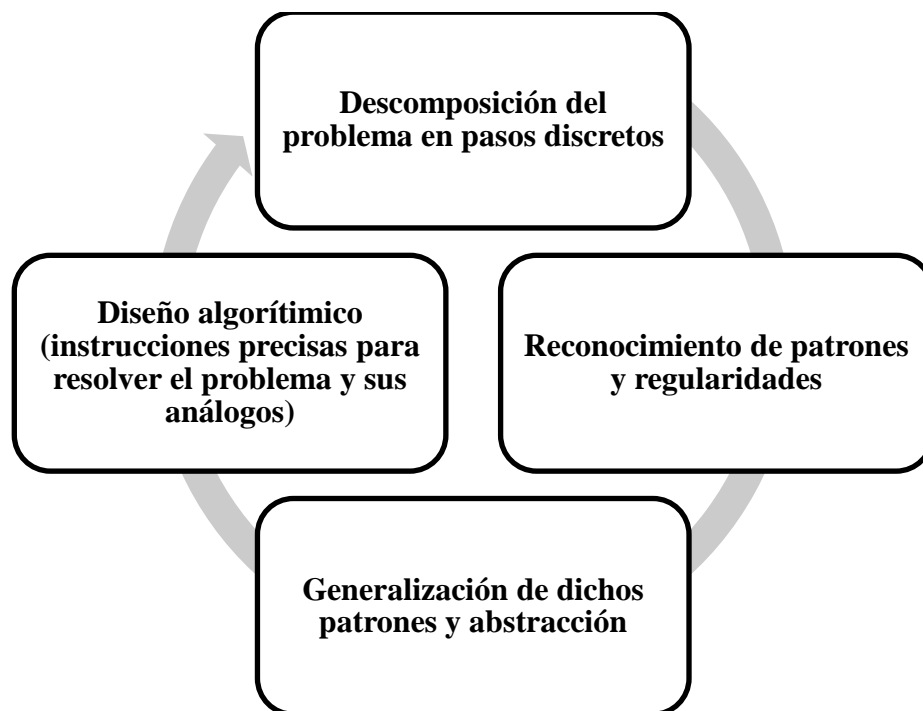


Figura 3.2. Las 4 fases-pasos cognitivos del pensamiento computacional

Veamos con algo más de detalle cada uno de estos 4 pasos, ejemplificando su aparición en distintos tipos de problemas para ilustrar cómo el PC se puede proyectar sobre distintos ámbitos (Google for Education, 2015, en línea):

- I. **Descomposición:** o la capacidad para fraccionar una tarea minuciosa y detalladamente en los pasos que la forman, de manera que luego podamos explicar unívocamente el proceso a una tercera persona o a un ordenador, o incluso como notas de procedimiento para uno mismo. Descomponer un problema frecuentemente conduce a un posterior reconocimiento y generalización de patrones, y así en última instancia a la capacidad para diseñar un algoritmo.

Ejemplos:

- a. Cuando probamos un nuevo plato de cocina e identificamos los distintos ingredientes que dan forma a su sabor, estamos *descomponiendo* el plato en sus ingredientes particulares.

- b. Cuando le damos a alguien indicaciones precisas para llegar a nuestro domicilio (p.e. “sales de la boca del metro, giras a la derecha, caminas recto y coges la segunda calle a la izquierda...”), estamos *descomponiendo* el proceso de ‘ir de un sitio a otro’
- c. En matemáticas, podemos *descomponer* un número, como 256,37, de la siguiente manera: $2*10^2 + 5*10^1 + 6*10^0 + 3*10^{-1} + 7*10^{-2}$

II. Reconocimiento de patrones: o la capacidad para percibir similitudes o diferencias comunes que nos ayudan a hacer predicciones y nos conducen hacia ‘atajos’ o ‘accesos directos’ (*shortcuts*) al núcleo de un problema. El reconocimiento de patrones es frecuentemente la base para el diseño algorítmico y la resolución de problemas.

Ejemplos:

- a. Los niños identifican *patrones* en las reacciones de sus padres y profesores a su comportamiento, en orden a establecer qué está bien y qué está mal. Y basan su comportamiento futuro en función de dichos patrones.
- b. Los corredores de bolsa buscan *patrones* en el valor de las acciones para decidir cuándo comprar y cuándo vender.
- c. En matemáticas, podemos seguir un *patrón* para explicar la lógica que subyace a que el producto de dos números negativos es un número positivo:

$$(-3) * (3) = -9$$

$$(-3) * (2) = -6$$

$$(-3) * (1) = -3$$

$$(-3) * (0) = 0$$

$$(-3) * (-1) = 3$$

$$(-3) * (-2) = 6$$

- d. En geometría, al calcular el mayor área posible para un rectángulo de un perímetro dado, podemos observar *patrones* que involucran el alto, ancho y área del mismo como:
 - ✓ *En la medida que el alto y el ancho se aproximan el uno al otro en sus valores, el área se incrementa.*
 - ✓ *En la medida que aumenta la diferencia entre los valores del alto y del ancho, el área se reduce.*
 - ✓ *Este patrón nos conduce a la conclusión de que el rectángulo con el área mayor es un cuadrado.*

III. Generalización de patrones y abstracción: o la capacidad para filtrar e ignorar toda la información que no es necesaria para resolver un cierto tipo de problema, y de generalizar la que sí es necesaria. La generalización de patrones y la abstracción nos permiten representar una idea o un proceso en sus términos generales-formales (p.e. variables) de manera que

podemos utilizar dicha representación para resolver problemas análogos de similar naturaleza.

Ejemplos:

- a. Un planificador diario o agenda utiliza una *abstracción* para representar la semana en términos de días y horas, ayudándonos a organizar nuestro tiempo
- b. Un mapamundi es una *abstracción* de la Tierra en términos de longitud y latitud, ayudándonos a determinar la localización geográfica de un determinado lugar.
- c. En matemáticas, escribimos fórmulas generalizadas en términos de variables, en vez de con números; de manera que podemos utilizarlas para resolver distintos problemas con valores diferentes: por ejemplo, $[(a+b) * (a-b) = a^2 - b^2]$

IV. Diseño algorítmico: o la capacidad de desarrollar una estrategia ‘paso por paso’ (secuencia de instrucciones perfectamente definida) para resolver un problema. El diseño algorítmico está basado a menudo en la descomposición previa de un problema y en la identificación de los patrones que ayudan en su resolución. En Ciencias de la Computación, así como en matemáticas, los algoritmos suelen escribirse de manera abstracta, utilizando variables en el lugar de números específicos.

Ejemplos:

- a. Cuando un chef de cocina escribe la receta para un plato, está creando un *algoritmo* que otros pueden seguir para replicar el plato.
- b. Cuando un entrenador crea el ‘libro de jugadas’ para su equipo de baloncesto, está diseñando un conjunto de *algoritmos* para que sean seguidos por sus jugadores a lo largo del partido.
- c. En matemáticas, por ejemplo, cuando calculamos el ‘cambio porcentual’ entre dos números, seguimos el siguiente *algoritmo* a lo largo de las siguientes líneas:
 - ✓ Si el número original es mayor que el nuevo número, entonces usar la ecuación siguiente para calcular el ‘cambio porcentual’: $[Disminución\ porcentual = 100 * (original - nuevo) / original]$
 - ✓ Si el nuevo número es mayor que el número original, entonces usar la ecuación siguiente para calcular el ‘cambio porcentual’: $[Incremento\ porcentual = 100 * (nuevo - original) / original]$
 - ✓ Si ninguna de las anteriores es cierta, entonces el número original y nuevo tienen el mismo valor, y el ‘cambio porcentual’ es cero

Podemos ir un paso más allá: una vez formulado correctamente un *algoritmo* (secuencia ordenada y exhaustiva de pasos para llegar unívocamente a la solución de un cierto tipo de problemas), podemos escribirlo en forma de *programa* (conjunto de instrucciones que pueden ser entendidas, procesadas y ejecutadas por un ordenador), a través de un *código* concreto que siga la sintaxis de un lenguaje de programación determinado. Así, el anterior *algoritmo* puede ser convertido en *programa* informático

a través de su *codificación* en, por ejemplo, lenguaje Python²⁰² (Figura 3.3); de manera que ya podemos aprovechar la potencia computacional de nuestro ordenador para resolver rápida y eficazmente el tipo de problema abordado. Así, pues, habríamos conseguido cerrar el círculo del pensamiento computacional: formular el problema en términos de pasos discretos o algoritmos, de manera que podamos apoyarnos luego en un ordenador para su resolución.

```
original = float(input('Enter the original number: '))
new = float(input('Enter the new number: '))
if original > new:
    percent_decrease = 100 * (original - new) / original
    print 'Percent decrease:', percent_decrease, '%'
elif new > original:
    percent_increase = 100 * (new - original) / original
    print 'Percent increase:', percent_increase, '%'
else:
    print 'There is no percent change.'
```

Figura 3.3. Algoritmo del ‘cambio porcentual’ escrito como *programa* informático a través de *código* Python

Otro ejemplo muy ilustrativo del ciclo cognitivo involucrado en el pensamiento computacional, lo extraemos del curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a), sobre el que se focalizará la evaluación de programas del Capítulo 7, ya en la parte empírica de esta tesis doctoral; específicamente de su ‘*Stage 3: Computational Thinking*’²⁰³. Ahí se insta al profesor a introducir el concepto de ‘pensamiento computacional’ entre sus estudiantes a través del siguiente ejemplo:

- “Di a tus estudiantes que quieres que sumen todos los números entre 1 y 200.
 - (Utiliza tu lenguaje corporal para indicarles que no se trata de un examen).
 - ✓ Ahora, hazles saber a tus estudiantes que deberán hacer la suma mentalmente.
 - ✓ Añade la limitación de tiempo a 30 segundos
 - ✓ Seguramente, tus estudiantes se sentirán abrumados por el cálculo solicitado. No te preocupes, es intencionado. Puedes comenzar la cuenta de 30 segundos con un resonante: “Preparados... listos... ya”
- Observa a la clase mientras cronometras el tiempo. ¿Cuántos de tus estudiantes se han perdido ya tratando de hacer la suma mentalmente?
- Cuando se cumpla el tiempo, pregunta si algún estudiante ha sido capaz de llegar al total.
- Pregunta si hay alguno que haya pensado que era tan difícil que ni siquiera lo ha intentado.
- ¿Hay alguno que lo haya intentado y sencillamente no haya llegado hasta el final?
 - ¿Qué estrategia ha seguido en su intento?
- Ahora guía a tus estudiantes por un proceso de pensamiento algo distinto.
 - Si ‘rompemos’ (**descomposición**) el problema en pequeñas partes, será más fácil de manejar y resolver.
 - Comencemos por los extremos: ¿cuánto es $200 + 1$?
 - ✓ ¿Y $199 + 2$?

²⁰² <https://www.python.org/>

²⁰³ <https://studio.code.org/s/20-hour/stage/3/puzzle/1>

- ✓ ¿Y $198 + 3$?
- ¿Veis algo que se repita? (**detección de patrones**).
- ¿Cuántas de estas parejas, sumas de dos dígitos, tenemos?
 - ✓ ¿Cuál es la última pareja que nos sale? $100 + 101$.
 - ✓ Eso significa que tenemos 100 parejas.
 - ✓ Si tenemos un total de 100 parejas, y cada una suma 201, ¿cómo calculamos el total?
 - ✓ ¿Cuánto da $100 * 201$?
- Ahora, ¿cómo hacemos para hallar el truco (la regla de cálculo) con otros números? (**generalización y abstracción**)
 - ✓ ¿Podemos hacerlo fácilmente con la suma de todos los números entre 1 y 2000?
 - ✓ ¿Y entre 1 y 20.000?
 - ✓ ¿Qué se mantiene constante entre ambos casos? ¿qué es diferente?
- Si utilizamos la abstracción para llegar a una expresión final que contenga algún espacio que pueda cambiar (le llamaremos “blank”²⁰⁴), entonces podemos llegar a un algoritmo que funcione para cualquier número (**diseño algorítmico**)
 - ✓ Trabaja con los estudiantes sobre el problema hasta llegar al algoritmo que lo soluciona: **Suma = (“blank”/2) * (“blank”+1)**
 - ✓ Haz varios ejemplos para mostrar que el algoritmo funciona para distintos valores de la variable (para “blank”=2; “blank”=3... “blank=10”... “blank=10.000”...)
- Como síntesis final, dile a tu clase: “Hemos hecho este juego para mostrar que si usas las herramientas mentales del Pensamiento Computacional (descomposición, detección de patrones, abstracción y generalización, y diseño algorítmico), entonces puedes descifrar cómo solucionar problemas que nadie antes te haya enseñado. Esto será una habilidad extraordinariamente potente para el resto de tu vida” [Extraído de ‘K-8 Intro to Computer Science. Unidad 3: Computational Thinking’ (Code.org, 2015a)]

Curiosamente, y en este juego simétrico de espejos con las altas capacidades que viene atravesando toda esta tesis, un ejemplo parecido se cuenta al ilustrar el genio precoz de Carl Friedrich Gauss a la hora de enfrentar problemas matemáticos:

“Carl Friedrich Gauss (1777-1855), gran matemático alemán, fue un niño precoz que destacó sobremanera en las matemáticas. De origen humilde, se cuentan de él anécdotas geniales. Una de ellas es que de niño en el colegio, el profesor puso a los alumnos la tarea de sumar todos los números del 1 a 100. Al cabo de muy poco tiempo Gauss ya tenía la respuesta. Y no porque hubiera sumado muy rápido sino porque razonó de forma muy particular, como tuvo que explicar al sorprendido profesor: $1 + 2 + 3 + \dots + 99 + 100 = (1+100) + (2+99) + (3+98) + \dots + (49+52) + (50+51) = 5050$. La forma general de expresar esta progresión sería: $1+2+\dots+n = (1+n)(n/2)$ ” (Jiménez Fernández, 2014, p. 29)*

²⁰⁴ “blank” es la forma genérica de nombrar una variable en la mayoría de lenguajes informáticos

Parece, pues, que el pensamiento computacional es un proceso cognitivo que aspira a ser desarrollado, en mayor o menor medida, en todos los sujetos (no sólo en los más capaces, aunque precisamente en éstos pueda aparecer de forma espontánea, tal y como se ilustra en el episodio de Gauss). Ello parece ineludible en una época en la cual ser capaz de formular los problemas en términos *algorítmicos* (y, por tanto, susceptibles de ser convertidos luego en *programas* informáticos a través de un determinado *código*) nos pone en situación de aprovechar toda la potencia computacional de nuestros ordenadores, cada vez más presentes, cada vez más veloces.

3.1.4. Definiciones educativo-curriculares

Más que definiciones en sentido estricto, incluimos en este epígrafe un par de modelos o marcos teóricos de integración del pensamiento computacional (*'computational thinking frameworks'*) en el sistema educativo.

a) Un marco desde Reino Unido: el modelo CAS (*'Computing at School'*)

'Computing at School' (CAS)²⁰⁵ es una alianza estratégica formada por el Ministerio de Educación del Reino Unido, diversas universidades e investigadores del país, y empresas del ámbito informático y tecnológico (p.e. Microsoft); cuyo principal objetivo es promover y apoyar la excelencia en la educación en Ciencias de la Computación dentro del sistema británico. Nace en el contexto del diseño e implantación en Reino Unido del nuevo currículum en Ciencias de la Computación ([UK Department of Education, 2013](#)), que introduce, ya desde el curso 2014/2015 y a lo largo de todas las etapas educativas obligatorias (desde los 5 a los 16 años), contenidos relativos a la programación informática, pero siempre tratados desde la perspectiva del pensamiento computacional.

Una de las funciones de CAS es la sensibilización y formación del profesorado para que la integración del nuevo currículum se haga efectiva en las aulas; a través de su comunidad web *'CAS Barefoot'*²⁰⁶. En dicha comunidad se proclama:

“Nuestro objetivo es ayudar a nuestros jóvenes a interiorizar y sentirse confiados con el pensamiento computacional, y a que tomen conciencia de cómo la tecnología y los datos están en la base de nuestra sociedad actual. Para ello, ofrecemos a nuestros profesores esta comunidad con recursos útiles y de alta calidad, unidades didácticas...” ([CAS, 2015, en línea](#))

Pues bien, la comunidad *'CAS Barefoot'* aporta su propia definición de pensamiento computacional, y un marco conceptual genérico para su integración en las aulas:

“El pensamiento computacional es abordar un problema de manera que un ordenador pueda ayudarnos luego en su resolución (...) éste es un proceso de dos fases: primera, pensamos en los pasos que se necesitan para resolver el problema; segunda, utilizamos nuestras habilidades técnicas para poner a nuestro ordenador a trabajar sobre el problema (...) el pensamiento computacional no es pensar sobre ordenadores ni pensar como ordenadores. Los ordenadores no piensan por sí mismos, ¡al menos no todavía!” ([CAS, 2015, en línea](#))

²⁰⁵ <http://www.computingatschool.org.uk/>

²⁰⁶ <http://barefootcas.org.uk/>

A partir de la definición anterior, desde ‘CAS Barefoot’ se define el siguiente modelo conceptual para la integración del pensamiento computacional en el aula. Así, defienden que el pensamiento computacional incluye **6 conceptos** (‘lógica’, ‘algoritmos’, ‘descomposición’, ‘patrones’, ‘abstracción’, y ‘evaluación sistemática’) y **5 aproximaciones** (‘experimentación’, ‘creación’, ‘depuración’, ‘perseverancia’, y ‘colaboración’). Se detalla en la Figura 3.4, póster que este doctorando ha visto personalmente colgado en múltiples aulas del Reino Unido a lo largo de su estancia de investigación durante el verano de 2015. Veamos cada uno de los conceptos y aproximaciones propuestos con algo más de detalle, en los términos en que aparecen expuestos originalmente (CAS, 2015, en línea)

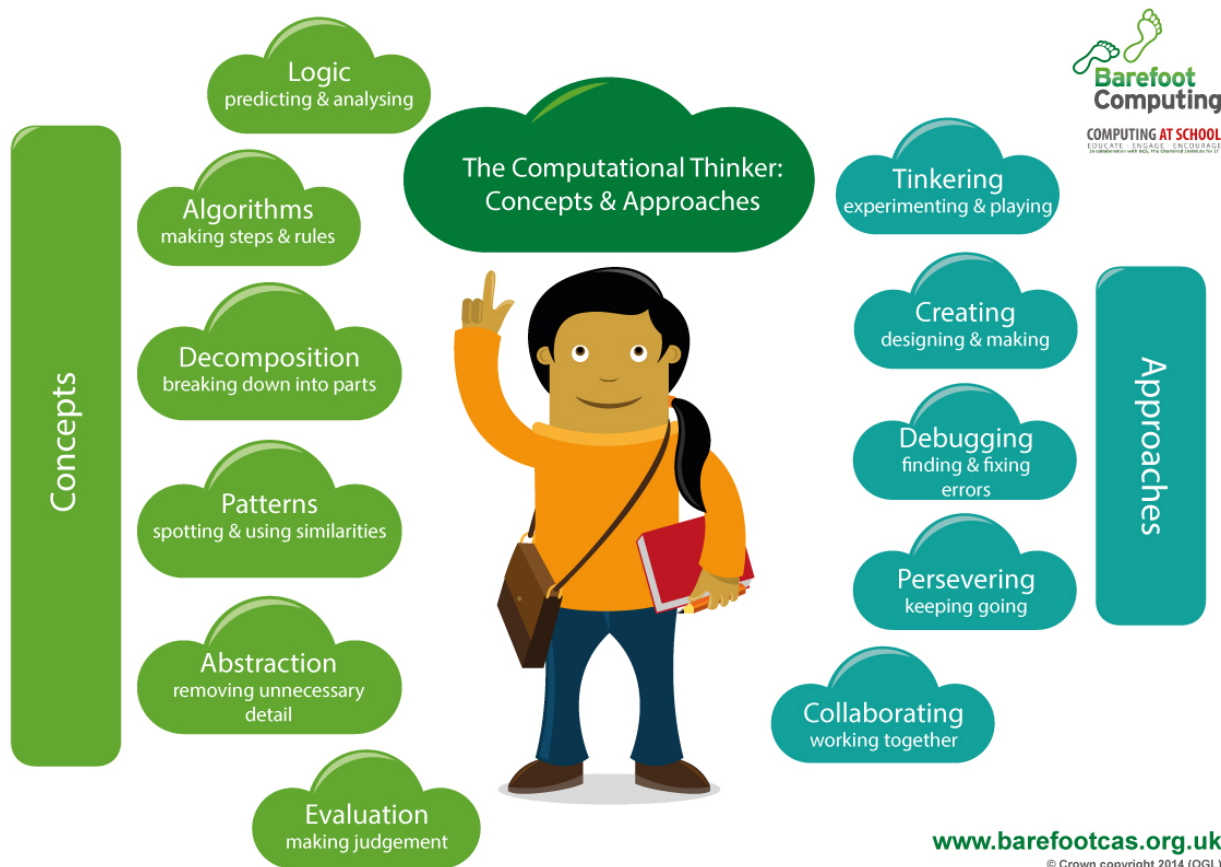


Figura 3.4. Marco-modelo de pensamiento computacional en el aula propuesto desde ‘CAS Barefoot’

Conceptos:

- **Lógica (‘Logic’)**: el razonamiento lógico nos ayuda a explicar por qué algo sucede. Si configuramos dos ordenadores de la misma manera, les damos las mismas instrucciones (el ‘programa’) y el mismo ‘input’, podemos entonces garantizar que llegaremos al mismo ‘output’. Los ordenadores no hacen las cosas según les apetezca, ni trabajan diferente en función de cómo se sientan ese día. Esto significa que son previsibles. Por esta razón, podemos utilizar el razonamiento lógico para determinar exactamente lo que un programa informático hará.

- **Algoritmos** (*'Algorithms'*): un algoritmo es una secuencia de instrucciones o conjunto de reglas para llevar algo a cabo. Podemos pensar, por ejemplo, en el camino más rápido para ir del colegio a casa (p.e. “*gira a la izquierda, sigue recto 5 kilómetros, gira a la derecha...*”) Podemos concebir esto como un algoritmo (una secuencia de instrucciones que nos lleva al destino seleccionado). Los ‘algoritmos’ están escritos generalmente para ser entendidos por humanos (p.e. una receta de cocina, Figura 3.5); en este sentido se diferencian de los ‘programas’, escritos para ser ejecutados por un ordenador.

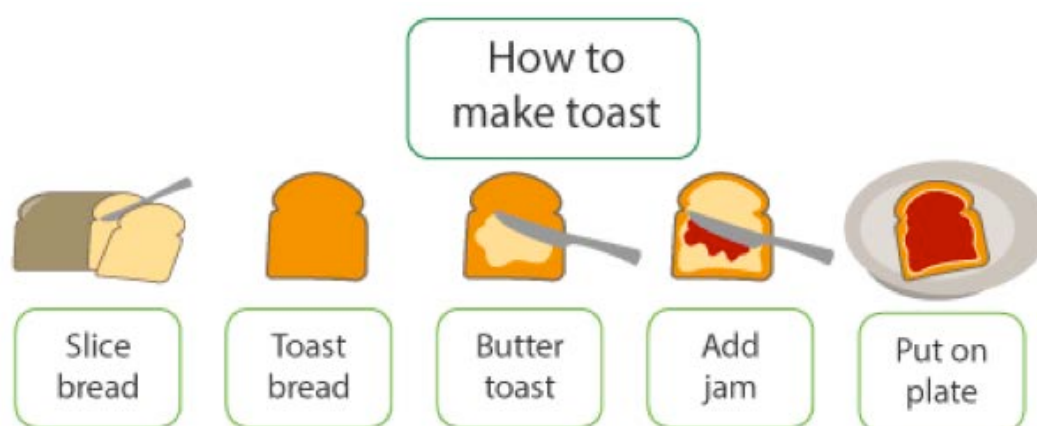


Figura 3.5. Ejemplo de algoritmo: secuencia de instrucciones para preparar una tostada²⁰⁷ (CAS, 2015, en línea)

- **Descomposición** (*'Decomposition'*): el proceso de fraccionar un problema en sus partes constitutivas, más pequeñas y manejables, se conoce como descomposición. La descomposición nos ayuda a resolver problemas complejos y a gestionar proyectos grandes. Esta aproximación tiene múltiples ventajas: hace que el proceso sea manejable y abordable (un problema grande es desalentador, pero un conjunto de problemas pequeños y relacionados entre sí es más llevadero). También implica que la tarea puede ser llevada a cabo por un equipo trabajando juntos y coordinados, cada uno aportando su experiencia y habilidades en la parte del problema adecuada.
- **Patrones** (*'Patterns'*): los patrones están por todas partes, por ejemplo, usamos patrones meteorológicos para predecir qué tiempo hará mañana; los niños pueden percibir patrones en cómo sus profesores reaccionan a sus conductas para decidir cómo comportarse la próxima ocasión. Al identificar patrones podemos hacer predicciones, crear reglas y resolver problemas más generales (Figura 3.6). En computación, el método de buscar una aproximación general a toda una clase de problemas análogos se llama ‘generalización’²⁰⁸

²⁰⁷ Pueden rastrearse en la Red varios vídeos grabados por profesores del Reino Unido, en los que se muestra cómo introducir lúdicamente a los estudiantes en el concepto de ‘algoritmo’ a través de dicho ejemplo con la tostada. Por ejemplo, se recomienda visualizar <https://youtu.be/leBEFaVHIIE>

²⁰⁸ Los científicos de la computación se esfuerzan por resolver problemas rápida y eficientemente, y para ello reutilizan métodos y soluciones creados previamente. Si detectan un patrón común a lo largo de un problema (o programa), tratarán de crear una solución común ‘tipo’ (o módulo) para ser reutilizada varias veces. Esto significa que sólo tendrán que diseñar este ‘módulo común’ una vez, en lugar de hacerlo en muchas ocasiones. Los módulos comunes de código se denominan habitualmente funciones (*'functions'*) o procedimientos (*'procedures'*); la mayoría de los lenguajes de programación tienen librerías compartidas de estas funciones (CAS, 2015, en línea)



Figura 3.6. La detección del patrón revela la regla ortográfica (ejemplo tomado de 'CAS Barefoot')

- **Abstracción ('Abstraction')**: la abstracción se sitúa en el corazón del pensamiento computacional. El proceso de abstraer, decidir a qué detalles necesitamos prestar atención y cuáles ignorar, atraviesa (está en el núcleo) del pensamiento computacional. La abstracción tiene que ver con simplificar las cosas; identificando qué es importante sin preocuparnos demasiado por lo anecdótico o irrelevante. La abstracción nos permite manejar la complejidad. Usamos abstracciones para gestionar la complejidad de la vida en las aulas. Por ejemplo, un horario escolar es una abstracción de lo que ocurre en una típica semana de curso: recoge la información clave (qué asignatura es impartida en cada aula, cada día, a cada hora, y por cuál profesor), dejando fuera múltiples capas de complejidad (objetivos de la asignatura, actividades y materiales, etc...)
- **Evaluación sistemática ('Evaluation')**: la evaluación tiene que ver con hacer juicios, de una manera objetiva y sistemática cuando sea posible. Evaluar es algo que hacemos cada día: hacemos juicios sobre qué hacer y qué pensar basados en una variedad de factores y criterios. Por ejemplo, al considerar la compra de un nuevo dispositivo digital para el aula, podría haber una serie de criterios a ser considerados, por ejemplo: el sistema operativo, la portabilidad, memoria, tamaño de pantalla, facilidad de uso, precio... En Ciencias de la Computación, la evaluación es sistemática y rigurosa; tiene que ver con juzgar la calidad, la efectividad y la eficiencia de las soluciones, sistemas, productos y procesos. La evaluación comprueba que las soluciones aportadas hacen el trabajo para el que fueron diseñadas.

Aproximaciones:

- **Experimentación ('Tinkering')**: El 'tinkering'²⁰⁹ significa probar a hacer (activamente) las cosas. Para los niños pequeños es la forma natural de aprender sobre algo: el juego espontáneo, la exploración y la experimentación. Para los estudiantes más mayores y adultos, es una exploración intencionada y un hacer basado en el 'ensayo, error y mejora'.

²⁰⁹ 'Tinker' es un término de jerga inglesa que tiene significados como 'reparar', 'hacer bricolaje', 'remendar'... y que se ha extendido al campo semántico de la codigoalfabetización y el pensamiento computacional al hacer hincapié en los procesos activos de 'ensayo y error' típicos en el aprendizaje de estas áreas; dando, por ejemplo, nombre a una de las plataformas-lenguajes de aprendizaje de código para niños más conocidas: <https://www.tynker.com/>

Habitualmente hacemos *'tinkering'* cuando nos encontramos con algo nuevo y queremos descubrir cómo funciona; por ejemplo, cuando adquirimos un teléfono móvil nuevo, podríamos probar todas sus características y funcionalidades, para posteriormente utilizar intencionadamente aquellas que nos sirvan. Tener la libertad de explorar a través de retos lúdicos en un ambiente no punitivo genera confianza y una actitud proactiva. Problemas abiertos animan la creatividad y la diversidad de ideas. Así, el *'tinkering'* (como aproximación al desarrollo del pensamiento computacional en el aula) debería ser divertido, libre, creativo y lleno de preguntas, retos y sorpresas.

- **Creación (*'Creating'*):** crear tiene que ver con planificar, hacer y evaluar cosas (p.e. animaciones, juegos o robots). Programar es un proceso creativo. El trabajo creativo implica tanto originalidad como la generación de un producto valioso: típicamente algo que es útil o que encaja con el propósito previsto. A veces lo que creamos son productos que dan respuesta a necesidades particulares; a veces lo que creamos parte de iniciativas para dar rienda suelta a nuestra expresión. Los productos del pensamiento computacional pueden ser tanto puramente digitales (a través de la creación de software), como artefactos físicos y tangibles (a través de la extensión hacia el hardware).
- **Depuración (*'Debugging'*):** los errores en un algoritmo, programa o código se denominan *'bugs'*²¹⁰, y el proceso de encontrarlos y arreglarlos se denomina *'debugging'* (puede traducirse al español como 'depuración'). El *'debugging'* habitualmente lleva mucho más tiempo que escribir originalmente el código. En la vida cotidiana, depuramos todo el tiempo: sencillamente es darse cuenta de los errores y solucionarlos. Por ejemplo, podemos comprobar una frase que hemos escrito en un correo electrónico para ver si tiene sentido y, en caso contrario, arreglarla-depurarla. El proceso de depurar un programa para que funcione bien puede ocasionar frustración y bullicio en el aula; para afrontarlo, una manera adecuada es proporcionar a los estudiantes un buen conjunto de estrategias de depuración que puedan usar para detectar-arreglar cualquier *'bug'* de programación. Algunos *'bugs'* son errores lógicos, otros son errores sintácticos. Los errores lógicos son como escribir una historia en la cual la trama no tiene sentido; los errores sintácticos son como escribir una historia con fallos gramaticales, ortográficos o de puntuación.
- **Perseverancia (*'Persevering'*)²¹¹:** la programación informática es una tarea dura. Esto es parte de su encanto: escribir código elegante y eficaz es un reto intelectual que requiere, no sólo una comprensión algorítmica del problema a resolver y codificar (y un conocimiento del lenguaje de programación utilizado para ello), sino también la disposición a perseverar sobre algo que es a menudo difícil y frustrante. Algunos ven en la experiencia de jugar a ciertos videojuegos desafiantes algo similar al *'coding'*: al jugar a un videojuego hay un continuo ciclo retroalimentación entre causas y efectos de la conducta del jugador, similar a cuando un programador codifica y depura. Ambos, *'coders'* y *'gamers'* experimentan una sensación de

²¹⁰ Literalmente *'bug'* en inglés significa 'bicho'; pero se ha extendido internacionalmente para denominar a los errores informáticos.

²¹¹ En este punto, la comunidad *'CAS Barefoot'* utiliza recurrentemente la siguiente cita atribuida a Einstein: *"It's not that I'm so smart, it's just that I stay with problems longer"* ("No es que yo sea tan listo, es simplemente que trabajo sobre los problemas durante más tiempo")

flujo en la cual permanecen absorbidos y focalizados en esa única tarea; lo que probablemente les ayuda a motivarse y perseverar hasta acabar el juego o resolver el problema.

- **Colaboración** (*'Collaborating'*): colaborar significa trabajar con otros para asegurar un mejor resultado. Es complicado pensar en cualquier trabajo o actividad de ocio en la sociedad digital actual que no involucre la colaboración. El trabajo colaborativo tiene además una larga tradición en nuestro sistema educativo (especialmente en la Educación Primaria), y la computación no debería suponer un cambio al respecto. Al programar, muchos ven el *'pair programming'* ('programación por parejas') con una forma particularmente eficaz de escribir código, con dos programadores compartiendo una pantalla y un teclado (Figura 3.7), trabajando juntos para crear software. Habitualmente, uno en la pareja actúa como 'conductor' (*'driver'*, el que teclea y presta atención al detalle del código), mientras que el otro toma el papel de 'navegador' (*'navigator'*, el que tiene en mente una visión amplia del problema, el 'paisaje completo'). Los dos miembros de la pareja intercambian regularmente sus roles, de manera que ambos tengan contacto con el código detallado y la visión de conjunto.



Figura 3.7. Ejemplo de *'pair programming'* (tomado de *'CAS Barefoot'*)

b) Un marco desde Estados Unidos: el modelo MIT-Harvard

Desde los Estados Unidos, los prestigiosos investigadores Karen Brennan (Universidad de Harvard) y Mitch Resnick (MIT) han formulado un modelo alternativo del pensamiento computacional (Brennan & Resnick, 2012)²¹². A lo largo de 5 años, han ido desarrollando su *'computational thinking framework'*, a través de la observación directa, entrevistas y análisis de proyectos de niños y jóvenes trabajando como diseñadores digitales interactivos (*'interactive media designers'*)

El contexto para la investigación y formulación de su modelo es Scratch²¹³: una plataforma-lenguaje visual de programación 'por bloques' que permite a los niños y jóvenes crear sus propias historias interactivas, animaciones, juegos y simulaciones; y posteriormente compartir dichas creaciones a través de una comunidad online con otros jóvenes programadores alrededor del mundo (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010; Resnick *et al.*, 2009). En la medida que Scratch se ha

²¹² Una síntesis web del modelo puede verse en <http://scratched.gse.harvard.edu/ct/index.html>

²¹³ <https://scratch.mit.edu/>

convertido en el lenguaje de programación para niños más usado en el mundo, y el que mejores características tiene para desarrollar el pensamiento computacional (Grover & Pea, 2013a), el modelo que presentamos a continuación es admitido como una referencia de primer orden a la hora de enmarcar la integración del PC en el aula.

Este modelo, que llamaremos MIT-Harvard, se articula alrededor de 3 dimensiones clave: a) ‘conceptos computacionales’ (*computational concepts*), es decir, los conceptos que los niños emplean cuando programan; b) ‘prácticas computacionales’ (*computational practices*), esto es, las prácticas que los niños emplean mientras programan; c) ‘perspectivas computacionales’ (*computational perspectives*), es decir, las perspectivas que los niños se forman sobre el mundo que les rodea y sobre sí mismos como consecuencia de la programación. Se diagrama en la Figura 3.8.

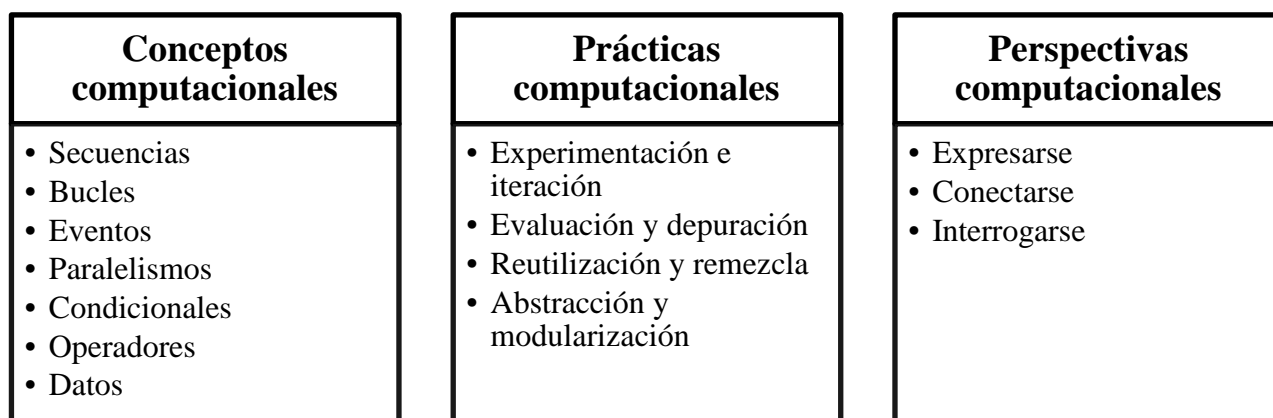


Figura 3.8. Diagrama resumen del Modelo MIT-Harvard de pensamiento computacional (Brennan & Resnick, 2012)

Veámoslo con algo más detalle:

- **Conceptos computacionales** (*computational thinking concepts*) [**¿Qué aprenden?**]: según los niños aprenden a programar, diseñando objetos digitales interactivos con Scratch, toman contacto con un conjunto de conceptos computacionales (alineados con los tipos de bloques de programación que ofrece el lenguaje Scratch), que son comunes a la mayoría de los lenguajes de programación. Brennan y Resnick (2012) han identificado 7 conceptos de alto uso y utilidad en un amplio rango de proyectos Scratch, y que pueden ser transferidos a otros entornos, tanto de programación como de resolución de problemas en general:
 - **Secuencias** (*sequences*): un concepto clave en programación es identificar y expresar una actividad o tarea como una serie de pasos individuales (discretos) y ordenados, que puedan ser ejecutados por un ordenador. Como una receta de cocina, una secuencia de instrucciones de programación específica el comportamiento o acción que debe producirse.
 - **Bucles** (*loops*): los bucles permiten ejecutar una misma secuencia de instrucciones en múltiples ocasiones; convirtiendo a los programas en expresiones más sucintas y elegantes. Así, en la Figura 3.9 vemos a la izquierda una secuencia de 9 instrucciones en lenguaje Scratch; a la derecha, la misma secuencia escrita de manera más sucinta al utilizar un bucle dentro del programa.



Figura 3.9. Una secuencia con instrucciones repetidas (a la izquierda), expresada a través de un bucle (a la derecha)

- **Eventos** ('events'): se refiere a 'cuando algo pasa, entonces causa (dispara) que otra cosa ocurra'. Los eventos son componentes esenciales de los objetos programados interactivos. Por ejemplo, un botón de 'start' que está programado para que, cuando sea pulsado, dispare el inicio de un vídeo musical; o un evento consistente en que, cuando una tecla determinada es presionada, un personaje se mueva en una dirección dada.
- **Paralelismos** ('parallelism'): son varias secuencias de instrucciones que se ejecutan al mismo tiempo, simultáneamente ('en paralelo'). La mayoría de lenguajes de programación modernos soportan paralelismos. En Figura 3.10 vemos un programa de Scratch, en el que el objeto (un gato) ha sido programado para ejecutar 3 secuencias de actividad en paralelo en respuesta al evento 'al presionar la bandera verde'

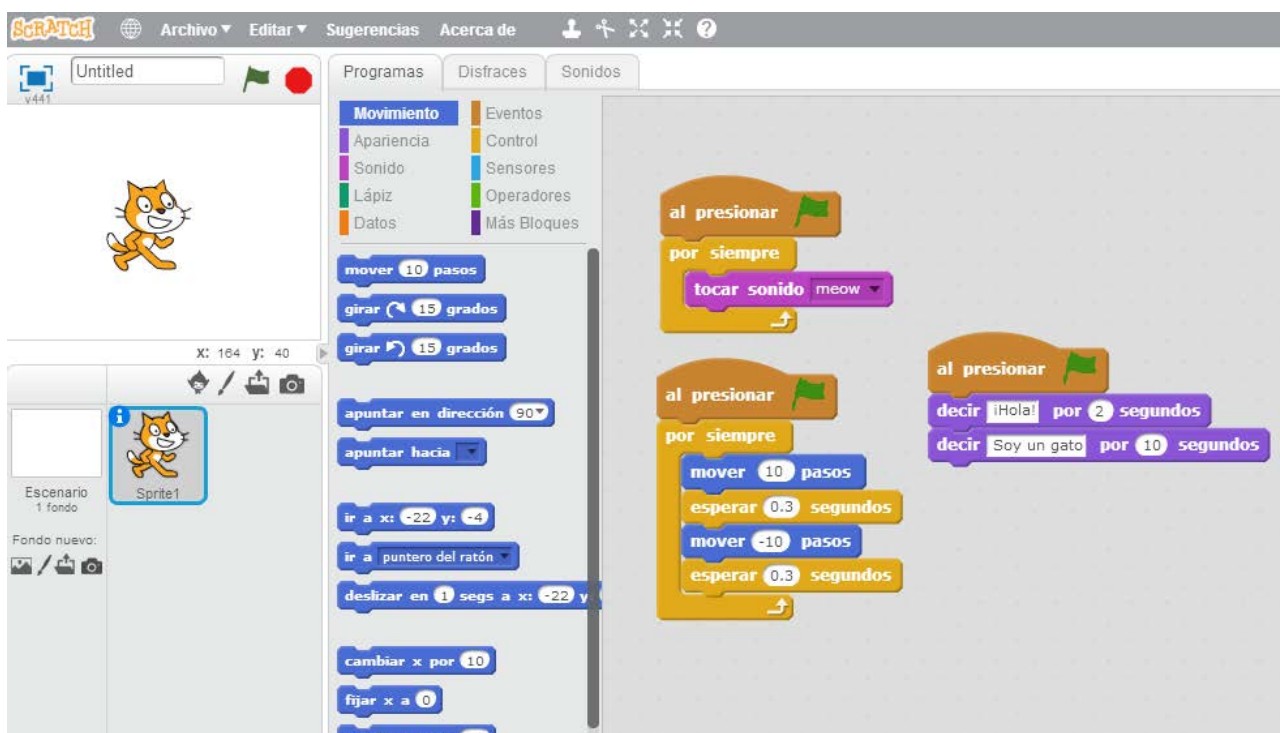


Figura 3.10 Ejemplo de 3 secuencias ejecutadas en paralelo en respuesta al mismo evento ('al presionar bandera verde')

- **Condicionales** (*'conditionals'*): otro concepto computacional clave en programación son los condicionales, es decir, la capacidad de tomar decisiones basadas en ciertos estados o situaciones. Ello permite a un programa expresar distintos resultados en función de las condiciones dadas, dotándolo de versatilidad.
- **Operadores** (*'operators'*): los operadores permiten al programador incluir expresiones lógicas, matemáticas y de cadena, en sus programas. Scratch soporta una amplia variedad de operadores (Figura 3.11)
- **Datos** (*'data'*): incluye el almacenamiento, recuperación y actualización de valores en un programa. Scratch, por ejemplo, ofrece dos tipos de contenedores de datos: 'variables' (que pueden almacenar, recuperar y actualizar un solo número; algo que se utiliza, por ejemplo, para hacer un marcador de tantos en un videojuego que va actualizando la puntuación del jugador) y 'listas' (que pueden almacenar, recuperar y actualizar conjuntos de números)

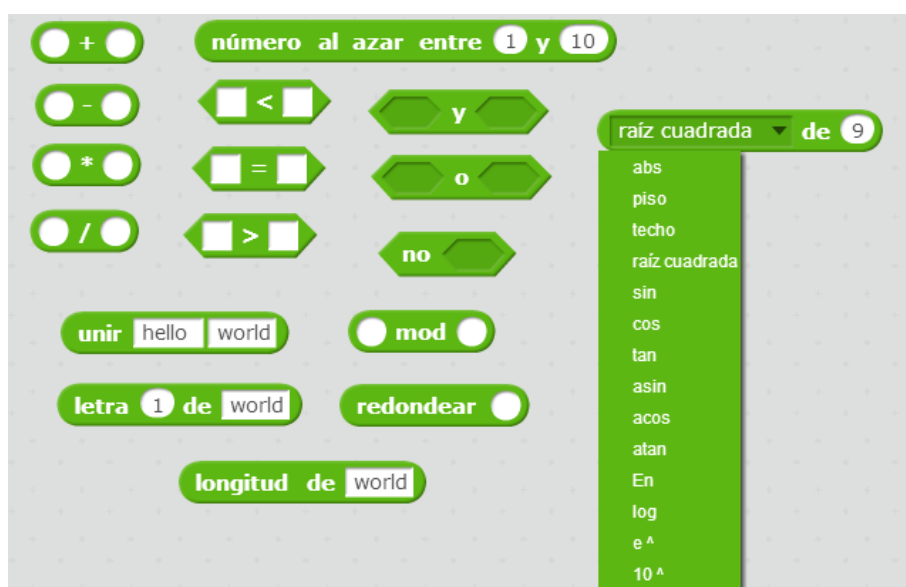


Figura 3.11. Operadores disponibles en Scratch

- **Prácticas computacionales** (*'computational thinking practices'*) [**¿Cómo lo aprenden?**]: si los conceptos computacionales se enfocan al contenido de aprendizaje (lo que el niño aprende al programar), las prácticas computacionales se focalizan en cómo lo aprenden. Es decir, las prácticas computacionales se refieren a qué tipo de procesos y prácticas ponen en marcha los niños cuando construyen sus programas. Brennan y Resnick (2012) identifican 4 tipos de prácticas:
 - **Experimentación e iteración** (*'experimenting and iterating'*): diseñar-programar un proyecto no es un proceso absolutamente 'limpio' y lineal, en el cual primero se concibe una idea, después se desarrolla un plan para su diseño, y finalmente se implementa dicho diseño en forma de código. Más bien, es un proceso adaptativo e iterativo, en el cual el plan de diseño va transformándose según se van encontrando las soluciones en forma de código, a través de sucesivas aproximaciones por pequeños

pasos. Según informan los propios *'scratchers'*, su proceso de construcción del programa se asemeja a ciclos iterativos de 'imaginar y construir': *'developing a little bit, then trying it out, then developing more'* ('desarrollar un poco, entonces probar, y entonces desarrollar un poco más')

- **Evaluación y depuración** (*'testing and debugging'*): los programas raramente funcionan a la primera y en la forma que fueron imaginados. Para los diseñadores-programadores es crítico desarrollar estrategias para anticipar y afrontar los errores que aparecen en sus programas. Según informan los propios *'scratchers'*, durante el proceso de construcción de sus programas ponen en marcha espontáneamente procedimientos de evaluación y depuración de los mismos a través de sucesivas pruebas de 'ensayo y error', y búsqueda de apoyo en otros pares de la comunidad con mayor nivel de pericia.
- **Reutilización y remezcla** (*'reusing and remixing'*): es decir, escribir un programa a partir de otros programas preexistentes. Construir a partir del trabajo de otros es una práctica habitual con larga tradición en el mundo de la programación; que ha sido amplificada a partir de que las tecnologías digitales se han conectado en red a través de Internet, proporcionando acceso a una enorme cantidad de trabajo de otros programadores listo para ser usado y remezclado. Uno de los objetivos de la comunidad en línea de Scratch es precisamente ayudar a los jóvenes diseñadores-programadores a reusar y remezclar, orientándoles para encontrar ideas y código sobre el que puedan construir nuevos programas; capacitándoles para crear objetos digitales mucho más complejos de lo que hubieran podido abordar por sí mismos²¹⁴. La reutilización y la remezcla promueven además la habilidad de 'lectura crítica' del código escrito por otros, y abren preguntas importantes acerca de la propiedad y la autoría de los programas.
- **Abstracción y modularización** (*'abstracting and modularizing'*): la abstracción y la modularización, que podemos caracterizar como el proceso de construir algo de gran tamaño a partir de ir agregando conjuntos de elementos más pequeños, es una práctica muy importante para la solución de problemas en general, y para el diseño-programación en particular. En Scratch es muy habitual desarrollar proyectos muy complejos a partir de agregaciones de conjuntos de código más sencillos.
- **Perspectivas computacionales** (*'computational thinking perspectives'*) [**¿Para qué lo aprenden?**]: a lo largo de sus entrevistas con *'scratchers'*, Brennan y Resnick (2012) recogen frecuentes testimonios de los jóvenes diseñadores-programadores relatando cómo, a partir de su aprendizaje con Scratch, ha evolucionado su comprensión de sí mismos, de la relación con los otros, y del mundo digital-tecnológico que les rodea. Así, los autores añaden una tercera dimensión a su modelo, las *perspectivas computacionales*, que se centran en los 'cambios de perspectiva' que experimentan los niños a partir de su aprendizaje con Scratch. Distinguen entre:

²¹⁴ Por ejemplo, en <https://scratch.mit.edu/projects/88486311/remixtree/> se accede al 'árbol de reinversiones-remezclas' del proyecto Scratch llamado *'Tour the world'*

- **Expresarse** (*‘expressing’*): estamos rodeados de objetos digitales interactivos, pero la mayoría de nuestras experiencias con dichos objetos son como ‘consumidores’. Un ‘pensador computacional’ concibe la tecnología como algo más que objetos de consumo; la computación es un medio que puede ser utilizado para la creación, el diseño y la autoexpresión.
- **Conectarse** (*‘connecting’*): los jóvenes ‘*scratchers*’ describen el poder que supone tener acceso, a través de la comunidad ‘*on-line*’ de Scratch, a otros jóvenes programadores, a sus proyectos e ideas: *‘I can do different things when I have access to others’* (‘Puedo hacer cosas distintas cuando tengo acceso a los otros’). Este ‘tener acceso a otros’ tendría una doble dimensión:
 - El valor de crear *con* otros (reusando y remezclando sus creaciones; o programando colaborativamente con otros; o consultando a otros cuando no se sabe cómo avanzar)
 - Y el valor de crear *para* otros, al tener acceso a audiencias reales que pueden disfrutar de tus diseños-programas (entreteniéndoles con tus creaciones y, a la vez, equipándoles con nuevos conjuntos de código que pueden reutilizar)
- **Interrogarse** (*‘questioning’*): se observa que, a través de la programación, los jóvenes ‘*scratchers*’ se sienten empoderados a hacerse preguntas acerca del mundo digital que les rodea. Al aprender a escribir código, aprenden también a no dar por sentado los objetos digitales que consumen a diario, y a preguntarse por su naturaleza y por los posibles cambios que se podrían introducir en los mismos.

3.1.5. Una definición propia del dominio muestral del constructo

En este punto, estamos en condiciones de ofrecer una definición propia de ‘pensamiento computacional’. Es una definición operativa u operacional ‘de trabajo’, muy focalizada en los ‘conceptos computacionales’ (*‘computational concepts’*), que nace con el ánimo de delimitar claramente el dominio muestral del constructo; y así posibilitar su posterior medida y evaluación desde una perspectiva psicométrica clásica.

Esta definición nos servirá de partida para el diseño y validación del ‘Test de Pensamiento Computacional’ (TPC), del que se da cuenta en el Capítulo 6 de esta tesis.

Así, para nosotros:

“El pensamiento computacional es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica inherente a los lenguajes informáticos de programación: secuencias o direcciones básicas, bucles, condicionales, funciones, y variables”

Hemos venido utilizando dicha definición en algunos trabajos previos a esta tesis, que han contado con alta receptividad por la comunidad científica (Román-González, 2015a, 2015b; Román-González, Pérez-González, & Jiménez-Fernández, 2015)

3.2. Desarrollo

En este epígrafe nos centramos en cómo desarrollar el pensamiento computacional (PC), especialmente en contextos educativos y durante el periodo de escolarización obligatoria. Veremos, en primer lugar, una terna de modelos de implantación del PC en distintos sistemas educativos; y, posteriormente, una revisión de experiencias y recursos para el desarrollo del PC en entornos escolares.

3.2.1. Modelos de desarrollo del pensamiento computacional

Presentamos una terna de modelos de desarrollo del pensamiento computacional, cada uno de los cuales parte de un enfoque bien distinto: en primer lugar, y nacido en el contexto escolar estadounidense, en el que las Ciencias de la Computación no tienen entidad de asignatura obligatoria y continuada en su currículum, detallamos el ‘*Modelo transversal de desarrollo del PC de Barr & Stephenson*’ (Barr & Stephenson, 2011). En segundo lugar, también desde los Estados Unidos, presentamos los estándares curriculares propuestos por la ‘*Computer Science Teachers Association*’ (CSTA, 2011) para el desarrollo del pensamiento computacional a lo largo de las distintas etapas de su sistema educativo. Finalmente, proveniente del contexto escolar británico en el que las Ciencias de la Computación sí han sido incluidas como asignatura obligatoria en su currículum a lo largo de todas las etapas obligatorias (UK Department of Education, 2013), desde los 5 a los 16 años, detallamos el ‘*Modelo longitudinal de desarrollo del PC: los itinerarios CAS*’ (Dorling, 2015).

a) Modelo transversal de desarrollo del PC de Barr & Stephenson

Según estos autores estadounidenses, los estudiantes de hoy en día viven, y vivirán, en sociedades fuertemente influenciadas por la computación; y muchos de ellos trabajarán en campos que están implicando progresivamente procesos computacionales en mayor o menor medida. En ese sentido, afirman que los estudiantes “*deben comenzar a adquirir habilidades de solución algorítmica de problemas, y a trabajar con métodos y herramientas computacionales desde las etapas de escolarización obligatoria*” (Barr & Stephenson, 2011, p. 49)

El objetivo de sus trabajos de investigación es articular un conjunto de conceptos computacionales clave que puedan ser impartidos y aplicados de manera transversal en las distintas disciplinas del currículum, “*destacando las habilidades de solución algorítmica de problemas y las aplicaciones de la computación a través de las distintas disciplinas y asignaturas (...) ayudando a integrar los métodos y herramientas de la computación en las diversas áreas de aprendizaje*” (Barr & Stephenson, 2011, p. 49) que marca el currículum obligatorio de los Estados Unidos; en el que las Ciencias de la Computación no tienen entidad de asignatura propia. Para abordarlo, parten de la siguiente definición de partida sobre el pensamiento computacional:

“El pensamiento computacional es un enfoque o aproximación particular a la solución de problemas, de manera que éstos puedan ser formulados y resueltos con la ayuda de un ordenador. Con su desarrollo, los estudiantes se convierten de meros usuarios de herramientas a constructores de las mismas. A través del desarrollo del pensamiento computacional, los estudiantes adquieren un conjunto de conceptos tales y como la abstracción, la recursividad, y la iteración, que utilizan para procesar y analizar datos, y

para crear artefactos tanto físicos como digitales. En suma, el pensamiento computacional es una metodología de solución de problemas que puede ser automatizada, transferida y aplicada a lo largo de las distintas materias y asignaturas” (Barr & Stephenson, 2011, p. 51)

Partiendo de esta definición general, y alineados igualmente con la definición operativa sobre pensamiento computacional formulada originalmente en el año 2011 por la ‘*Computer Science Teachers Association*’ de los Estados Unidos (CSTA & ISTE, 2015, en línea) que vimos algo más arriba; Barr & Stephenson (2011) enuncian un modelo de integración transversal de los distintos conceptos y capacidades que componen el PC a través de las distintas áreas del currículo. Dicho modelo se resume en la siguiente Tabla 3.1.

Tabla 3.1 Modelo transversal de desarrollo del PC de Barr & Stephenson (2011)

Concepto/Capacidad del Pensamiento Computacional	Área Curricular				
	Informática y Tecnología	Matemáticas	Ciencias Naturales	Ciencias Sociales	Lengua y Literatura
Recogida de datos	Encontrar una fuente de datos para un problema o conjunto de problemas	Encontrar una fuente de datos para un tipo de problemas, por ejemplo, el ‘lanzamiento de monedas’ o ‘el lanzamiento de dados’	Recoger datos de un experimento	Analizar/estudiar las estadísticas de una batalla, o los datos demográficos	Realizar análisis lingüísticos de frases
Análisis de datos	Escribir un programa para hacer cálculos estadísticos básicos sobre un conjunto de datos	Contar las ocurrencias de aparición de caras en el lanzamiento de monedas y dados, y analizar los resultados	Analizar los datos provenientes de un experimento	Identificar tendencias en los datos estadísticos de un fenómeno social	Identificar patrones para distintos tipos de frases y estructuras sintácticas
Representación de datos	Utilizar estructuras de datos como matrices, listas vinculadas, apilamientos, gráficos de dispersión, etc...	Utilizar histogramas, gráficos de barras y de sectores, para representar los datos. Utilizar listas y apilamientos para alojar datos	Sintetizar/resumir los datos de un experimento	Sintetizar/resumir y representar tendencias de fenómenos sociales	Representar los patrones detectados en distintos tipos de frases y estructuras sintácticas
Descomposición de problemas	Definir objetos y métodos; definir funciones	Aplicar el orden de las operaciones en una expresión	Hacer una taxonomía de especies		Escribir un esquema
Abstracción	Utilizar procedimientos para encapsular un conjunto de órdenes que se repiten frecuentemente en una función; utilizar condicionales, bucles, recursividad	Utilizar variables en álgebra; identificar los hechos esenciales en un problema expresado verbalmente; estudiar funciones de álgebra y compararlas con funciones equivalentes en programación	Construir un modelo de alguna entidad o fenómeno físico	Sintetizar/resumir hechos; deducir conclusiones de los hechos	Utilizar símiles y metáforas; escribir una historia con ramas y bifurcaciones

<i>Concepto/Capacidad del Pensamiento Computacional</i>	<i>Área Curricular</i>				
	<i>Informática y Tecnología</i>	<i>Matemáticas</i>	<i>Ciencias Naturales</i>	<i>Ciencias Sociales</i>	<i>Lengua y Literatura</i>
<i>Algoritmos & Procedimientos</i>	Estudiar algoritmos clásicos; diseñar e implementar un algoritmo para resolver un determinado tipo de problema	Realizar factorizaciones para resolver divisiones grandes; acarrear en las operaciones de adición y sustracción	Llevar a cabo un procedimiento experimental		Escribir un conjunto de instrucciones
<i>Automatización</i>		Utilizar herramientas como: geometer sketch pad ²¹⁵ , star logo ²¹⁶	Utilizar ‘probeware’ ²¹⁷	Utilizar críticamente Excel	Utilizar críticamente un corrector ortográfico
<i>Paralelización</i>	Dividir los datos y disponerlos en diferentes hilos para que puedan ser procesados en paralelo	Resolver sistemas de ecuaciones lineales; realizar multiplicaciones matriciales	Ejecutar simultáneamente varias réplicas del mismo experimento con diferentes parámetros de configuración		
<i>Simulación</i>	Animación algorítmica; Uso de ‘parámetros de barrido’	Representar dinámicamente una función en un plano cartesiano, modificando los valores de las variables	Simular el movimiento del Sistema Solar	Jugar críticamente a ‘Age of Empires’ ²¹⁸	Hacer una representación o recreación de una historia

En orden a promover una cultura de aula que potencie el pensamiento computacional, Barr & Stephenson (2011) identifican una serie de estrategias generales que serían beneficiosas para lograr dicho objetivo:

- Incremento del uso, tanto por parte de estudiantes como de profesores, de vocabulario computacional apropiado (p.e. ‘algoritmo’, ‘iteración’, ‘paralelización’, etc...); que les permita describir correctamente problemas y soluciones en términos computacionales.
- Aceptación, tanto por parte de estudiantes como de profesores, de los intentos fallidos de solución de un problema; reconociendo que un fallo temprano nos sitúa a menudo en el camino correcto hacia una solución exitosa.
- Trabajo en equipo de los estudiantes, con uso explícito de:
 - ‘Descomposición’ (*decomposition*): fraccionar los problemas en partes más pequeñas, lo que hace más sencilla su resolución.

²¹⁵ Geometer Sketch Pad es un software para la visualización dinámica de geometrías matemáticas. Disponible en <http://www.dynamicgeometry.com/>

²¹⁶ http://education.mit.edu/portfolio_page/starlogo-tng/

²¹⁷ ‘Probeware’ es un término que designa herramientas de software que recogen datos en tiempo real de un entorno experimental; representando y analizando dichos datos igualmente en tiempo real.

²¹⁸ <http://www.ageofempires.com/>

- ‘Abstracción’ (*‘abstraction’*): simplificar de lo concreto a lo general a medida que las soluciones están siendo desarrolladas.
- ‘Negociación’ (*‘negotiation’*): los distintos sujetos y grupos, trabajando juntos y en equipo, deben unir las diferentes partes de la solución en un todo que funcione.
- ‘Construcción de consensos’ (*‘consensus building’*): trabajar construyendo solidariamente alrededor de una idea y una solución común.

b) Los estándares curriculares de la CSTA

También desde los Estados Unidos, y muy vinculados al modelo anterior, presentamos los estándares curriculares para las Ciencias de la Computación propuestos por la ‘*Computer Science Teachers Association*’ (CSTA, 2011), que cubren todas las etapas educativas pre-universitarias del sistema educativo estadounidense. Tal y como venimos comentando, las ‘Ciencias de la Computación’ (*‘Computer Science’*) no existe como asignatura obligatoria y continuada a lo largo del currículum educativo de los Estados Unidos; por lo tanto, estos estándares se conciben para ser abordados de manera transversal a través de las distintas materias o asignaturas que el estudiante va cursando en su trayectoria educativa (que sí puede incluir las ‘Ciencias de la Computación’, a modo de asignatura optativa en algunos cursos concretos).

Los estándares están organizados en 5 familias o hebras: ‘Pensamiento Computacional’ (*‘Computational Thinking’*); ‘Prácticas de Computación y Programación’ (*‘Computing Practice and Programming’*); ‘Colaboración’ (*‘Collaboration’*); ‘Dispositivos Informáticos y de Comunicación’ (*‘Computer and Communication Devices’*); ‘Impactos Éticos, Globales y sobre la Comunidad’ (*‘Community, Global and Ethical Impacts’*). Se entiende que las 5 familias se complementan e interpenetran; y se representan en la Figura 3.12:

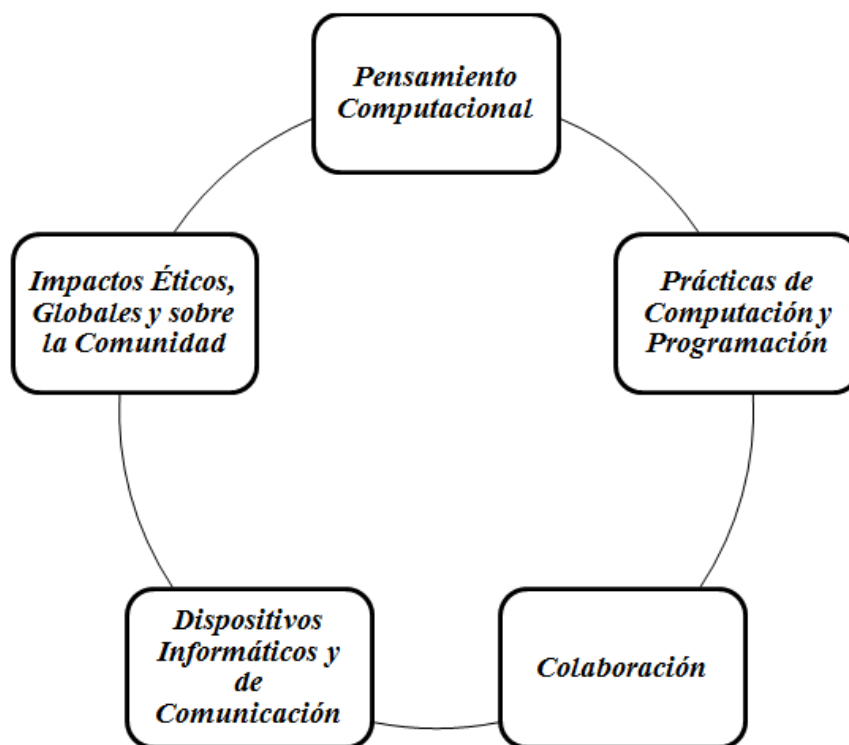


Figura 3.12. Familias de estándares curriculares para las ‘Ciencias de la Computación’ de la CSTA (2011)

A su vez, las 5 familias de estándares se diferencian en función de 3 niveles escolares:

- ✓ ‘Nivel 1’ (*‘Level 1’*, incluye los Grados K1-K6, equivalentes a 1º Primaria-6º Primaria);
- ✓ ‘Nivel 2’ (*‘Level 2’*, incluye los Grados K6-K9, equivalentes a 6º Primaria-3º ESO);
- ✓ ‘Nivel 3’ (*‘Level 3’*, incluye los Grados K9-K12, equivalentes a 3º ESO-2º Bachillerato)

Centrándonos en las 2 primeras familias de estándares, es revelador comprobar cómo la CSTA distingue conceptualmente entre el ‘pensamiento computacional’ y la ‘programación informática’. Tal y como venimos repitiendo a lo largo de esta tesis, mientras la programación informática es una práctica particular de lectoescritura (lectoescritura de programas informáticos usando la sintaxis particular de un lenguaje de programación); el pensamiento computacional es un proceso cognitivo de resolución de problemas, que se apoya en los conceptos de la computación, y que sirve de base a las tareas de programación pero también a muchas otras, siendo transferible a gran cantidad de áreas. Dicho de otra manera, cuando programamos siempre estamos utilizando y desarrollando nuestro pensamiento computacional; pero podemos también utilizar y desarrollar nuestro pensamiento computacional en tareas que no incluyen la programación informática.

Así se desprende de las definiciones que, al comienzo de sus estándares curriculares, ofrece la CSTA al respecto de estas 2 familias: ‘Pensamiento Computacional’ (*‘Computational Thinking’*) y ‘Prácticas de Computación y Programación’ (*‘Computing Practice and Programming’*). Concretamente, al respecto del pensamiento computacional, se dice:

“Consideramos que el pensamiento computacional puede ser usado a través de todas las disciplinas para resolver problemas, diseñar sistemas, crear nuevo conocimiento, y mejorar nuestra comprensión sobre el poder y los límites de la computación en la era digital. El estudio y desarrollo del pensamiento computacional capacita a todos los estudiantes a conceptualizar mejor, analizar, y resolver problemas complejos mediante la selección y aplicación de las estrategias y herramientas adecuadas; tanto en un entorno digital como físico, virtual o real (...) así pues, el pensamiento computacional es una metodología de resolución de problemas que puede servir de puente entre las Ciencias de la Computación y el resto de las disciplinas, proporcionando un particular enfoque a la formulación y resolución de problemas, de manera que éstos puedan ser solucionados computacionalmente. Con su foco puesto en la abstracción, la automatización y el análisis, el pensamiento computacional es un elemento central de la disciplina más amplia de las Ciencias de la Computación, y por dicha razón se incluye como una de las familias de estándares a lo largo de todos los niveles de aprendizaje” (CSTA, 2011, p. 9-10)

En la siguiente Tabla 3.2 se presentan los estándares curriculares para ambas familias propuestas por la CSTA a lo largo de todos los niveles educativos contemplados. En conjunto, la sucesión de estándares expuesta puede considerarse un modelo de desarrollo del pensamiento computacional (y de las prácticas de programación informática) a lo largo de todas las etapas educativas pre-universitarias:

Tabla 3.2. Estándares curriculares propuestos por la CSTA (2011) para el ‘Pensamiento Computacional’ y las ‘Prácticas de Computación y Programación’

Familia de estándares		
	Pensamiento Computacional (<i>Computational Thinking – CT</i>)	Prácticas de Computación y Programación (<i>Computing Practices and Programming – CPP</i>)
Nivel 1 (<i>‘Level 1’ – LI</i>)	<i>Grados K1-K3 (LI:3.CT)</i> El estudiante será capaz de: <ol style="list-style-type: none"> 1. Utilizar recursos tecnológicos-digitales (p.e. puzzles, programas de pensamiento lógico) para resolver problemas adecuados a su edad. 2. Utilizar herramientas de escritura, cámaras digitales, y herramientas de dibujo para ilustrar pensamientos, ideas e historias de manera ‘paso por paso’ 3. Entender cómo organizar (clasificar) información en un orden útil (como, por ejemplo, ordenar a sus compañeros por fecha de nacimiento); sin necesidad de utilizar un ordenador. 4. Reconocer que el software es creado para controlar las operaciones que realiza el ordenador. 5. Demostrar como los 0s y 1s pueden ser usados para representar información (modo binario) 	<i>Grados K1-K3 (LI:3.CPP)</i> El estudiante será capaz de: <ol style="list-style-type: none"> 1. Utilizar recursos tecnológico-digitales para llevar a cabo investigaciones apropiadas a su edad. 2. Utilizar recursos digitales-multimedia adecuados a su etapa evolutiva (p.e. libros educativos y software educativo), para apoyar su aprendizaje a través del currículum. 3. Crear productos digitales-multimedia adecuados a su etapa evolutiva con el apoyo de sus profesores, familia y compañeros de aula. 4. Construir un conjunto de sentencias/afirmaciones que sirvan para acometer una tarea sencilla (p.e. una receta de cocina) 5. Identificar trabajos que utilizan la computación y la tecnología. 6. Reunir y organizar información utilizando herramientas de mapeo conceptual

Familia de estándares

Pensamiento Computacional
(*Computational Thinking – CT*)

Prácticas de Computación y Programación
(*Computing Practices and Programming – CPP*)

Nivel 1

Grados K3-K6 (L1:6.CT)

Grados K3-K6 (L1:6.CPP)

(*‘Level 1’ – L1*)

El estudiante será capaz de:

1. Comprender y utilizar los pasos básicos de la solución algorítmica de problemas (p.e. planteamiento y exploración del problema, examen de casos de la muestra, diseño, implementación y evaluación)
2. Desarrollar una comprensión sencilla de un algoritmo (p.e. un algoritmo de búsqueda, un algoritmo de clasificación, o una secuencia de eventos), a través de ejercicios sin ordenador.
3. Demostrar como una cadena de bits puede ser utilizada para representar información alfanumérica.
4. Describir cómo se puede utilizar una simulación para resolver un problema.
5. Hacer una lista de sub-problemas a considerar a la hora de afrontar un problema de magnitud mayor.
6. Comprender las conexiones entre las ciencias de la computación y otras áreas.

El estudiante será capaz de:

1. Utilizar recursos tecnológico-digitales (p.e. calculadoras, instrumentos de recogida de datos, dispositivos móviles, videos, software educativo, y herramientas web) para la resolución de problemas y el aprendizaje auto-dirigido.
2. Utilizar herramientas y periféricos que apoyen y promuevan la productividad personal, remedien los déficits de habilidades, y faciliten el aprendizaje.
3. Utilizar herramientas tecnológico-digitales (p.e. presentaciones multimedia, herramientas web, cámaras digitales, y escáneres) para realizar escritura individual y colaborativa, así como actividades de comunicación y publicación.
4. Reunir y manipular datos utilizando una amplia variedad de herramientas digitales.
5. Construir/escribir un programa como conjunto de instrucciones ‘paso por paso’ que debe ser llevado a cabo (p.e. cómo hacer un sándwich de mantequilla y mermelada)
6. Implementar soluciones a problemas utilizando lenguajes de programación visuales ‘por bloques’.
7. Utilizar dispositivos computacionales-digitales para acceder a información remota, y comunicarse con otros para apoyar el aprendizaje directo e independiente; y perseguir los intereses personales.
8. Navegar a través de páginas web utilizando hipervínculos, y llevar a cabo búsquedas simples utilizando motores de búsqueda.
9. Identificar un amplio rango de trabajos que requieren conocimientos de computación.
10. Reunir u manipular datos utilizando una amplia variedad de herramientas digitales.

Familia de estándares

Pensamiento Computacional
(*Computational Thinking – CT*)

Prácticas de Computación y Programación
(*Computing Practices and Programming – CPP*)

Nivel 2

Grados K6-K9 (L2.CT)

Grados K6-K9 (L2.CPP)

(*'Level 2' – L2*)

El estudiante será capaz de:

1. Utilizar los pasos básicos de la resolución algorítmica de problemas para diseñar soluciones eficaces a los mismos.
2. Describir el proceso de paralelización en la medida que se relaciona con la solución de problemas.
3. Definir un algoritmo como una secuencia de instrucciones que puede ser procesada por un ordenador.
4. Evaluar de qué manera diferentes algoritmos pueden ser usados para resolver el mismo problema.
5. Representar algoritmos de búsqueda y algoritmos de ordenación.
6. Describir y analizar una secuencia de instrucciones que está siendo seguida-ejecutada (p.e. describir el comportamiento de un personaje en un videojuego como objeto controlado por reglas y algoritmos)
7. Representar datos en una amplia variedad de formas, incluyendo texto, sonidos, imágenes y números.
8. Utilizar representaciones visuales para dar cuenta del estado de un problema o estructura de datos (p.e. gráficos, cuadros, diagramas de red, diagramas de flujo)
9. Interactuar con modelos y simulaciones relativas a contenidos específicos (p.e. ecosistemas, epidemias, dinámica molecular), para apoyar su aprendizaje e investigación.
10. Evaluar qué tipo de problemas pueden ser resueltos utilizando modelos y simulaciones.
11. Analizar el grado en el que un modelo computacional representa con exactitud el mundo real.
12. Utilizar la abstracción para descomponer un problema en sub-problemas.
13. Comprender la noción de jerarquía y abstracción en la computación, incluyendo lenguajes de alto nivel, traducción, conjuntos de instrucciones, y circuitos lógicos.
14. Examinar las conexiones entre elementos de las matemáticas y de las ciencias de la computación, incluyendo números binarios, lógicas, variables y funciones.
15. Proporcionar ejemplos de aplicaciones interdisciplinarias del pensamiento computacional.

El estudiante será capaz de:

1. Seleccionar las herramientas y recursos tecnológico-digitales apropiados para acometer una amplia variedad de tareas y solucionar problemas.
2. Utilizar una amplia variedad de herramientas y periféricos multimedia-digitales para apoyar la productividad personal y el aprendizaje a lo largo del currículo.
3. Diseñar, desarrollar, publicar y presentar productos (p.e. páginas web, aplicaciones móviles, animaciones) utilizando recursos tecnológico-digitales, que demuestren y comuniquen conceptos curriculares.
4. Demostrar la comprensión de algoritmos y sus aplicaciones prácticas.
5. Implementar soluciones a problemas utilizando un lenguaje informático de programación, incluyendo: bucles, sentencias condicionales, expresiones lógicas, variables y funciones.
6. Demostrar buenas prácticas en lo relativo a la seguridad de la información personal; utilizando contraseñas, encriptación y transacciones seguras.
7. Identificar carreras interdisciplinarias que son potenciadas por las ciencias de la computación.
8. Demostrar una actitud positiva a la resolución de problemas abiertos y a la programación (p.e. sentirse cómodo con la complejidad, persistencia, adaptabilidad, paciencia, propensión a la experimentación, creatividad, aceptación de retos)
9. Recoger y analizar datos que se emitan como resultado de múltiples ejecuciones de un programa informático.

Familia de estándares		
	Pensamiento Computacional (Computational Thinking – CT)	Prácticas de Computación y Programación (Computing Practices and Programming – CPP)
Nivel 3	<i>Grados K9-K12 (L3.CT)</i>	<i>Grados K9-K12 (L3.CPP)</i>
(‘Level 3’ – L3)	<p>El estudiante será capaz de:</p> <ol style="list-style-type: none"> 1. Utilizar funciones y parámetros predefinidos, clases y métodos para dividir un problema complejo en partes más simples. 2. Describir el proceso de desarrollo del software utilizado para resolver los problemas en el mismo (p.e. diseño, codificación, evaluación y verificación) 3. Explicar cómo la secuenciación, la selección, la iteración y la recursividad forman los bloques de construcción de los algoritmos. 4. Comparar distintas técnicas para analizar colecciones masivas de datos. 5. Describir las relaciones entre las representaciones binarias y hexadecimales. 6. Analizar la representación (e intercambio/traducción) entre distintas formas de información digital. 7. Describir cómo varios tipos de datos son almacenados en un sistema informático. 8. Utilizar la modelización y la simulación para representar y comprender fenómenos naturales. 9. Discutir/debatir el valor de la abstracción para gestionar la complejidad de los problemas. 10. Describir el concepto de procesamiento en paralelo como una estrategia para resolver problemas de magnitud superior 11. Describir cómo la computación comparte características con el arte y la música; en la traducción de una intención humana a un objeto o artefacto. 	<p>El estudiante será capaz de:</p> <ol style="list-style-type: none"> 1. Crear y organizar páginas web a través del uso de una amplia variedad de herramientas de diseño y programación web. 2. Utilizar dispositivos móviles y emuladores para diseñar, desarrollar, e implementar aplicaciones móviles (<i>‘apps’</i>) 3. Utilizar varios métodos de depuración y testeo para asegurar la corrección de los programas informáticos. 4. Aplicar técnicas de análisis, diseño e implementación para resolver problemas (p.e. usar uno o más modelos cíclicos-iterativos de software) 5. Usar APIs (<i>‘Application Program Interfaces’</i>) y librerías para facilitar soluciones de programación. 6. Seleccionar formatos de archivo adecuados para diversos tipos y usos de datos. 7. Describir/conocer una variedad de lenguajes de programación disponibles para resolver problemas y desarrollar sistemas. 8. Explicar el proceso de ejecución de un programa. 9. Explicar los principios de la seguridad informática a través del estudio de la encriptación, la criptografía, y las técnicas de autenticación. 10. Explorar una variedad de carreras profesionales en las cuales es central la computación. 11. Describir/conocer técnicas para localizar y recoger conjuntos de datos a pequeña y gran escala. 12. Describir/conocer cómo las funciones matemáticas y estadísticas, así como los conjuntos y la lógica, son utilizados en la computación.

c) Modelo longitudinal de desarrollo del PC: los itinerarios CAS

Finalmente, proveniente del contexto escolar británico en el que las Ciencias de la Computación han sido incluidas como asignatura obligatoria en su currículum a lo largo de todas las etapas obligatorias (UK Department of Education, 2013), desde los 5 años (*'Year 1'*) a los 16 años (*'Year 11'*), detallamos el *'Modelo longitudinal de desarrollo del PC: los itinerarios CAS'* (Dorling, 2015).

Estos itinerarios, denominados *'Computing Progression Pathways'*, han sido desarrollados en el marco del consorcio CAS (*'Computing at School'*) de Reino Unido, del que ya hemos hablado anteriormente. Su objetivo es trazar con detalle un camino o progresión a lo largo de la nueva asignatura de *'Ciencias de las Computación'*, fijando los logros que deben ser adquiridos por los estudiantes, específicamente entre el *'Year 1'* (5-6 años, equivalente a nuestro último curso de Educación Infantil) y el *'Year 9'* (13-14 años, equivalente a nuestro 2º ESO). El modelo distingue entre 6 itinerarios diferentes, todos ellos deben ser transitados por el estudiante:

- *'Algoritmos'* (*'Algorithms'*)
- *'Programación y Desarrollo'* (*'Programming & Development'*)
- *'Datos y Representación de Datos'* (*'Data & Data Representation'*)
- *'Hardware y Procesamiento'* (*'Hardware & Processing'*)
- *'Comunicación y Redes'* (*'Communication & Networking'*)
- *'Tecnologías de la Información'* (*'Information Technology'*)

Tal y como puede observarse, y a diferencia del esquema desarrollado por la CSTA de los Estados Unidos, el *'Pensamiento Computacional'* no aparece como itinerario diferenciado. Por el contrario, en el contexto británico, el PC se entiende como central, en el corazón, de todos los itinerarios propuestos; de manera que, para todos los logros que se fijan en ellos, se indica qué concepto del pensamiento computacional (*'computational thinking concept'*) se está desarrollando:

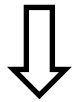
- *'Abstracción'* (*'Abstraction'* = AB)
- *'Descomposición'* (*'Decomposition'* = DE)
- *'Pensamiento Algorítmico'* (*'Algorithmic Thinking'* = AL)
- *'Evaluación'* (*'Evaluation'* = EV)
- *'Generalización'* (*'Generalization'* = GE)

En la siguiente Tabla 3.3 se reproducen los 6 itinerarios²¹⁹, indicando la progresión de logros que se espera a lo largo de cada uno de ellos, junto con el concepto computacional involucrado (Dorling, 2015)

²¹⁹ Accesible en su versión original inglesa en PDF: <http://community.computingschool.org.uk/files/5094/original.pdf>

Tabla 3.3. ‘CAS Computing Progressión Pathways’ (Traducción de Dorling, 2015)

<i>Progresión del estudiante</i>	<i>Itinerarios</i>					
	<i>‘Algoritmos’</i>	<i>‘Programación y Desarrollo’</i>	<i>‘Datos y Representación de Datos’</i>	<i>‘Hardware y Procesamiento’</i>	<i>‘Comunicación y Redes’</i>	<i>‘Tecnologías de la Información’</i>
<p>Comprende qué es un algoritmo y es capaz de expresar algoritmos simples-lineales (sin ramas) de manera simbólica (AL)</p> <p>Comprende que los ordenadores necesitan instrucciones precisas (AL)</p> <p>Muestra cuidado y precisión para evitar errores (AL)</p>	<p>Sabe que los usuarios pueden desarrollar sus propios programas, y lo muestra creando un programa simple en un entorno que no requiere texto (p.e. robots programables) (AL)</p> <p>Ejecuta, comprueba y modifica programas (AL)</p> <p>Comprende que los programas se ejecutan siguiendo instrucciones precisas (AL)</p>	<p>Reconoce que el contenido digital puede ser representado de muchas formas (AB) (GE)</p> <p>Distingue entre algunas de estas formas y puede explicar sus diferencias a la hora de comunicar información (AB)</p>	<p>Comprende que los ordenadores no tienen inteligencia, y que no hacen nada al menos que un programa sea ejecutado (AL)</p> <p>Reconocen que todo el software ejecutado en los dispositivos digitales está programado (AL) (AB) (GE)</p>	<p>Obtiene contenido de Internet utilizando un navegador web (AL)</p> <p>Comprende la importancia de comunicarse de manera segura y respetuosa on-line, y la necesidad de mantener en privado la información personal (EV)</p> <p>Sabe qué hacer cuando le afecta/preocupa algún contenido o es contactado on-line (AL)</p>	<p>Utiliza software bajo el control del profesor para crear, almacenar y editar contenido digital, utilizando nombres apropiados para carpetas y archivos (AB) (GE) (DE)</p> <p>Comprende que la gente interactúa con las computadoras.</p> <p>Comparte su uso/conocimiento de la tecnología en la escuela</p> <p>Conoce usos comunes de las tecnologías de la información más allá del aula (GE)</p> <p>Habla acerca de su trabajo y hace cambios para mejorarlo (EV)</p>	



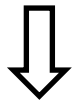
Year 1-2

'Algoritmos'	'Programación y Desarrollo'	'Datos y Representación de Datos'	'Hardware y Procesamiento'	'Comunicación y Redes'	'Tecnologías de la Información'
<p>Comprende que los algoritmos son implementados en los dispositivos digitales como programas informáticos (AL)</p> <p>Diseña algoritmos simples utilizando bucles y condiciones (p.e. sentencias 'If') (AL)</p> <p>Utiliza el razonamiento lógico para predecir resultados (AL)</p> <p>Detecta y corrige errores (p.e. depuración) en algoritmos (AL)</p>	<p>Utilizar operadores aritméticos, sentencias condicionales 'If', y bucles en sus programas (AL)</p> <p>Utiliza el razonamiento lógico para predecir el comportamiento de los programas (AL)</p> <p>Detecta y corrige errores simples semánticos (p.e. depuración) en los programas (AL)</p>	<p>Reconoce diferentes tipos de datos: texto, número (AB) (GE)</p> <p>Aprecia que los programas pueden funcionar con diferentes tipos de datos (GE)</p> <p>Reconoce que los datos pueden ser estructurados en tablas para hacerlos más útiles (AB) (DE)</p>	<p>Reconoce que existe un amplio abanico de dispositivos digitales que pueden ser considerados como un ordenador (AB) (GE)</p> <p>Reconoce y puede usar una variedad de dispositivos que soportan 'inputs' y 'outputs'</p> <p>Comprende cómo los programas especifican las funciones de un ordenador, dentro de su propósito general (AB)</p>	<p>Navega la red y puede llevar a cabo búsquedas web sencillas para recopilar contenido digital (AL) (EV)</p> <p>Evidencia un uso de los ordenadores seguro y responsable, conociendo diversas vías para informar de contenidos o contactos on-line inaceptables.</p>	<p>Utiliza la tecnología con una independencia creciente, organizando contenido digital con arreglo a un objetivo (AB)</p> <p>Muestra conciencia sobre la calidad del contenido digital recopilado (EV)</p> <p>Utiliza una amplia variedad de software para manipular y presentar contenido digital: datos e información (AL)</p> <p>Comparte sus experiencias con la tecnología en el aula y más allá de la misma (GE) (EV)</p> <p>Habla sobre su trabajo y hace mejoras a sus soluciones basadas en el feedback recibido (EV)</p>




Year 2-3

'Algoritmos'	'Programación y Desarrollo'	'Datos y Representación de Datos'	'Hardware y Procesamiento'	'Comunicación y Redes'	'Tecnologías de la Información'
<p>Diseña soluciones (algoritmos) que usan repeticiones y condicionales compuestos (p.e. <i>'If/else'</i>) (AL)</p> <p>Utiliza diagramas para expresar soluciones (AB)</p> <p>Utiliza el razonamiento lógico para predecir resultados (<i>'outputs'</i>), mostrando conciencia de su afectación por las entradas (<i>'inputs'</i>) (AL)</p>	<p>Crea/escribe programas que implementan algoritmos para lograr metas fijadas (AL)</p> <p>Declara y asigna variables (AB)</p> <p>Usa bucles indefinidos (<i>'post-tested loops'</i>, p.e. <i>'repeat until'</i>) y secuencias de sentencias condicionales compuestas (p.e. <i>'If/else'</i>) (AL)</p>	<p>Comprende la diferencia entre datos e información (AB)</p> <p>Sabe por qué ordenar datos en un archivo plano puede mejorar la búsqueda de información (EV)</p> <p>Utiliza filtros o puede realizar búsquedas unicriteriales de información (AL)</p>	<p>Sabe que los ordenadores recogen datos de varios dispositivos de <i>'input'</i>, incluidos sensores y software de aplicaciones (AB)</p> <p>Comprende la diferencia entre hardware y software de aplicaciones, y sus roles dentro del sistema informático (AB)</p>	<p>Comprende la diferencia entre Internet y un servicio de internet (p.e. la <i>'World Wide Web'</i>) (AB)</p> <p>Muestra conciencia de, y utiliza, una variedad de servicios de internet (p.e. VOIP²²⁰)</p> <p>Reconoce qué es un comportamiento aceptable e inaceptable cuando se utilizan tecnologías y servicios on-line.</p>	<p>Recopila, organiza y presenta datos e información en forma de contenido digital (AB)</p> <p>Crea contenido digital para logra una meta determinada, a través de la combinación de distintos paquetes de software y servicios de internet para comunicar con una audiencia más amplia (p.e. <i>'blogging'</i>) (AL)</p> <p>Hace mejoras apropiadas a sus soluciones basadas en el feedback recibido, y puede comentar acerca del éxito de dicha solución (EV)</p>



Year 3-4

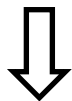
²²⁰ 'Voz sobre protocolo de internet' (VOIP), es un conjunto de recursos que hacen posible que la señal de voz viaje a través de Internet empleando el protocolo IP (Protocolo de Internet)

	<i>‘Algoritmos’</i>	<i>‘Programación y Desarrollo’</i>	<i>‘Datos y Representación de Datos’</i>	<i>‘Hardware y Procesamiento’</i>	<i>‘Comunicación y Redes’</i>	<i>‘Tecnologías de la Información’</i>
 Year 4-5	<p>Muestra conciencia de qué tareas son completadas mejor por personas o por ordenadores (EV)</p> <p>Diseña soluciones a través de la descomposición de un problema, y crea una sub-solución para cada una de estas partes (DE) (AL) (AB)</p> <p>Reconoce que existen diferentes soluciones para el mismo problema (AL) (AB)</p>	<p>Comprende la diferencia, y los usos apropiados, entre las sentencias condicionales simples (<i>‘If’</i>) y las compuestas (<i>‘If/else’</i>) (AL)</p> <p>Utiliza variables y operadores relacionales²²¹ dentro de un bucle para determinar su terminación (AL) (GE)</p> <p>Diseña, escribe y depura programas modulares utilizando procedimientos y funciones (AL) (DE)</p> <p>Sabe que un procedimiento o función puede ser utilizado para ocultar el detalle de una sub-solución (AL) (DE) (AB) (GE)</p>	<p>Realiza búsquedas más complejas de información (p.e. utilizando operadores Booleanos y operadores relacionales) (AL) (GE) (EV)</p> <p>Analiza y evalúa datos e información, y reconoce que una pobre calidad de los datos conduce a resultados no confiables y a conclusiones inexactas (AL) (EV)</p>	<p>Comprende por qué y cuándo se utilizan los ordenadores (EV)</p> <p>Comprende las funciones principales de un sistema operativo (DE) (AB)</p> <p>Conoce la diferencia entre redes físicas, sin cables y móviles (AB)</p>	<p>Comprende cómo usar de manera efectiva los motores de búsqueda, y sabe cómo se seleccionan los resultados de búsqueda; incluyendo que los buscadores utilizan programas de ‘rastreo web’²²² (AB) (GE) (EV)</p> <p>Selecciona, combina y usa distintos servicios de internet (EV)</p> <p>Demuestra un uso responsable de las tecnologías y servicios on-line, y conoce diversas vías para informar de sus inquietudes y preocupaciones.</p>	<p>Hace juicios sobre contenidos digitales, con el fin de evaluarlos y proponer mejoras dirigidas a audiencias determinadas (EV) (GE)</p> <p>Identifica la audiencia cuando diseña y crea contenido digital (EV)</p> <p>Comprende el potencial de las tecnologías de la información para el trabajo colaborativo cuando los ordenadores están conectados en red (GE)</p> <p>Utiliza criterios para evaluar la calidad de las soluciones; puede identificar mejoras haciendo retoques sobre una solución actual, y futuras soluciones (EV)</p>

²²¹ Los ‘operadores relacionales’ son aquéllos que permiten establecer relaciones del tipo ‘mayor que’, ‘menor que’, ‘igual’, ‘diferente’, etc...

²²² Un ‘rastreador web’ es un programa que inspecciona las páginas del World Wide Web de forma metódica y automatizada.

'Algoritmos'	'Programación y Desarrollo'	'Datos y Representación de Datos'	'Hardware y Procesamiento'	'Comunicación y Redes'	'Tecnologías de la Información'
<p>Comprende que la iteración es la repetición de un proceso, como un bucle (AL)</p> <p>Reconoce que existen diferentes algoritmos para el mismo problema (AL) (GE)</p> <p>Representa soluciones utilizando una notación estructurada (AL) (AB)</p> <p>Puede identificar similitudes y diferencias entre situaciones, y utilizarlo para resolver problemas análogos (reconocimiento de patrones) (GE)</p>	<p>Comprende que la programación funciona de puente entre las soluciones algorítmicas y las computadoras (AB)</p> <p>Tiene experiencia práctica en lenguajes textuales de programación de alto nivel²²³, utilizando librerías estandarizadas de código cuando programa (AB) (AL)</p> <p>Utiliza un amplio abanico de operadores y expresiones (p.e. operadores booleanos²²⁴), y los aplica en el contexto del control de los programas (AL)</p> <p>Selecciona apropiadamente entre los distintos tipos de datos (AL) (AB)</p>	<p>Sabe que los ordenadores y dispositivos digitales utilizan el sistema binario para representar todos los datos (AB)</p> <p>Comprende cómo los patrones de bits representan números e imágenes (AB)</p> <p>Sabe que los ordenadores transfieren los datos en sistema binario (AB)</p> <p>Comprende la relación entre el sistema binario y el tamaño de los archivos (no comprimidos) (AB)</p> <p>Define diferentes tipos de datos: números reales y booleanos (AB)</p> <p>Consulta datos en una tabla utilizando un lenguaje de consulta²²⁵ ('query language') habitual (AB)</p>	<p>Reconoce y comprende la función de las principales partes internas que forman la arquitectura básica de un ordenador (AB)</p> <p>Comprende los conceptos que hay detrás de los ciclos de ejecución y procesamiento (AB) (AL)</p> <p>Sabe que hay un amplio abanico de sistemas operativos y de software de aplicaciones para un mismo hardware (AB)</p>	<p>Comprende cómo los motores de búsqueda clasifican y ordenan los resultados de la misma (AL)</p> <p>Comprende cómo construir una página web estática utilizando HTML²²⁶ y CSS²²⁷ (AL) (AB)</p> <p>Comprende la transmisión de datos entre dispositivos digitales a través de las redes, incluyendo Internet (p.e. direcciones IP y conmutación de paquetes²²⁸) (AL) (AB)</p>	<p>Evalúa la adecuación de distintos dispositivos digitales, servicios de internet y software de aplicaciones para lograr metas determinadas (EV)</p> <p>Reconoce los aspectos éticos en torno a la aplicación de las tecnologías de la información más allá de la escuela.</p> <p>Diseña criterios para evaluar de manera crítica la calidad de las soluciones; utiliza dichos criterios para identificar mejoras y puede realizar retoques apropiados a la solución (EV)</p>



Year 5-6

²²³ Un lenguaje de programación 'de alto nivel' se caracteriza por expresar el algoritmo de una manera adecuada a la capacidad cognitiva humana, en lugar de la capacidad ejecutora de las máquinas.


²²⁴ Los 'operadores booleanos' son operadores de tipo lógico; los principales son: AND, OR y NOT

²²⁵ Un 'lenguaje de consulta' es un lenguaje informático usado para hacer consultas en bases de datos y sistemas de información.

²²⁶ HTML, siglas de '*HyperText Markup Language*' ('lenguaje de marcas de hipertexto'), hace referencia al lenguaje de marcado para la elaboración de páginas web.

²²⁷ Hoja de estilo en cascada o CSS (siglas en inglés de '*cascading style sheets*') es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML

²²⁸ La conmutación de paquetes es un método de envío de datos en una red de computadoras.


	<i>‘Algoritmos’</i>	<i>‘Programación y Desarrollo’</i>	<i>‘Datos y Representación de Datos’</i>	<i>‘Hardware y Procesamiento’</i>	<i>‘Comunicación y Redes’</i>	<i>‘Tecnologías de la Información’</i>
 Year 6-7	<p>Comprende que una solución recursiva a un problema aplica repetidamente la misma solución a casos particulares del mismo (AL) (GE)</p> <p>Reconoce que algunos problemas comparten las mismas características y utilizan el mismo algoritmo para ser resueltos (AL) (GE)</p> <p>Entiende la noción de rendimiento de un algoritmo, y aprecia que algunos algoritmos tienen distinto nivel de rendimiento aplicados en una misma tarea (AL) (EV)</p>	<p>Utiliza sentencias condicionales anidadas (AL)</p> <p>Aprecia la necesidad de, y escribe, funciones personalizadas a través del uso de parámetros (AL) (AB)</p> <p>Conoce la diferencia entre, y usa apropiadamente, procedimientos y funciones (AL) (AB)</p> <p>Comprende y utiliza la negación en los operadores (AL)</p> <p>Utiliza y manipula estructuras de datos unidimensionales (AB)</p> <p>Detecta y corrige errores sintácticos (AL)</p>	<p>Comprende cómo los números, las imágenes, los sonidos, o los conjuntos de caracteres pueden usar los mismos patrones de bits (AB) (GE)</p> <p>Realiza operaciones simples utilizando patrones de bits (p.e. adición binaria) (AB) (AL)</p> <p>Comprende la relación entre la resolución y la profundidad de color de una imagen digital, incluido su efecto sobre el tamaño del archivo (AB)</p> <p>Distingue entre los datos usados en un programa particular (una variable) y la estructura de almacenamiento para esos datos (AB)</p>	<p>Comprende la arquitectura Von Neumann²²⁹ en relación con los ciclos de ejecución, incluyendo cómo los datos son almacenados en la memoria (AB) (GE)</p> <p>Comprende las operaciones y funciones básicas de la memoria localizada y direccionable (AB)</p>	<p>Conoce los nombres del hardware (p.e. <i>‘hubs’</i>²³⁰, <i>‘routers’</i>²³¹, <i>‘switches’</i>²³²) y de los protocolos (p.e. SMTP, iMAP, POP, FTP, TCP/ IP) asociados con las redes de sistemas informáticos.</p> <p>Utiliza las tecnologías y servicios online de manera segura, y sabe cómo identificar e informar de conductas inapropiadas (AL)</p>	<p>Justifica la elección de, y combina y usa de manera independiente, múltiples dispositivos digitales, servicios de internet y software de aplicaciones para lograr unas metas determinadas (EV)</p> <p>Evalúa la fiabilidad del contenido digital, y toma en consideración la usabilidad de los distintos aspectos del diseño visual a la hora de crear artefactos digitales para una audiencia conocida (EV)</p> <p>Identifica y explica cómo el uso de la tecnología puede impactar sobre la sociedad.</p> <p>Diseña criterios para los usuarios para evaluar la calidad de las soluciones, utiliza el feedback de los usuarios para identificar mejoras, y puede realizar ajustes adecuados a la solución (EV)</p>

²²⁹ La arquitectura Von Neumann, también conocida como modelo de Von Neumann o arquitectura Princeton, es una arquitectura de computadoras basada en la descrita en 1945 por el matemático y físico John von Neumann y otros.


²³⁰ Concentrador (*‘hub’*) es el dispositivo que permite centralizar el cableado de una red de computadoras, para luego poder ampliarla.

²³¹ Un *‘router’* (también conocido como enrutador o encaminador de paquetes) es un dispositivo que proporciona conectividad a nivel de red.

²³² Conmutador (*‘switch’*) es el dispositivo digital lógico de interconexión de equipos que opera en la capa de enlace de datos. Su función es interconectar dos o más segmentos de red.

	<i>‘Algoritmos’</i>	<i>‘Programación y Desarrollo’</i>	<i>‘Datos y Representación de Datos’</i>	<i>‘Hardware y Procesamiento’</i>	<i>‘Comunicación y Redes’</i>	<i>‘Tecnologías de la Información’</i>
 <i>Year 7-8</i>	Reconoce que el diseño de un algoritmo es distinto de su expresión particular en un lenguaje de programación (que dependerá de los comandos y sintaxis de programación disponibles) (AL) (AB)	Aprueba el efecto del alcance de una variable (p.e. una variable local no puede ser accedida desde el exterior de su función) (AB) (AL)	Conoce la relación entre la representación de los datos y la calidad de los datos (AB)	Sabe que los procesadores tienen conjuntos de instrucciones, y que éstos se relacionan con las instrucciones de bajo nivel que son llevadas a cabo por el ordenador (AB) (AL) (GE)	Conoce el propósito del hardware y los protocolos asociados con las redes de sistemas informáticos (AB) (AL)	Emprende proyectos creativos que recogen, analizan, y evalúan datos para cubrir las necesidades de un determinado grupo de usuarios (AL) (DE) (EV)
	Evalúa la efectividad de algoritmos y modelos aplicados a problemas similares (AL) (AB) (GE)	Comprende y aplica la transferencia de parámetros (AB) (GE) (DE)	Comprende la relación entre el sistema binario y los circuitos eléctricos, incluyendo la lógica Booleana (AB)		Comprende el modelo cliente-servidor, incluido cómo las páginas web dinámicas utilizan secuencias de comandos del lado del servidor, y procesan y almacenan datos introducidos por el usuario (AL) (AB) (DE)	Diseña de manera efectiva artefactos digitales para una audiencia amplia y remota (AL) (DE)
	Reconoce qué información puede ser filtrada e ignorada para generalizar la soluciones a un problema (AL) (AB) (GE)	Comprende la diferencia entre, y usa, los bucles indefinidos ²³³ ‘pre-tested’ (p.e. ‘while loop’), y ‘post-tested’ (p.e. ‘until loop’) (AL)	Comprende cómo, y por qué, los valores son datos escritos o codificados en muchos lenguajes diferentes, cuando son manipulados durante los programas (AB)		Reconoce que la persistencia y perdurabilidad de los datos en Internet requiere de una protección cuidados de la identidad y la privacidad on-line	Considera las propiedades de los medios cuando los incorpora a sus artefactos digitales (AB)
	Utiliza el razonamiento lógico para explicar cómo funciona un algoritmo (AL) (AB) (DE)	Aplica una aproximación modular a la detección y corrección de errores (AB) (DE) (GE)				Documenta el feedback de los usuarios, las mejoras identificadas y los ajustes realizados a la solución (AB)
	Representa algoritmos utilizando un lenguaje formal-estructurado (AL) (DE) (AB)					Explica y justifica cómo el uso de la tecnología impacta en la sociedad, desde una perspectiva tanto social como económica, política, legal, ética y moral (EV)

²³³ Un bucle indefinido, en vez de repetirse un número determinado de veces, lo hace mientras se cumpla una condición (‘while loop’), o hasta que una condición se cumpla (‘until loop’)

	<i>‘Algoritmos’</i>	<i>‘Programación y Desarrollo’</i>	<i>‘Datos y Representación de Datos’</i>	<i>‘Hardware y Procesamiento’</i>	<i>‘Comunicación y Redes’</i>	<i>‘Tecnologías de la Información’</i>
 Year 8-9	Diseña la solución a un problema que depende de soluciones a casos particulares dentro del mismo problema (recursividad) (AL) (DE) (AB) (GE)	Diseña y escribe programas modulares anidados que refuerzan la reusabilidad utilizando sub-rutinas siempre que sea posible (AL) (AB) (GE) (DE)	Realiza operaciones utilizando patrones de bits (p.e conversión del sistema binario al hexadecimal, sustracción binaria, etc...) (AB) (AL) (GE)	Tiene experiencia práctica con algún lenguaje de programación de bajo nivel ²³⁵ (AB) (AL) (DE) (GE)	Comprende el hardware asociado con las redes de sistemas informáticos, incluyendo WANs ²³⁷ y LANs ²³⁸ ; comprende su propósito y cómo funcionan, incluyendo las direcciones MAC ²³⁹ (AB) (AL) (DE) (GE)	Comprende los aspectos éticos en torno a la aplicación de las tecnologías de la información, y la existencia de marcos legales que rigen su uso (p.e. ‘Data Protection Act’ ²⁴⁰ , ‘Computer Misuse Act’, ‘Copyright’, etc.) (EV)
	Comprende que algunos problemas no pueden ser resueltos computacionalmente (AB) (GE)	Comprende la diferencia entre un bucle ‘while’ (‘repetir mientras’) y un bucle ‘for’ que usa un contador (AL) (AB)	Comprende y puede explicar la necesidad de comprimir los datos, y puede ejecutar métodos simples de compresión (AL) (AB)	Comprende y puede explicar la Ley de Moore ²³⁶ (GE)	Comprende y puede explicar la capacidad multitarea de los ordenadores (AB) (AL) (DE)	
		Comprende y utiliza estructuras de datos bidimensionales (AB) (DE)	Conoce qué es un base de datos relacional ²³⁴ (‘relational database’), y comprende los beneficios de almacenar datos en múltiples tablas (AB) (GE) (DE)			

Concepto de Pensamiento Computacional: ‘Abstracción’ (AB); ‘Descomposición’ (DE); ‘Pensamiento Algorítmico’ (AL); ‘Evaluación’ (EV); ‘Generalización’ (GE)

²³⁴ Una ‘base de datos relacional’ (BDR) es un tipo de base de datos (BD) que permite establecer interconexiones o relaciones entre los datos que la forman.

²³⁵ Un lenguaje de programación de características ‘bajo nivel’ es aquel en el que sus instrucciones ejercen un control directo sobre el hardware y están condicionados por la estructura física de las computadoras que lo soportan.

²³⁶ La Ley de Moore expresa que aproximadamente cada dos años se duplica el número de transistores en un microprocesador.

²³⁷ Una red de área amplia, o WAN, (‘Wide Area Network’ en inglés), es una red de computadoras que abarca varias ubicaciones físicas, proveyendo servicio a una zona, un país, incluso varios continentes

²³⁸ Una red de área local o LAN (por las siglas en inglés de ‘Local Area Network’) es una red de computadoras que abarca un área reducida a una casa, un departamento o un edificio.

²³⁹ En las redes de computadoras, la dirección MAC (siglas en inglés de ‘Media Access Control’; en español ‘Control de Acceso al Medio’) es un identificador de 48 bits (6 bloques hexadecimales) que corresponde de forma única a una tarjeta o dispositivo de red.

²⁴⁰ Equivalente a nuestra Ley de Protección de Datos.

3.2.2. Experiencias y recursos para el desarrollo del pensamiento computacional

Tal y como venimos comentando a lo largo de esta tesis, la programación informática (*'coding'*) es la manera más habitual y evidente para el desarrollo del pensamiento computacional (PC) (Basogain, Olabe, & Olabe, 2015; Compañ-Rosique, Satorre-Cuerda, Llorens-Largo, Molina-Carmona, 2015). La programación informática implica necesariamente que el sujeto ponga en juego su PC, al tener que manejar los distintos conceptos computacionales de manera coordinada y orientada a la resolución del problema al que el programa trata de dar respuesta. Dicho de otra manera, la programación informática hace que el pensamiento computacional cobre vida, al proyectarse y cristalizar en la construcción de artefactos digitales concretos (Kafai & Burke, 2013; Resnick *et al.*, 2009). Pero, igualmente, el pensamiento computacional, como proceso cognitivo de resolución de problemas que es, puede ser desarrollado en entornos que no incluyen la programación informática y, aún más, que no incluyen siquiera ordenadores (entornos *'unplugged'*)

Desde esta dualidad, en este epígrafe: en primer lugar realizamos una revisión de la literatura científica que da cuenta de recientes experiencias de desarrollo del pensamiento computacional explícitamente a través de la programación informática, y en etapas escolares pre-universitarias (K-12); en segundo lugar, mostramos algunos ejemplos de experiencias y recursos para el desarrollo del pensamiento computacional en entornos *'unplugged'*, sin necesidad de ordenadores.

A modo de síntesis integradora de ambos polos, se finaliza con un sub-epígrafe que detalla el programa-curso trimestral *'K-8 Intro to Computer Science'* (Code.org, 2015a), objeto central de la evaluación de programas que abordaremos en el Capítulo 7 de esta tesis, y excelente ejemplo de curso que integra en su estructura tanto sesiones de *'coding on-line'* como sesiones *'unplugged'*.

- a) Revisión de experiencias de desarrollo del pensamiento computacional a través de la programación informática en etapas escolares pre-universitarias.

Es la proliferación y disponibilidad de los nuevos lenguajes visuales de programación, gratuitos y de aprendizaje *'amigable'* como Scratch (Resnick *et al.*, 2009) o Alice (Cooper, Dann, & Pausch, 2000; Graczyńska, 2010), el hecho fundamental que viene potenciando en los últimos años un renovado interés por introducir la programación informática en los entornos escolares pre-universitarios (Lye & Koh, 2014)

En contraste con los lenguajes tradicionales de programación (lenguajes *'textuales'* como Java o C++), cuya forma de representación y sintaxis se aproxima más a la forma de procesar de un ordenador (Smith, Cypher, & Tesler, 2000); los lenguajes visuales de programación están más cercanos a la forma humana de representar el lenguaje (Figura 3.13). Además, los lenguajes visuales de programación (p.e. Scratch o Alice) son más sencillos de comprender y comenzar a utilizar, pues proveen feedback visual en tiempo real del programa que se está escribiendo en forma de objetos digitales animados, permitiendo al estudiante crear animaciones y juegos digitales-interactivos que posteriormente puede compartir.

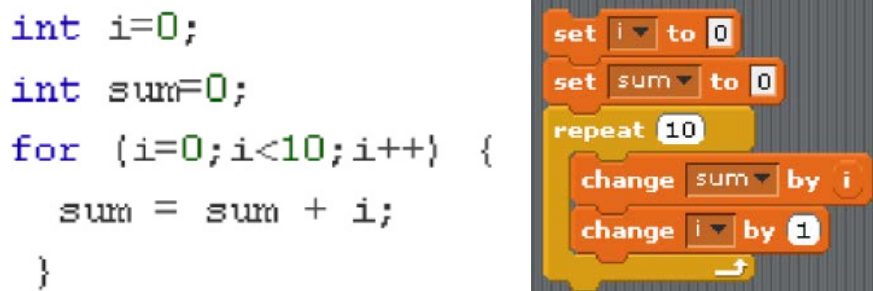


Figura 3.13. El mismo programa informático escrito en un lenguaje textual (Java, a la izquierda) y en un lenguaje visual (Scratch, a la derecha)

Así, en conjunto, parece que los lenguajes visuales de programación son más adecuados que los lenguajes tradicionales ‘textuales’ a la hora de facilitar, en contextos escolares pre-universitarios, las 3 dimensiones del pensamiento computacional definidas por Brennan y Resnick (2012): conceptos computacionales (*‘computational concepts’*), prácticas computacionales (*‘computational practices’*) y perspectivas computacionales (*‘computational perspectives’*). Así lo defienden Lye y Koh (2014) cuando dicen que:

- Los lenguajes visuales de programación reducen o eliminan elementos de sintaxis innecesarios que están presentes en los lenguajes textuales (p.e. el uso de paréntesis, llaves, y puntos y coma); y los comandos-órdenes de los lenguajes visuales están más próximos al lenguaje humano natural-hablado. Al utilizar los lenguajes visuales, habitualmente el estudiante sólo necesita ‘arrastrar y soltar’ (*‘drag-and-drop’*) los bloques de comandos, que encajan unos con otros según su funcionalidad, para ir construyendo el programa. Con estas características, los lenguajes visuales de programación ayudan a reducir la carga cognitiva del sujeto y “*permiten a los estudiantes focalizarse en la lógica y las estructuras implicadas en la programación, en vez de preocuparse sobre los aspectos mecánico-sintácticos de la escritura del programa*” (Kelleher & Pausch, 2005). Así las cosas, parece que los lenguajes visuales de programación potencialmente permiten a los estudiantes, especialmente en etapas escolares, adquirir los conceptos computacionales (*‘computational concepts’*) más fácilmente, sin la necesidad farragosa de aprender sintaxis de programación complejas como las de los lenguajes textuales.
- Igualmente, los lenguajes visuales de programación facilitan que los estudiantes promuevan y desarrollen la distintas prácticas computacionales (*‘computational practices’*), dado que los resultados de su tarea de programación pueden ser visualizados en forma de objetos digitales animados. Dicha visualización hace que algunas prácticas computacionales, como el testeo (*‘testing’*) o la depuración (*‘debugging’*) de programas, sean cognitivamente menos demandantes. Además, este tipo de lenguajes convierten a la tecnología progresivamente de un ‘objeto de aprendizaje’ a un ‘compañero de aprendizaje’ (*‘technology as a partner’* en la terminología de Jonassen *et al.*, 2008), y posibilitan que los estudiantes expandan y transfieran estas prácticas computacionales a la potenciación de su habilidad general de solución de problemas (Lin & Liu, 2012; Ratcliff & Anderson, 2011)
- Finalmente, los lenguajes visuales de programación enganchan a los estudiantes en la construcción de productos digitales multimedia, y por tanto convierten la actividad de

programación en una vía a través de la cual los estudiantes pueden expresar sus ideas. Dicha actividad puede contribuir a perfilar la perspectiva computacional (*'computational perspective'*) del estudiante acerca del mundo tecnológico-digital que le rodea. Estos lenguajes visuales permiten que los estudiantes desarrollen un alfabetismo digital de alto nivel para crear, compartir y remezclar recursos digitales (Mills, 2010; Ng, 2012); y en ese proceso los estudiantes pasan de ser consumidores pasivos a creadores activos de tecnología (Resnick *et al.*, 2009). Por todo ello, los lenguajes visuales de programación, como herramientas especialmente adecuadas para edades escolares, se están convirtiendo en entornos de máxima importancia para proporcionar a los estudiantes experiencias de alfabetización digital de máximo nivel (Mills, 2010)

Sin embargo, en el actual estado de la literatura científica, hay aún escasez de artículos que exploren específicamente el desarrollo del pensamiento computacional a través de la programación en contextos escolares K-12 (pre-universitarios) (Grover & Pea, 2013a). Recientemente, Lye y Koh (2014) han realizado una revisión sistemática al respecto, encontrando tan solo 9 artículos que den cuenta de experiencias e investigaciones que cumplan con dichos parámetros (*'computational thinking' AND 'programming' AND 'K-12'*). Partimos de dicha revisión, ampliándola con otros 3 artículos de corte empírico, publicados en el año 2015 en el primer monográfico sobre pensamiento computacional en lengua española (Bender, Urrea, & Zapata-Ros, 2015); en el cual, este doctorando ha tenido el privilegio de participar (Moreno-León, Robles, & Román-González, 2015)

En la Tabla 3.4 se sintetizan los 12 artículos revisados, indicándose: autor/es del trabajo; número, perfil y edad de los estudiantes participantes en el estudio; duración de la intervención; lenguaje de programación utilizado; y asignatura marco de la intervención.

Tabla 3.4. Relación de artículos, en orden alfabético, que estudian explícitamente el desarrollo del pensamiento computacional a través de la programación en contextos escolares K-12

<i>Autor/es</i>	<i>Nº de Participantes</i>	<i>Perfil de los participantes</i>	<i>Edad de los participantes</i>	<i>Duración de la intervención</i>	<i>Lenguaje de programación</i>	<i>Asignatura marco</i>
Bender & Urrea (2015)	45	Niños/as de Educación Primaria	8-12 años	12 meses	Logo Visual	Arte y Diseño
Burke (2012)	10	Estudiantes masculinos en una actividad extraescolar voluntaria	12-14 años	7 semanas	Scratch	Lengua
Burker (2015)	14	Estudiantes masculinos en un campamento de verano	11-14 años	2 semanas	Logo Visual	Tecnología
Denner <i>et al.</i> (2012)	59	Estudiantes féminas en una actividad extraescolar voluntaria	12-14 años	14 meses	Stagecast Creator	Informática

Fessakis <i>et al.</i> (2013)	10	Niños/as de Educación Infantil	5-6 años	75 minutos	Logo Visual	Matemáticas
Kahn <i>et al.</i> (2011)	31	Estudiantes de alto rendimiento	9-13 años	7-10 semanas	Toontalk	Matemáticas
Kzakoff & Bers (2012)	58	Niños/as de Educación Infantil	4,5-6,5 años	20 horas	Entorno híbrido de programación-robótica	Informática
Lee (2010)	1	Caso único en actividad extraescolar	9 años	6 meses	Scratch	Lengua
Lin & Liu (2012)	3	Participantes en un campamento de verano	9-10 años	5 días	Logo Visual	Informática
Miller (2009)	1	Discapacitado auditivo	13 años	3 meses	Logo Visual	Lengua
Moreno-León <i>et al.</i> (2015)	109	Niños/as de Educación Primaria y Secundaria	10-14 años	1 hora	Scratch	Informática
Wang & Chen (2010)	115	Estudiantes de instituto con experiencia previa en Flash	12-14 años	6 semanas	Flash Action Script	Informática

Tal y como puede observarse en la Tabla 3.4, hay una gran variedad de intervenciones: desde estudios con más de 100 estudiantes (p.e. [Moreno-León *et al.*, 2015](#); [Wang & Chen, 2010](#)) hasta estudios de caso único ([Lee, 2010](#); [Miller, 2009](#)); con sujetos cuyas edades oscilan entre los 4 y 14 años, abarcando las etapas de Educación Infantil, Primaria y Secundaria; tanto en entornos educativos formales (p.e. [Bender & Urrea, 2015](#)) como no formales (p.e. [Lin & Liu, 2012](#)); tanto en grupos mixtos (p.e. [Fessakis *et al.*, 2013](#)), como en grupos ‘*single sex*’ ya sean masculinos ([Burke, 2012](#)) o femeninos ([Denner *et al.*, 2012](#)), e incluso con población discapacitada ([Miller, 2009](#)); con intervenciones cuya duración abarca desde una sola sesión de clase ([Moreno-León *et al.*, 2015](#)), a un curso académico completo ([Bender & Urrea, 2015](#)).

Los lenguajes de programación más utilizados fueron Scratch y versiones ‘visuales por bloques’ de Logo. Ambos lenguajes tienen en común ser ‘*low floor, high ceiling, wide walls*’ ([Resnick *et al.*, 2009](#)); características que los hacen ideales para ser utilizados en poblaciones escolares que pretendan desarrollar su pensamiento computacional:

- ‘*Low floor*’ (de ‘suelo bajo’): es decir, son lenguajes que no requieren que el estudiante tenga conocimientos previos de programación para comenzar a ser utilizados. Son sencillos e intuitivos, posibilitando que el estudiante novel comience a experimentar con ellos, obteniendo resultados de manera inmediata. Estos lenguajes han bajado las ‘barreras de entrada’ a la programación.

- *'High ceiling'* (de 'techo alto'): es decir, pese a su sencillez inicial, permiten a los estudiantes crear programas extraordinariamente sofisticados. De manera que el estudiante puede avanzar en la escritura de programas de creciente complejidad sin encontrar techo a su aprendizaje. Scratch permite, incluso, que el estudiante cree sus propios bloques personalizados de programación
- *'Wide walls'* (de 'paredes anchas'): es decir, permiten proyectarse y transferirse a gran cantidad de áreas. Con estos lenguajes se puede hacer una narración multimedia, un videojuego interactivo, codificar una melodía, diseñar una geometría, modelizar una ley física, etc...

Así, en la docena de estudios revisados, además de la natural integración de las experiencias de aprendizaje en la asignatura de Informática, la programación se proyectó sobre:

- La asignatura o área de Lengua: por ejemplo, Burke (2012) sugiere en las conclusiones de su estudio que Scratch ofrece *"un nuevo medio a través del cual los niños pueden ejercitar las habilidades de composición escrita adquiridas durante las clases tradicionales de Lengua, al tiempo que adicionalmente adquieren los beneficios de aprender 'coding' en edades tempranas"* (Burke, 2012, p. 131)
- La asignatura o área de Matemáticas: a través del *'coding'* se adquieren conceptos matemáticos, y éstos se externalizan al representarlos en forma de objetos digitales programados por los estudiantes (p.e. adquisición de conceptos geométricos a través de la codificación de formas). Pueden consultarse al respecto los estudios de Fessakis *et al.* (2013) o Khan *et al.* (2011)
- La asignatura o área de Tecnología: por ejemplo, en Burker (2015) se relata cómo los estudiantes, a través de una versión visual 'por bloques' del lenguaje Logo, programan una serie de diseños geométricos que, posteriormente, son procesados y exportados a la realidad física a través de la tecnología de impresión 3D. Es un buen ejemplo de la creciente fusión entre mundo digital y físico, entre *'hackers'* y *'makers'*. Como dice el autor: *"El software y el hardware deben ser vistos como herramientas en el aula moderna, indistinguibles en importancia y potencial creativo, como lo son el lápiz y el papel"* (Burker, 2015, p. 1)
- Finalmente, también encontramos integraciones del *'coding'* para el desarrollo del pensamiento computacional en el área de arte y diseño (Bender & Urrea, 2015). En la Figura 3.14. se muestra una galería de diseños artísticos producidos por los estudiantes que participaron en este estudio.

Recordemos que, precisamente, el enfoque del pensamiento computacional pone el énfasis en la transferencia de esta habilidad de solución de problemas a gran variedad de áreas y asignaturas. Vista desde el prisma del pensamiento computacional, la programación informática, por tanto, no sería un objetivo de aprendizaje en sí mismo (*'Learn to Code'*) sino un medio, o un enfoque, a través del cual potenciar el aprendizaje de otras áreas (*'Code to Learn'*); tal y como ya adelantamos en el apartado 2.7 de transición entre el Capítulo 2 y el Capítulo 3.

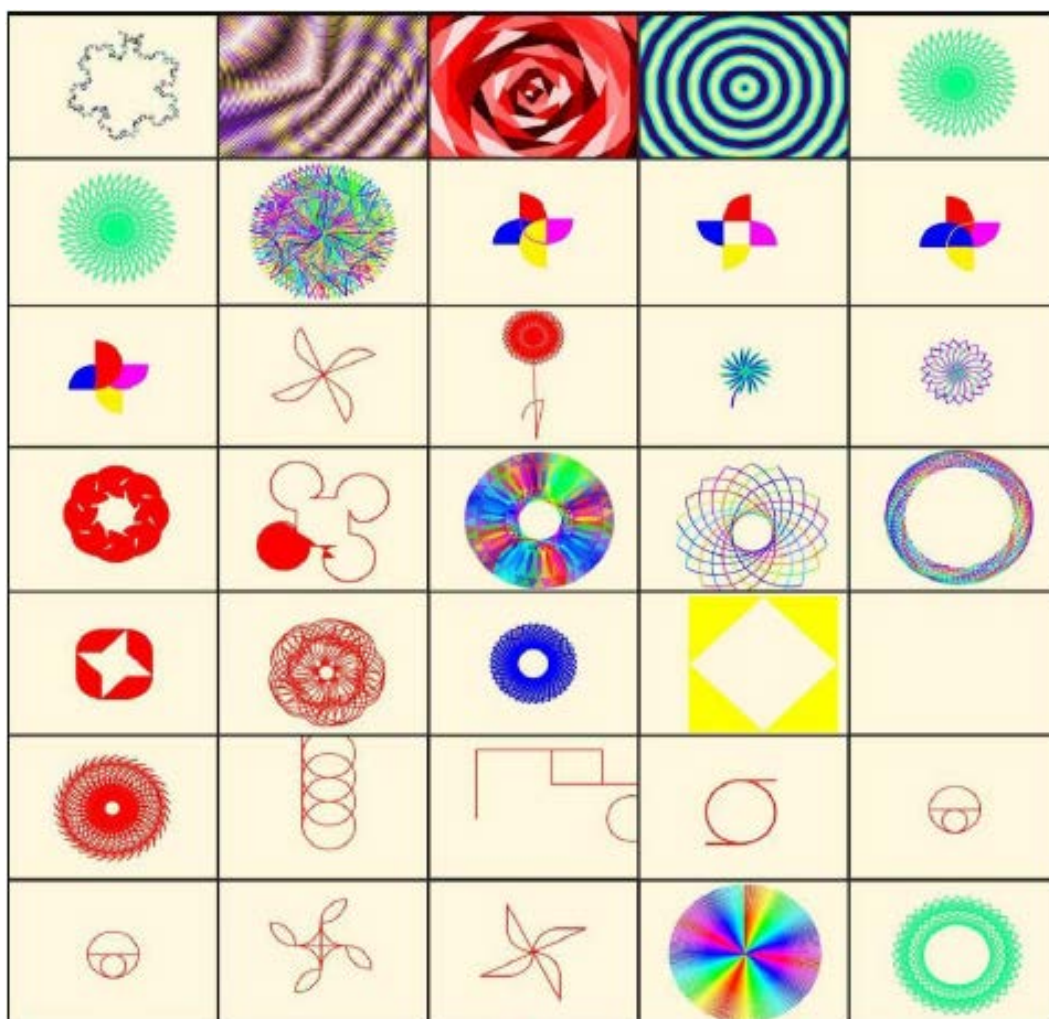


Figura 3.14. Galería de diseños artísticos producidos por estudiantes a través del lenguaje Logo (Bender & Urrea, 2015)

Por otra parte, podemos comparar esta docena de estudios en función de cuál es el tipo de diseño de investigación utilizado; y según cuál o cuáles de las dimensiones del pensamiento computacional (Brennan & Resnick, 2012) han sido abordadas en el trabajo. Tal y como puede apreciarse en la Tabla 3.5, prácticamente no se encuentran diseños de tipo experimental, con grupo experimental y de control, una carencia que trataremos de contribuir a paliar con la evaluación de programas que presentamos en el Capítulo 7. En los dos estudios experimentales revisados (Kazakoff & Bers, 2012; Wang & Chen, 2010), se informa de resultados a favor de los grupos experimentales en los conceptos computacionales medidos (p.e. bucles y variables)

Por otra parte, en la Tabla 3.5 se aprecia que la mayoría de estudios se focalizan en los ‘conceptos computacionales’ (10 artículos). En menor medida, aparecen trabajos que tomen en consideración las ‘prácticas computacionales’ (5 artículos): por ejemplo, Moreno-León *et al.* (2015) muestran los efectos positivos de su intervención en una práctica que implica testeo (*‘testing’*), depuración (*‘debugging’*) y mejora por iteración (*‘iterating’*) de proyectos Scratch. Y, por último, apenas encontramos trabajos que tomen en consideración las ‘perspectivas computacionales’ (2 artículos), destacando el trabajo de Burke (2012) en el cual los estudiantes reflexionan y se expresan a través de la codificación de narraciones e historias digitales con Scratch. Parece pues, que, por el momento, existe cierto vacío de trabajos que integren las 3 dimensiones del PC propuestas.

Tabla 3.5. Comparación de artículos según diseño de investigación y dimensiones del pensamiento computacional

<i>Autor/es</i>	<i>Diseño de investigación</i>	<i>Intervención</i>	<i>Dimensiones del Pensamiento Computacional</i>		
			<i>Conceptos Computacionales</i>	<i>Prácticas Computacionales</i>	<i>Perspectivas Computacionales</i>
Bender & Urrea (2015)	Análisis cualitativo de productos	Concurso de programación entre escolares	*	*	
Burke (2012)	Estudio de caso	Narración digital	*		*
Burker (2015)	Estudio de caso	Creación e impresión 3D de diseños	*	*	
Denner <i>et al.</i> (2012)	Análisis cualitativo de productos	Creación de juegos	*		
Fessakis <i>et al.</i> (2013)	Estudio de caso	Guiada por el profesor a toda el aula con pizarra digital interactiva		*	
Kahn <i>et al.</i> (2011)	Estudio de caso	Modelado computacional de conceptos matemáticos			*
Kazakoff & Bers (2012)	Experimental	Experimental: Uso de un lenguaje ‘tangible’ de programación combinando bloques físicos y en pantalla	*		
		Control: Actividades tradicionales de arte			
Lee (2010)	Estudio de caso	Trabajo individual	*		
Lin & Liu (2012)	Estudio de caso	Programación por pares	*	*	
Miller (2009)	Estudio de caso	Modelado	*		
Moreno-León <i>et al.</i> (2015)	Encuesta	Desarrollo de proyectos a través de ‘feed-back’ automatizado formativo	*	*	
Wang & Chen (2010)	Experimental	Experimental: juegos que implican desafío y programación	*		
		Control: juegos que implican sólo desafío			

Si analizamos qué tipo de aproximaciones de intervención se están utilizando para potenciar el pensamiento computacional a través de la programación, encontramos 2 enfoques bien diferenciados:

- **Reforzamiento de conceptos computacionales:** en esta aproximación, los conceptos computacionales son reforzados con la ayuda del ordenador, que proporciona *'feed-back'* automático en un entorno de aprendizaje gamificado (p.e. Wang & Chen, 2010). Este enfoque parte de una concepción conductista del aprendizaje, en la cual la ejecución deseada es moldeada a través de técnicas de control de la conducta como refuerzo y castigo.

Se identifica con los recursos de aprendizaje del *'coding'* del tipo de *'resolución de problemas cerrados / puzzles / retos de programación'* (*'close-ended'*) que vimos en la taxonomía del epígrafe 2.4.1.

- **Construcción de programas con andamiaje:** en esta aproximación, los aprendices construyen sus propios programas con alguna ayuda o andamiaje. Este tipo de enfoque, que potencialmente podría abordar las 3 dimensiones del PC citadas anteriormente, descansa en el *'construccionismo'* que *"concede especial importancia al papel de las construcciones en el mundo, como un apoyo a las construcciones que están en la cabeza"* (Papert, 1994, p. 143); superando y diferenciándose así de las doctrinas puramente mentalistas (léase *'constructivismo'*). En el construccionismo, los estudiantes están activamente comprometidos en la construcción de conocimiento, a través de la creación de productos significativos para otros o para sí mismos (Kafai & Resnick, 1996). Para dicha creación de productos, los estudiante construyen programas informáticos con ayuda y andamiaje de distinto tipo: andamiaje proporcionado por el propio ordenador o *'computer scaffolding'* (Moreno-León et al., 2015); andamiaje del profesor o *'teacher's scaffolding'* (p.e. Burke, 2012; Fessakis et al., 2013; Kahn et al., 2011); guía parental (Lin & Liu, 2012); o aprendizaje por pares o *'peer to peer'* (Denner et al., 2012), que si se realiza *'on-line'* nos sitúa ya en una perspectiva *'conectivista'* (Siemens, 2005, 2010). La metodología de construcción de programas con andamiaje es consistente con numerosas evidencias de investigación que concluyen que el puro aprendizaje por exploración y descubrimiento no es un método efectivo para la adquisición de altas habilidades de programación (Mayer, 2004); mientras que el descubrimiento guiado o andamiado parece ser la vía más adecuada para ello (p.e. Clements & Merriman, 1988; Lehrer, Lee, & Jeong, 1999)

Esta aproximación se identifica con los recursos de aprendizaje del *'coding'* del tipo *'diseño y creación de objetos-proyectos digitales abiertos (narraciones, aplicaciones, simulaciones, etc...)'* (*'open-ended'*) que vimos en la taxonomía propuesta en el epígrafe 2.4.1.

Finalmente, Lye & Koh (2014), a partir de la exhaustiva revisión realizada, extraen una serie de recomendaciones para la investigación futura en esta área, que tomaremos en consideración para nuestros estudios empíricos:

- Realizar estudios que se contextualicen naturalmente en el aula ordinaria (*'naturalistic classroom settings'*), y con población escolar ordinaria; dado que la mayoría de investigación realizada hasta la fecha en edades pre-universitarias se ha orientado a contextos extraescolares y con poblaciones específicas.

- Realizar estudios que tomen en consideración, y de manera conjunta, ‘conceptos computacionales’, ‘prácticas computacionales’ y ‘perspectivas computacionales’.
 - Examinar el proceso mental del estudiante a la hora de programar: por ejemplo, a través de la grabación de las consolas de programación utilizadas por el sujeto durante el proceso (lo cual permitiría analizar los procedimientos de codificación llevados a cabo); sería análogo a grabar la pantalla de un escritor mientras utiliza un procesador de textos. Otra opción es solicitar a los estudiantes que verbalicen sus procesos cognitivos mientras programan.
 - Analizar los productos digitales que se hayan programado a través de una aproximación complementaria cuantitativa-cualitativa.
- b) Algunas experiencias y recursos para el desarrollo ‘*unplugged*’ del pensamiento computacional

Veamos brevemente un par de iniciativas para el desarrollo del pensamiento computacional a través de actividades ‘*unplugged*’, es decir, sin necesidad de ordenador.

Una de ellas es ‘*CS4FN*’,²⁴¹ (*‘Computer Science for Fun’*), promovida desde la Queen Mary University of London. ‘*CS4FN*’ es una página web, con un magazine digital asociado, desde la cual se proporcionan recursos que ilustran el lado divertido de las Ciencias de la Computación, y cómo contribuyen a la resolución de problemas y creación de artefactos en el mundo real. Por ejemplo, recientemente han publicado el especial “*The Magic of Computer Science*” (Curzon & McOwan, 2015), que propone actividades para aprender divertidos trucos de magia y mentalismo aprovechando el uso de algoritmos mentales (p.e. un algoritmo que permite adivinar la carta elegida al azar por el espectador a partir de una serie de preguntas aparentemente no relacionadas)

La otra iniciativa, más destacada si cabe, proviene de la Universidad de Canterbury (Nueva Zelanda) en colaboración con Google, y se denomina ‘*CS Unplugged*’,²⁴². Aunque tenga su origen en nuestras antípodas, es bien conocida y utilizada a lo largo de todo el planeta, y hace escasas fechas encontraba eco en la revista digital española ‘Educación 3.0’, en el post ‘*CS Unplugged: aprendiendo programación en el aula... ¡sin ordenadores!*’:

“La Universidad de Canterbury, de Nueva Zelanda, en colaboración con la empresa americana Google, llevan desde 1998 creando, renovando y rediseñando uno de esos recursos fundamentales para el mundo de la programación: Computer Science Unplugged²⁴³ que, por si fuera poco, son recursos completamente gratuitos y están disponibles para todo aquel que desee utilizarlos (...) Su principal activo es un libro de 243 páginas (Bell, Witten, & Fellows, 2015) con recursos para enseñar programación sin ordenadores de por medio, y que incluyen ejercicios desde la base más inicial hasta alcanzar a comprender el Test de Turing, pasando por el aprendizaje del código binario o una primera introducción en el mundo de la algorítmica (...) Todos los ejercicios están planteados de forma que puedan

²⁴¹ <http://www.cs4fn.org/>

²⁴² <http://csunplugged.org/>

²⁴³ Un tráiler de video muy ilustrativo de las actividades propuestas por ‘*CS Unplugged*’ se puede consultar en <https://youtu.be/KOYy4kyLEHs>

realizarse de forma sencilla, con apenas papel y algún lápiz o bolígrafo, o a lo sumo con fichas que están incluidas. En cada uno de ellos se plantean un conjunto de instrucciones sobre cómo plantearlos, además de por supuesto unos objetivos y edades mínimas recomendadas. CS Unplugged está diseñado para niños de edades a partir de los 7 años, aunque en algunos casos los ejercicios pueden ser para 9 u 11 años, en adelante, siendo un recurso fundamental para todo docente de la informática en primaria e incluso, en muchos casos, también en secundaria. Como decimos es completamente gratuito y aquí podéis acceder a su web oficial donde encontraréis toda la información” (Espeso, 2015, en línea)

Detallamos algo más a continuación una de las actividades propuestas por ‘CS Unplugged’ para que el lector adquiera una idea más clara de su naturaleza; en concreto, la actividad ‘Sorting Networks’²⁴⁴ (‘Redes de ordenación y clasificación’), en la cual los estudiantes representan físicamente el funcionamiento de un algoritmo sencillo de clasificación ‘en paralelo’. Es una actividad recomendada para edades a partir de 7 años, y en la cual se desarrolla el pensamiento computacional, especialmente la parte algorítmica, de forma física y colaborativa (Bell, Witten, & Fellows, 2015, p. 80-86). Para realizarla, en primer lugar se debe dibujar en el suelo (p.e. en el patio, con tiza) la siguiente red (Figura 3.15)

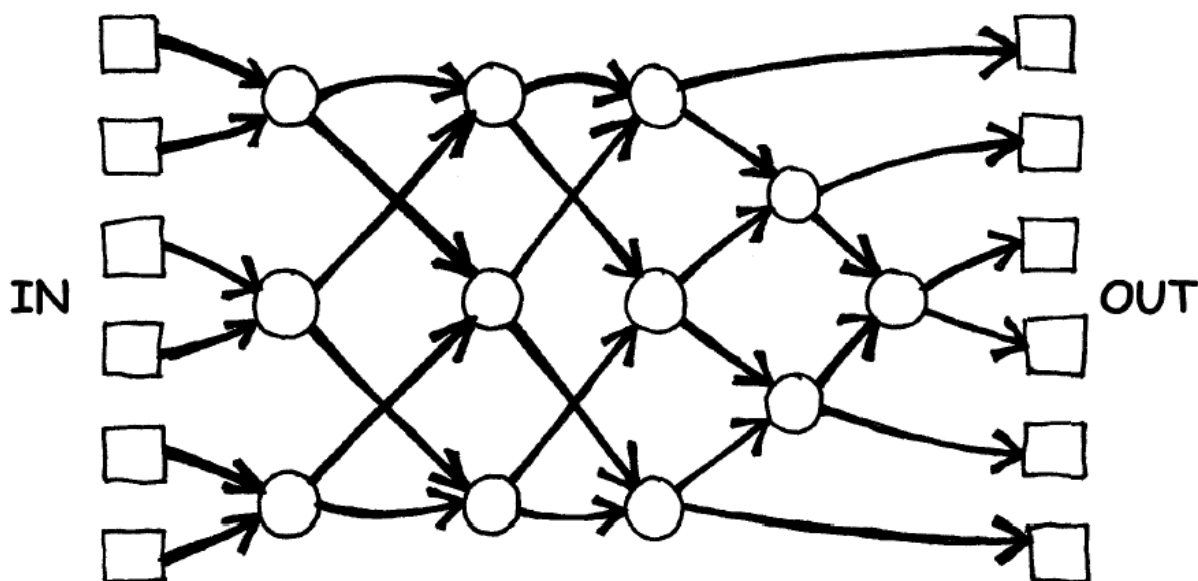


Figura 3.15. Red de ordenación y clasificación (Tomado de Bell, Witten, & Fellows, 2015, p. 81)

La actividad muestra al estudiante cómo las computadoras son capaces de ordenar conjuntos aleatorios de números, u otros datos, utilizando una red de clasificación (‘*sorting network*’). Para ello: a) se divide a la clase en grupos de 6 estudiantes; b) cada miembro del grupo coge aleatoriamente una tarjeta con un número; cada miembro comienza, aleatoriamente, en uno de los cuadrados de la parte izquierda de la red; d) los miembros van avanzando por la red según indican las flechas: cuando se llega a un círculo se debe esperar la llegada de otro miembro y comparar los números de ambos; e) el sujeto con el número menor toma el camino de la izquierda, y el sujeto con el número mayor toma el camino de la derecha; f) al final de la red, los números quedan ordenados de menor a mayor.

²⁴⁴ Véase un grupo de estudiantes realizando la actividad en <https://youtu.be/30WcPnvfiKE>

3.2.2.1. Un ejemplo de integración ‘online-unplugged’: el curso ‘K-8 Intro to Computer Science’ de Code.org

A modo de síntesis integradora, se finaliza con un sub-epígrafe que detalla el programa-curso trimestral ‘K-8 Intro to Computer Science’²⁴⁵ (Code.org, 2015a) (Figura 3.16), objeto central de la evaluación de programas que abordaremos en el Capítulo 7 de esta tesis, y excelente ejemplo de curso que integra en su estructura tanto sesiones de ‘coding on-line’ como sesiones ‘unplugged’.



Figura 3.16. Código QR de enlace al curso ‘K-8 Intro to Computer Science’ (Code.org, 2015a)

El curso ‘K-8 Intro to Computer Science’ (también denominado ‘Accelerated Intro to CS Course’) está diseñado por la fundación estadounidense sin ánimo de lucro Code.org, cuyo objetivo fundamental es la desmitificación de la programación entre los estudiantes y su introducción en los colegios desde edades tempranas.²⁴⁶

Este curso introduce al estudiante en las ciencias de la computación y en los conceptos fundamentales de la programación informática, desde la perspectiva del pensamiento computacional. Está diseñado específicamente para el nivel ‘middle school’ de los Estados Unidos, es decir, para los grados 6º, 7º y 8º de su sistema educativo (equivalentes respectivamente a nuestro 6º Primaria, 1º y 2º de la ESO); con un tiempo estimado de aprendizaje de entre 15 y 25 horas. Se puede acceder al mismo ‘on-line’ y de manera gratuita a través de la URL: <https://studio.code.org/s/20-hour>

A continuación, se aporta la traducción de algunos de los aspectos psicopedagógicos más relevantes del curso²⁴⁷:

a) ¿Qué es ‘K-8 Intro to Computer Science’?

‘K-8 Intro to Computer Science’ es un curso gratuito que tiene como objetivo desmitificar las ciencias de la computación y mostrar a los estudiantes K-8 que su aprendizaje puede ser divertido, colaborativo y creativo. El curso está diseñado con el fin de motivar a estudiantes y educadores para proseguir en el futuro con el aprendizaje de las ciencias de la computación, mejorando así las relaciones entre la escuela y el mundo real, progresivamente más tecnológico y digital; promoviendo las conexiones entre educación y la vida cotidiana, repleta de objetos programables.

Este curso enseñará a los estudiantes acerca de las ciencias de la computación, tratando de desarrollar su pensamiento computacional y sus habilidades básicas de programación.

²⁴⁵ Acceso a la portada-índice del curso en <https://studio.code.org/s/20-hour>

²⁴⁶ Una breve visión de los fines y resultados de Code.org disponible en: <http://code.org/files/Code.orgOverview.pdf>

²⁴⁷ Extraído y traducido de <https://code.org/educate/20hr>

Por último, con este curso los educadores tendrán la oportunidad de fomentar un ambiente de aprendizaje colaborativo que haga hincapié en la toma de riesgos. Se enseña que “*el éxito no viene en el primer intento, al igual que los problemas más difíciles del mundo no se resuelven en el primer intento. El desafío es bueno cuando es apoyado por planes y herramientas que conducen al éxito*”. Este curso ayudará a los estudiantes a perseverar en la solución de problemas.

b) ¿Cuáles son los conceptos clave (*‘key concepts’*) impartidos en el curso?

- ¿Qué son las Ciencias de la Computación?
- ¿Qué es un científico de la computación?
- Ser un científico de la computación ético y responsable
- Aplicaciones de las Ciencias de la Computación
- Comprensión básica del código binario
- Cómo depurar errores (*‘debugging’*)
- Cuál es el funcionamiento de Internet
- Conceptos básicos de programación = Conceptos computacionales:
 - o Secuenciación
 - o Bucles (*‘loops’*)
 - o Condicionales
 - o Funciones
 - o Funciones con parámetros
 - o Variables
- Pensamiento computacional
 - o Descomposición
 - o Patrones
 - o Abstracción
 - o Algoritmos

c) ¿Qué itinerario curricular sigue el curso?

El curso sigue un modelo *‘b-learning’* (*‘blended learning’*) de enseñanza de las Ciencias de la Computación, lo cual implica que los estudiantes aprenden a partir de una mezcla de actividades autoguiadas de *‘coding on-line’*; y de actividades sin conexión (*‘unplugged’*), que son dirigidas de manera más tradicional por el profesor sin necesidad de uso del ordenador.

El conjunto de actividades resultante se estructura en 20 etapas: 9 etapas de *‘coding on-line’* y 11 etapas *‘unplugged’*, que se van alternando a lo largo del itinerario curricular del curso. A su vez, las etapas de *‘coding on-line’* están internamente compuestas por una serie de niveles-retos de programación; con un total de 98 niveles-retos existentes a lo largo de las 9 etapas.

Cada etapa está diseñada para ser abordada en una sesión de clase de alrededor de 45-60 minutos de duración. Habitualmente, este curso se integra en el aula ordinaria impartándose 2 etapas semanales (2 sesiones de clase a la semana), con lo cual su completamiento lleva alrededor de las 10 semanas (aproximadamente un trimestre escolar). En la siguiente Figura 3.17 se ilustra el itinerario curricular completo del curso.

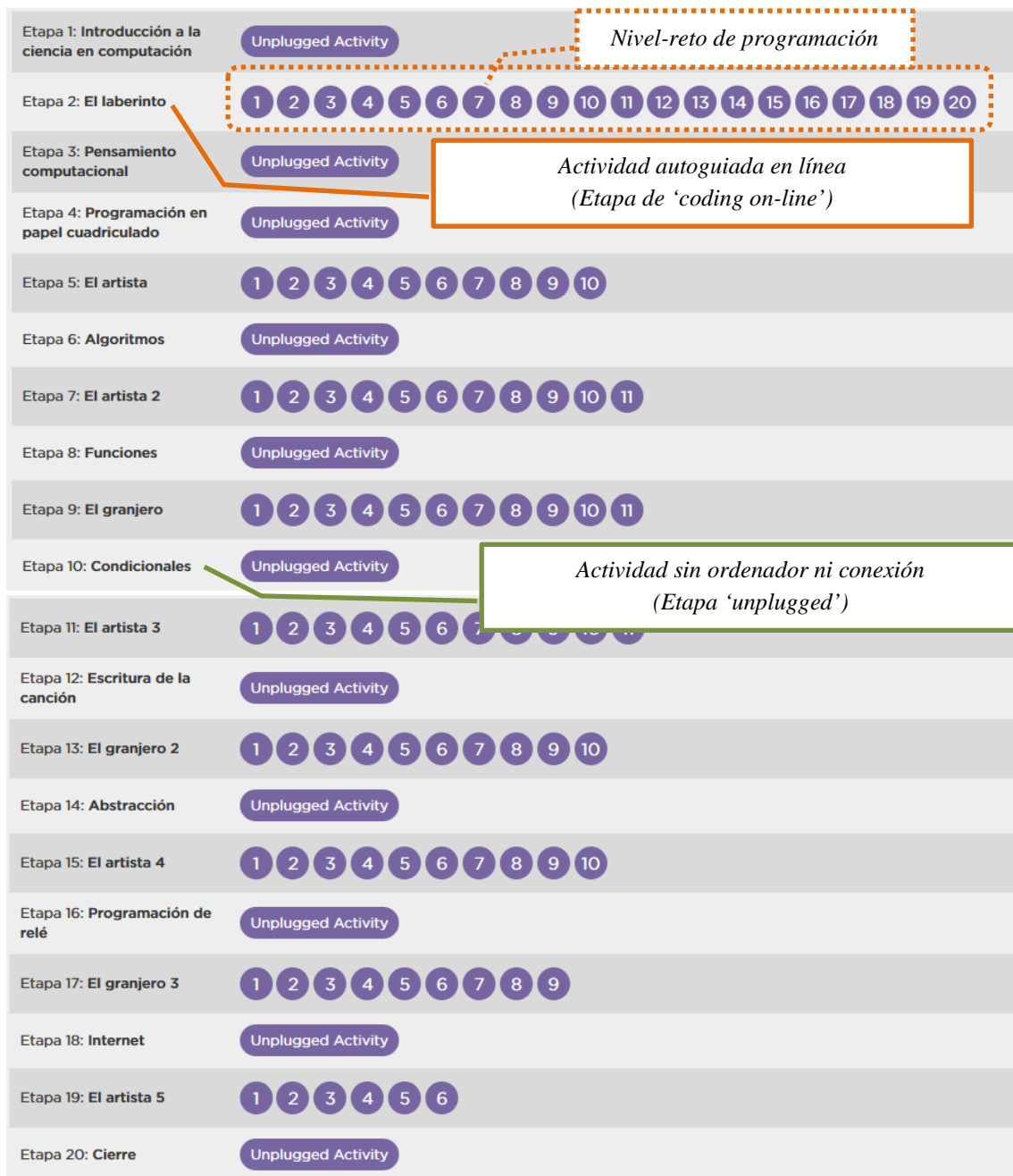


Figura 3.17. Itinerario curricular-formativo del curso 'K-8 Intro to Computer Science' (Code.org, 2015a)

En las Figuras 3.18 y 3.19 se muestran sendos ejemplos de los niveles-retos de programación que componen las etapas de 'coding on-line'. Como puede apreciarse, en la parte izquierda de la pantalla se muestra el problema-reto a resolver, que debe ser solucionado mediante la codificación de un algoritmo; en la parte central se muestran los bloques de código disponibles, que son arrastrados por el aprendiz a la parte derecha para después ejecutar el programa (finalmente, también se muestra el código JavaScript subyacente). Igualmente, tal y como puede adivinarse en la parte inferior izquierda, se ofrecen al estudiante ayudas y pistas adicionales para resolver el reto, en forma de breves videotutoriales.

Etapa 2: Puzzle ●●●●●●●●●●●●●●●●●●●● 15 ●●●●●●●●●●●●●●●●●●●● 20 | Inicia sesión para guardar tu progreso | MAS

Bloques Coloca tus bloques aquí: 4 / 4

avanzar

girar a la izquierda

girar a la derecha

repetir hasta que

hacer

si hay camino a la derecha

hacer

```

while (notFinished()) {
  moveForward();
  if (isPathRight()) {
    turnRight();
  }
}
    
```

Ejecutar programa

Ok, esto es como el último puzzle, pero tienes que recordar cómo usaste el bloque "si" y el bloque "repetir" juntos.

¿Necesitas ayuda? Ver estos videos y sugerencias

Figura 3.18. Nivel 15 de la Etapa 2 del curso 'K-8 Intro to Computer Science'. Disponible en <https://studio.code.org/s/20-hour/stage/2/puzzle/15>

Etapa 7: Puzzle ●●●●●●●●●●●●●●●●●●●● 7 ●●●●●●●●●●●●●●●●●●●● 11 | Inicia sesión para guardar tu progreso | MAS

Bloques Coloca tus bloques aquí: 7 / 7

mover hacia adelante 100 pixeles

girar a la derecha por 90 grados

definir color

definir color color aleatorio

repetir 4 veces

hacer

repetir 10 veces

haz

definir color color aleatorio

repetir 4 veces

haz

mover hacia adelante 20 pixeles

girar a la derecha por 90 grados

mover hacia adelante 20 pixeles

```

for (var count2 = 0; count2 < 10; count2++) {
  penColour(colour_random());
  for (var count = 0; count < 4; count++) {
    moveForward(20);
    turnRight(90);
  }
  moveForward(20);
}
    
```

Ejecutar programa

Aquí está el código para dibujar el cuadrado del puzzle anterior. Puedes repetirlo para dibujar 10 cuadrados adyacentes como una escalera? Pista: sólo necesitas 2 bloques más.

¿Necesitas ayuda? Ver estos videos y sugerencias

Figura 3.19. Nivel 7 de la Etapa 7 del curso 'K-8 Intro to Computer Science'. Disponible en <https://studio.code.org/s/20-hour/stage/7/puzzle/7>

Por otro lado, cada una de las etapas ‘*unplugged*’ se inicia con el visionado de un video introductorio (en casa o en el aula), y se apoya en una ficha didáctica para el profesor (‘*lesson plan*’) que explicita la actividad a desarrollar²⁴⁸: sus objetivos, su desarrollo, el material necesario, etc...

d) ¿Qué estándares de aprendizaje aborda el curso?

El curso ‘*K-8 Intro to Computer Science*’ está alineado con los estándares de aprendizaje²⁴⁹ propuestos por la ‘*Computer Science Teachers Association*’ de los Estados Unidos (CSTA, 2011), que ya revisamos en la **Tabla 3.2** de este capítulo. En concreto, cumple con los siguientes estándares:

- Pensamiento Computacional
 - ✓ *CT.L2-01: Utilizar los pasos básicos en la resolución de problemas a través de algoritmos, para el diseño de soluciones óptimas.*
 - ✓ *CT.L2-07: Representar los datos en una variedad de maneras, incluyendo texto, sonidos, imágenes y números.*
- Prácticas de Computación y Programación
 - ✓ *CPP.L2-05: Implementar soluciones de problemas utilizando un lenguaje de programación, incluyendo: bucles, sentencias condicionales, operadores lógicos, expresiones, variables y funciones.*
- Ordenadores y dispositivos de comunicación:
 - ✓ *CD.L2-01: Reconocer que las computadoras y ordenadores son dispositivos que ejecutan programas.*

e) ¿Qué posibles ventajas metodológicas tiene el curso?

A priori, observamos ciertas ventajas metodológicas que incitan a la implantación del curso en nuestras aulas, y su posterior evaluación:

- Permite el registro gratuito tanto del perfil ‘profesor’ como del perfil ‘alumno’. El profesor puede generar en el interior del curso su propia aula en la que realizar un seguimiento detallado y en tiempo real del avance de sus estudiantes (‘*tracking*’). Se ilustra en la Figura 3.20 (izquierda)
- En el perfil del estudiante queda guardado su avance por el itinerario del curso. Dicho avance está individualizado para cada alumno y ‘gamificado’: el estudiante obtiene ‘*feed-back*’ inmediato de sus errores y recompensas a sus progresos en forma de ‘*badges*’ (trofeos o insignias digitales). Se ilustra en la Figura 3.20 (derecha)

²⁴⁸ Por ejemplo, para la Etapa 3, ‘*unplugged*’, que trata específicamente sobre Pensamiento Computacional [<https://studio.code.org/s/20-hour/stage/3/puzzle/1>], se comienza con el visionado del vídeo <https://youtu.be/injJWiSA0pw>; y se apoya en el ‘*lesson plan*’ disponible para el profesor en <https://studio.code.org/unplugged/unplug2-es-ES.pdf>

²⁴⁹ Información extraída del repositorio documental de Code.org: https://docs.google.com/document/d/16UYAETJBWLQ3e9HyEkTJlqgyAF3bU5n6RHV2GZqv_Y/pub

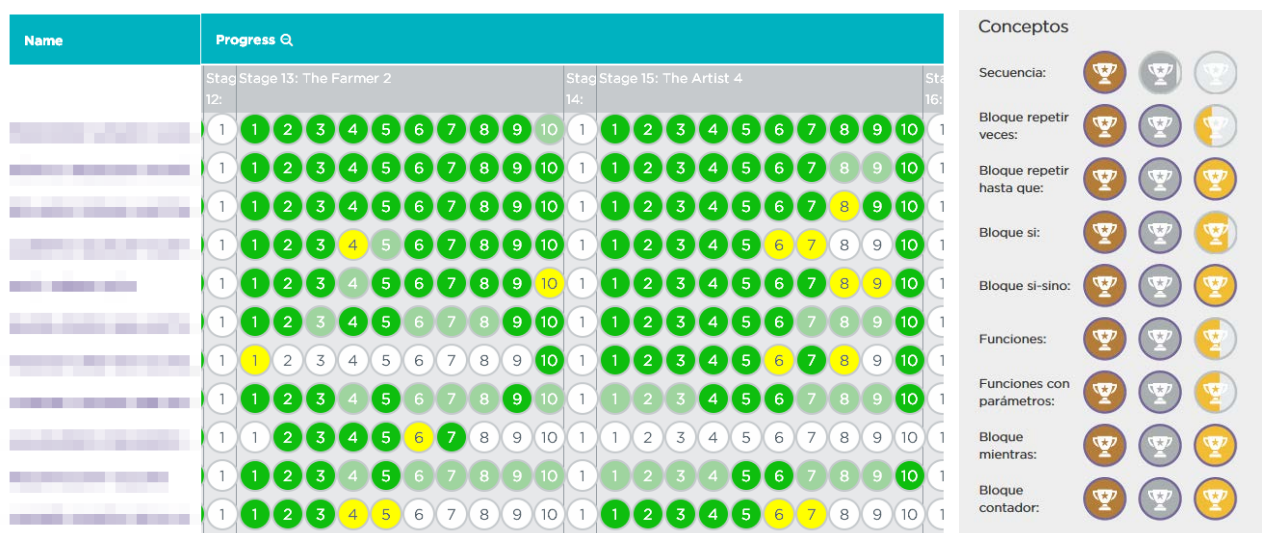


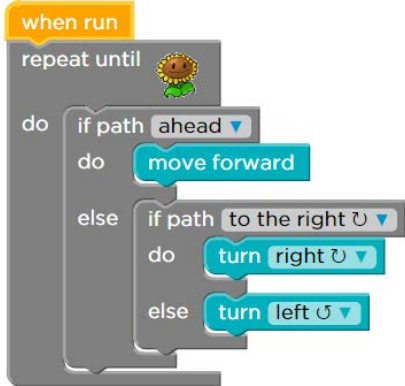
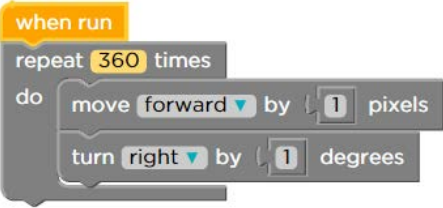
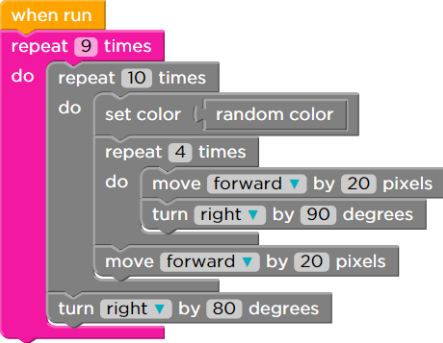
Figura 3.20. Panel de seguimiento (*'tracking'*) del profesor (a la izquierda); y galería de trofeos digitales (*'badges'*) del estudiante (a la derecha) en el curso *'K-8 Intro to Computer Science'*

- El curso está disponible tanto en inglés como en español. Por tanto, permite ser utilizado transversalmente como actividad de desarrollo de la competencia lingüística extranjera (es de reseñar que los lenguajes de programación se apoyan fundamentalmente en la lengua inglesa)
- La metodología *'b-learning'* del curso permite flexibilidad a la hora de organizar las distintas sesiones, y repartir los tiempos de aprendizaje entre el aula y el domicilio del estudiante. Así pues, posibilita utilizar un esquema *'self-paced'* (ritmo de aprendizaje autoguiado por el propio estudiante). Y requiere escasa preparación previa por parte del profesor, dado que todos los materiales y recursos ya se encuentran diseñados.
- El curso es *'browser-based'* (basado en navegador); y es accesible tanto a través de ordenadores PC u ordenadores portátiles, como a través de tabletas y otros dispositivos móviles. Sólo se requiere conexión a Internet para las actividades *'on-line'* (las actividades *'unplugged'* no requieren ordenador)
- El curso permite y promueve el aprendizaje socio-colaborativo por pares a través de la metodología *'pair programming'*²⁵⁰ (dos estudiantes por cada ordenador). Además, el curso permite a los estudiantes compartir sus productos de aprendizaje a través de redes sociales; detalle que puede enriquecer extraordinariamente el potencial educativo del mismo (Colás-Bravo, González-Ramírez, & de Pablos-Pons, 2013)


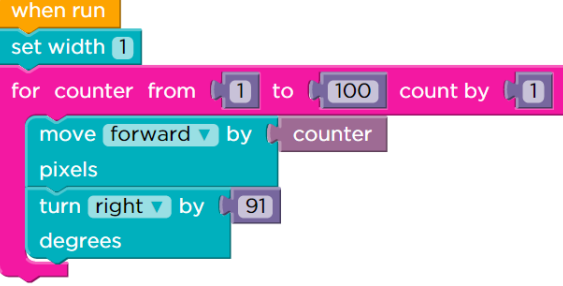
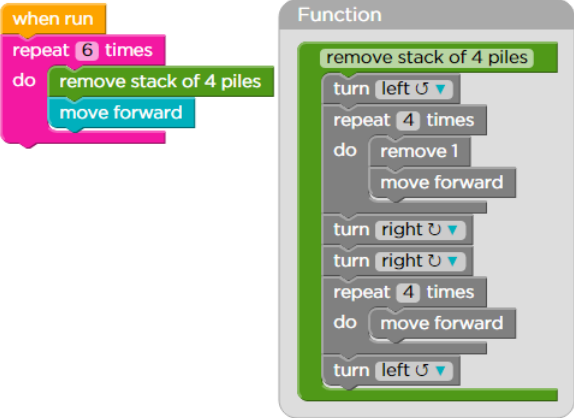
A modo de resumen de los contenidos y recursos que se incluyen en el curso *'K-8 Intro to Computer Science'*, en las siguientes Tablas 3.6 y 3.7 se presentan, de manera sintetizada, las etapas de *'coding on-line'* y *'unplugged'* que lo componen.

²⁵⁰ Más información sobre la metodología *'pair programming'* en <http://www.ncwit.org/resources/pair-programming-box-power-collaborative-learning>

Tabla 3.6. Resumen de las 9 etapas de 'coding on-line' del curso 'K-8 Intro to Computer Science'

Etapa	Nº de niveles-retos que componen la etapa	Tipo de Interfaz utilizada	Conceptos computacionales abordados	Programa necesario para resolver el último o penúltimo nivel-reto de la etapa
'Stage 2: The Maze' ('Etapa 2: El Laberinto')	20 niveles ²⁵¹ Desde: https://studio.code.org/s/20-hour/stage/2/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/2/puzzle/20	Laberintos	Se introduce al estudiante en 2 conceptos computacionales clave: Bucles (tanto 'Repeat times' como 'Repeat until'); y Condicionales (tanto simples 'If', como compuestos 'If/Else')	
'Stage 5: The Artist' ('Etapa 5: El Artista')	10 niveles Desde: https://studio.code.org/s/20-hour/stage/5/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/5/puzzle/10	Lienzo de diseño	Se introduce al alumno en la interfaz de dibujo computacional, practicando con los comandos de: Direcciones básicas (avanzar y girar), aprendiendo cómo secuenciarlos dentro de bucles con el bloque 'Repetir veces' ('Repeat times')	
'Stage 7: The Artist 2' ('Etapa 7: El Artista 2')	11 niveles Desde: https://studio.code.org/s/20-hour/stage/7/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/7/puzzle/11	Lienzo de Diseño	El estudiante continúa trabajando sobre la interfaz de dibujo computacional, practicando con los comandos de direcciones básicas (avanzar y girar), y aprendiendo cómo secuenciarlos dentro de bucles con el bloque 'Repetir veces'. La novedad está en la introducción de bucles anidados	

²⁵¹ Estas 20 pantallas coinciden precisamente con las que Code.org propone en su tutorial de 1 hora para el evento global 'The Hour of Code'

Etapa	Nº de niveles-retos que componen la etapa	Tipo de Interfaz utilizada	Conceptos computacionales abordados	Programa necesario para resolver el último o penúltimo nivel-reto de la etapa
'Stage 9: The Farmer' ('Etapa 9: El Granjero')	11 niveles Desde: https://studio.code.org/s/20-hour/stage/9/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/9/puzzle/11	Parrilla/Cuadrícula ('Grid')	Se introduce al alumno en la interfaz de 'El Granjero': el personaje debe avanzar por un camino dado llenando las zanjas y eliminando los montones que encuentra en su recorrido. Se trabajan los siguientes conceptos computacionales: 'Direcciones básicas', 'Repetir veces', 'Condicional simple (If)' y se introduce por primera vez el bloque 'Mientras (While)'.	
'Stage 11: The Artist 3' ('Etapa 11: El Artista 3')	11 niveles Desde: https://studio.code.org/s/20-hour/stage/11/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/11/puzzle/11	Lienzo de diseño	El estudiante continúa trabajando sobre la interfaz de dibujo computacional. Si en el nivel anterior de 'El Artista', los alumnos aprendieron a secuenciar bucles simples y anidados con el bloque 'Repetir veces'; en esta etapa la novedad está en la introducción de 'Funciones' y del bucle 'Contador' (lo cual les introduce en el concepto de iteración).	
'Stage 13: The Farmer 2' ('Etapa 13: El Granjero 2')	10 niveles Desde: https://studio.code.org/s/20-hour/stage/13/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/13/puzzle/10	Parrilla/Cuadrícula ('Grid')	El alumno continúa profundizando en la interfaz de 'El Granjero'. La novedad que se introduce en esta etapa es que el alumno debe definir y llamar 'Funciones' para resolver las distintas pantallas que se le plantean.	

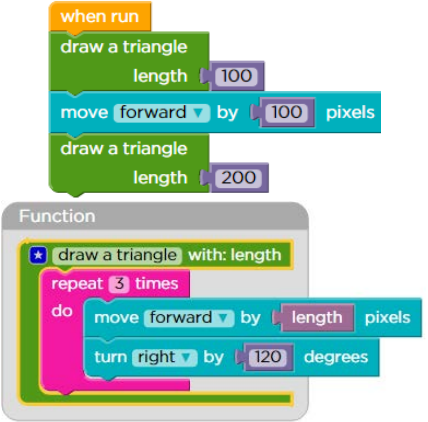
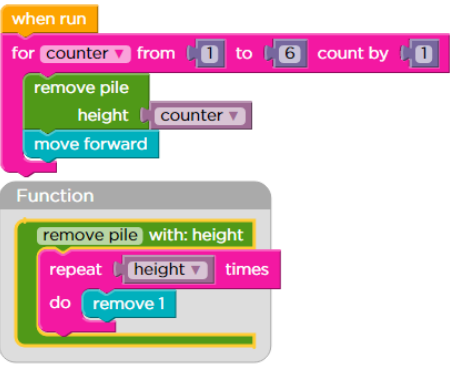
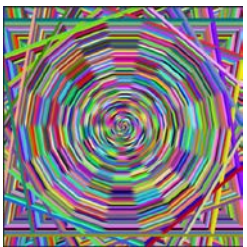
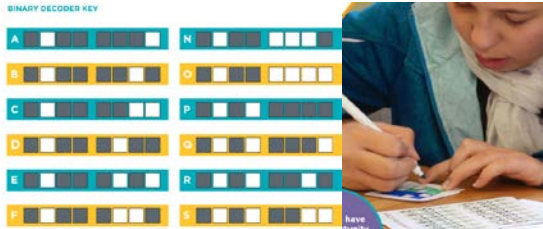
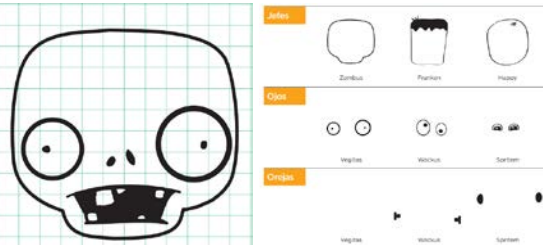
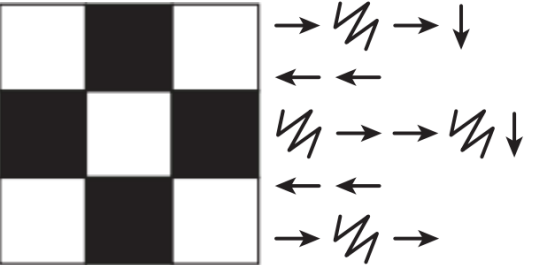
<i>Etapa</i>	<i>Nº de niveles-retos que componen la etapa</i>	<i>Tipo de Interfaz utilizada</i>	<i>Conceptos computacionales abordados</i>	<i>Programa necesario para resolver el último o penúltimo nivel-reto de la etapa</i>
<p><i>'Stage 15: The Artist 4'</i> ('Etapa 15: El Artista 4')</p>	<p>10 niveles</p> <p>Desde: https://studio.code.org/s/20-hour/stage/15/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/15/puzzle/10</p>	<p>Lienzo de diseño</p>	<p>El estudiante continúa trabajando sobre la interfaz de dibujo computacional. La novedad en esta etapa es la introducción de las 'Funciones con parámetros'</p>	
<p><i>'Stage 17: The Farmer 3'</i> ('Etapa 17: El Granjero 3')</p>	<p>9 niveles</p> <p>Desde: https://studio.code.org/s/20-hour/stage/17/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/17/puzzle/9</p>	<p>Parrilla/Cuadrícula ('Grid')</p>	<p>Esta etapa es un repaso sobre lo ya visto en las etapas 9 y 13, añadiendo las 'Funciones con parámetros' y el bucle 'Contador'. Y con la dificultad añadida de que lo que se solicita al estudiante es depurar ('debug') un programa (cognitivamente es más complejo depurar un programa dado, que secuenciarlo)</p>	
<p><i>'Stage 19: The Artist 5'</i> ('Etapa 19: El Artista 5')</p>	<p>6 niveles</p> <p>Desde: https://studio.code.org/s/20-hour/stage/19/puzzle/1 Hasta: https://studio.code.org/s/20-hour/stage/19/puzzle/6</p>	<p>Lienzo de diseño</p>	<p>Se experimenta en la interfaz de dibujo computacional sobre algoritmos complejos previamente dados, en los que se invita al estudiante a probar ligeras modificaciones para ver sus efectos sobre la geometría resultante. El diseño resultante se guarda en su galería y/o se comparte por redes sociales a través de la URL permanente que se genera (ver ejemplo a la derecha)</p>	 <p>https://studio.code.org/c/77294845/edit</p>



Tabla 3.7. Resumen de las 11 etapas ‘unplugged’ del curso ‘K-8 Intro to Computer Science’

Etapa	Vídeo y Ficha didáctica (‘Lesson Plan’) de apoyo	Breve descripción	Ilustración gráfica
<p>‘Stage 1: Introduction to Computer Science’</p> <p>(‘Etapa 1: Introducción a las Ciencias de la Computación’)</p>	<p>Vídeo: https://youtu.be/HsXaVV6FDY</p> <p>‘Lesson Plan’: https://studio.code.org/unplugged/unplug1-es-ES.pdf</p>	<p>El objetivo principal es: ‘Dar a la clase una idea de qué son las Ciencias de la Computación, y cómo éstas pueden serles útiles en su vida’.</p> <p>La actividad final que se propone es que los estudiantes codifiquen en binario su nombre (o sus iniciales) en unas tiras de papel cuadrículadas (que luego pueden llevarse como pulseras), o en unas cuadrículas (que luego se pueden pegar a un imán y colocarse en la nevera)</p>	
<p>‘Stage 3: Computational Thinking’</p> <p>(‘Etapa 3: Pensamiento Computacional’)</p>	<p>Vídeo: https://youtu.be/injJWiSA0pw</p> <p>‘Lesson Plan’: https://studio.code.org/unplugged/unplug2-es-ES.pdf</p>	<p>El objetivo principal es: ‘Introducir el pensamiento computacional como una manera de abordar y formular problemas reales de manera que éstos sean resolubles a través de ordenadores y otros medios digitales’</p> <p>La actividad que se propone consiste básicamente en que, divididos por grupos, analicen varios dibujos de ‘monstruos’, los descompongan y abstraigan sus partes, y escriban un algoritmo para que otro grupo dibuje un nuevo ‘monstruo’</p>	
<p>‘Stage 4: Graph Paper Programming’</p> <p>(‘Etapa 4: Programación sobre hoja cuadrículada’)</p>	<p>Vídeo: https://youtu.be/4e3fi_ls8CA</p> <p>‘Lesson Plan’: https://studio.code.org/unplugged/unplug3-es-ES.pdf</p>	<p>El objetivo principal es: ‘Ayudar a los estudiantes a entender qué significa programar’</p> <p>La actividad que se propone consiste en que los estudiantes codifiquen las instrucciones necesarias para que un compañero pueda dibujar un diseño dado sobre un papel cuadrículado (a base de rellenar cuadraditos)</p>	

Etapa	Vídeo y Ficha didáctica ('Lesson Plan') de apoyo	Breve descripción	Ilustración gráfica
<p>'Stage 6: Algorithms' (Etapa 6: Algoritmos')</p>	<p>Vídeo: https://youtu.be/gW_aPXjgBTc 'Lesson Plan': https://studio.code.org/unplugged/unplug4-es-ES.pdf</p>	<p>El objetivo principal es explicar que: 'Un mismo problema puede ser abordado de maneras distintas, y que hay maneras mejores (más eficientes) que otras'.</p> <p>La actividad se divide en 2 partes. En la primera parte, por parejas, un estudiante hace de 'programador' y otro de 'computadora' (luego alternan). El 'programador' tiene que dar las instrucciones precisas a la 'computadora' para que ejecute un diseño geométrico dado del clásico Tangram sobre un papel cuadriculado. En la segunda parte, ambos investigan las diversas maneras de doblar un folio en 16 cuadrados iguales, y discuten cuál es la solución más eficiente.</p>	 <p>No sólo una computadora puede mal interpretar lo que quisiste indicar, sino que podemos expresar la misma tarea en una computadora de maneras diferentes.</p>
<p>'Stage 8: Functions' (Etapa 8: Funciones')</p>	<p>Vídeo: https://youtu.be/NAVINTHS3LU 'Lesson Plan': https://studio.code.org/unplugged/unplug5-es-ES.pdf</p>	<p>El objetivo principal es: 'Ilustrar cómo las tareas repetitivas pueden agruparse en una pequeña unidad (función) que puede ser 'llamada' varias veces, en lugar de perder espacio y tiempo con una gran cantidad de copias de la misma instrucción'.</p> <p>En la actividad los estudiantes tienen que construir un collar de cuentas-abalorios siguiendo una secuencia dada. Detectan conjuntos de instrucciones que se repiten en la secuencia y las agrupan bajo un nombre-función. Finalmente escriben el programa para construir el collar, llamando a las funciones creadas en el orden correspondiente.</p>	 <p>SKILL 1:</p> <ol style="list-style-type: none"> 1) BEAD 2) KNOT 3) BEAD 4) KNOT 5) SPACER 6) KNOT
<p>'Stage 10: Conditionals' (Etapa 10: Condicionales')</p>	<p>Vídeo: https://youtu.be/yr6g0AWE7yw 'Lesson Plan': https://studio.code.org/unplugged/unplug6-es-ES.pdf</p>	<p>El objetivo principal es: 'Introducir a los estudiantes en los condicionales a través del juego con una/s baraja/s de cartas'.</p> <p>En la actividad se introduce el concepto de 'condicional' y se aplica sobre un juego de cartas. Así, se escribe una pequeña sentencia 'condicional' para añadir o quitar puntos a los equipos en que se divide la clase en función de la carta que se extraiga de la baraja (roja o negra, por ejemplo). También se aprovecha para familiarizar a los estudiantes con la sintaxis de los condicionales en JavaScript (recordándoles que es el código que hay por detrás de los bloques visuales que ellos utilizan en el curso) y en 'pseudocódigo'</p>	 <p>JavaScript</p> <pre>if (tarjeta.color == negro) { equipo.puntos += 1; } else { equipo.puntos -=1; }</pre>

Etapa	Vídeo y Ficha didáctica ('Lesson Plan') de apoyo	Breve descripción	Ilustración gráfica
<p>'Stage 12: Song Writing' (‘Etapa 12: Escritura de canciones’)</p>	<p>Vídeo: https://youtu.be/hx0tBso08aA ‘Lesson Plan’: https://studio.code.org/unplugged/unplug7-es-ES.pdf</p>	<p>El objetivo principal es que: ‘Los estudiantes aprendan cómo definir una función (<i>Function Definition</i>) y cómo llamar a dicha función (<i>Function Call</i>) posteriormente’.</p> <p>En la actividad se introduce el concepto de ‘definir/llamar una función’ haciendo una analogía con cómo se escriben (y luego se cantan) algunas canciones: se define el estribillo (<i>chorus</i>) al inicio, y luego se le va llamando a lo largo de la canción cuando toca repetirlo de nuevo. Aún más, se propone ir un poco más lejos en esta etapa, introduciendo el concepto de ‘funciones con parámetros’ siguiendo con la analogía anterior con canciones cuyo estribillo tiene una parte fija y otra variable (p.e. la conocida ‘<i>Old MacDonald had a farm</i>’²⁵²)</p>	 <p>The image shows a person's hand pointing to a document with lyrics. The lyrics include: CHORUS: Oh, dear! What can the matter be? / Dear, dear! What can the matter be? / Oh, dear! What can the matter be? / Johnny's so long at the fair. SONG: CHORUS: He promised to buy me a tricklet to please me / And then for a smile, oh, he vowed he would tease me / He promised to buy me a bunch of blue ribbons / To tie up my bonnie brown hair. CHORUS: He promised to bring me a basket of posies / A garland of lilies, a gift of red roses.</p>
<p>'Stage 14: Abstraction' (‘Etapa 14: Abstracción’)</p>	<p>Vídeo: https://youtu.be/FGpzpYD9ZVs ‘Lesson Plan’: https://studio.code.org/unplugged/unplug8-es-ES.pdf</p>	<p>El objetivo principal es: ‘Ayudar a los estudiantes a darse cuenta de lo frecuentemente que utilizan la abstracción en su vida cotidiana’.</p> <p>En la actividad se introduce el concepto de ‘abstracción’: eliminar los detalles de una solución, de manera que dicha solución pueda funcionar para una amplia variedad de problemas. Para ilustrar el concepto, se proporciona a los estudiantes una plantilla-estructura común para que cada uno escriba una historia completando los huecos (luego se leen y comparan las distintas historias producidas); y más tarde, a la inversa, se les proporciona 2 historias que difieren en sólo los detalles, para que abstraigan su estructura común.</p>	 <p>The image shows a woman with red hair speaking. To her left is a template for a story with several blank spaces in parentheses: First you take your (), then add a layer of () before you pour on a hearty dose of (). Next, press some () down into the () before covering with a sprinkle of (). That's how I make a ().</p>
<p>'Stage 16: Relay Programming' (‘Etapa 16: Programación por relevos’)</p>	<p>Vídeo: https://youtu.be/FFpvt00Urzg ‘Lesson Plan’: https://studio.code.org/unplugged/unplug9-es-ES.pdf</p>	<p>El objetivo principal es: ‘Enfatizar la importancia de escribir los programas en el orden adecuado, y de revisar-depurar los algoritmos escritos por otras personas en tu equipo’.</p> <p>Esta actividad es una continuación de la Etapa 4. Por equipos, y a modo de carrera de relevos, se va escribiendo progresivamente el algoritmo correspondiente a un diseño gráfico previamente dado. Cada nuevo relevo debe revisar (y depurar en su caso) el algoritmo escrito por sus predecesores, y completarlo un poco más.</p>	 <p>The image shows several students sitting on the floor, working together on a project. They appear to be looking at a large sheet of paper or a screen, possibly a design or algorithm.</p>

²⁵² Se puede consultar dicha canción en <https://youtu.be/5oYKonYBujg>

Etapa	Vídeo y Ficha didáctica ('Lesson Plan') de apoyo	Breve descripción	Ilustración gráfica
'Stage 18: The Internet' (‘Etapa 18: Internet’)	Vídeo: https://youtu.be/Zs26JgnWbTE ‘Lesson Plan’: https://studio.code.org/unplugged/unplug10-es-ES.pdf	El objetivo principal es que: ‘Los estudiantes aprendan cómo funciona Internet, en lo que se refiere a las direcciones URL y páginas web’. Como actividad, se propone representar mediante movimiento el proceso que sigue un mensaje desde que es enviado por un internauta hasta su llegada y descodificación en un servidor; y en función de si el mensaje se transmite a través de Wi-Fi, cable ADSL o fibra óptica.	
'Stage 20: Wrap-up' (‘Etapa 20: Repaso y cierre’)	Vídeo: https://youtu.be/Rbi7keldjZY ‘Lesson Plan’: https://studio.code.org/unplugged/unplug11-es-ES.pdf	El objetivo principal es: ‘Ayudar a la clase a revivir-repasar sus momentos favoritos durante el curso K-8, y orientarles en los próximos pasos que pueden tomar por sí mismos si están interesados en continuar su viaje de aprendizaje de las Ciencias de la Computación’ En esta última actividad se les divide en grupos; a cada grupo se le asigna un conjunto de etapas o conceptos computacionales de los aprendidos durante el curso y, en base a ellos, se les anima a diseñar una nueva actividad-juego ‘unplugged’ que ponga en juego esos conceptos. También se propone facilitar al estudiantes todo el conjunto de recursos en línea para que puedan continuar su aprendizaje por sí mismos más allá del K-8: http://code.org/learn/beyond Se insiste en recordar a los alumnos que han aprendido en 20 lecciones mucho más de lo que la mayoría de los adultos saben sobre ‘coding’ y pensamiento computacional. También puede ser un buen momento para entregar a cada estudiante su certificado de haber realizado el K-8.	

3.3. Evaluación

Tal y como hemos visto en los apartados previos, se reconoce que aún no existe consenso, ni en una definición de ‘pensamiento computacional’ (PC), ni en cómo éste se desarrolla y se puede incorporar a los distintos sistemas educativos; existiendo una enorme variedad y heterogeneidad de intervenciones educativas al respecto (Lye & Koh, 2014). Igualmente, hay un enorme vacío sobre cómo evaluar y medir el PC, hecho que debe ser abordado. Así, se viene señalando desde hace varios años por diversos grupos de expertos (Fincher & Petre, 2004) la necesidad de poner el foco precisamente en “*la evaluación de los aspectos cognitivos vinculados con la programación informática y la comprensión de las computadoras*” (Holmboe, McIver, & George, 2001), esto es, del pensamiento computacional. Y es que:

“Sin la atención suficiente sobre la medida y evaluación del PC, será muy difícil que éste se abra camino exitosamente en cualquier currículum. Aún más, para poder juzgar la efectividad de cualquier currículum que incorpore el PC, es necesario validar previamente instrumentos de medida que permitan a los educadores evaluar en qué grado los estudiantes han desarrollado esa habilidad de pensamiento” (Grover & Pea, 2013a, p. 41)

Este apartado se organiza de la siguiente forma: en primer lugar, revisamos un par de modelos comprensivos de evaluación del pensamiento computacional; en segundo lugar, revisamos instrumentos de medida del pensamiento computacional, haciendo especial hincapié en los dos que utilizaremos para el estudio de validez convergente de nuestro ‘Test de Pensamiento Computacional’: las ‘tareas Bebras’ y ‘Dr. Scratch’.

3.3.1. Modelos de evaluación del pensamiento computacional

Revisamos un par de marcos o modelos que abordan la evaluación del pensamiento computacional desde una perspectiva holística y comprensiva.

a) Modelo de evaluación MIT-Harvard

Dentro del pasado epígrafe ‘3.1.4. Definiciones educativo-curriculares’ del pensamiento computacional, ya vimos con detalle el ‘*computational thinking framework*’ propuesto por Brennan y Resnick (2012), que distingue las siguientes 3 dimensiones del PC: ‘conceptos computacionales’ (*‘computational concepts’*), ‘prácticas computacionales’ (*‘computational practices’*), y ‘perspectivas computacionales’ (*‘computational perspectives’*).

A partir de dicho marco teórico, los autores proponen 3 aproximaciones complementarias para evaluar el desarrollo del PC entre los estudiantes (Brennan & Resnick, 2012), específicamente en el contexto del diseño de objetos digitales interactivos con Scratch:

- **‘Aproximación #1: Análisis de portfolio-proyectos’** (*‘project portfolio analysis’*). Cada miembro de la comunidad ‘*on-line*’ de Scratch tiene una página de perfil que guarda y muestra sus creaciones y proyectos. Dicho conjunto de proyectos puede ser considerado un portfolio, que puede ser sometido a análisis automático de contenido a través de herramientas

como ‘*Scrape – User Analysis*’²⁵³ (Wolz, Hallberg, & Taylor, 2011). Esta herramienta analiza el código interno de los proyectos Scratch, registrando qué tipos de bloques-comandos de programación han sido utilizados por el autor (y con qué frecuencia) en cada uno de sus proyectos. Como resultado del análisis, se genera una representación visual que ilustra de manera resumida e intuitiva el contenido computacional del conjunto de proyectos de un sujeto (Figura 3.21): cada columna representa un proyecto Scratch y todos los bloques-comandos de programación que lo forman; cada fila representa un tipo específico de bloque de programación (el color más oscuro en una celda indica un uso muy frecuente de dicho bloque de programación a lo largo de ese proyecto); la columna final muestra los bloques de programación que nunca han sido utilizados por el sujeto.

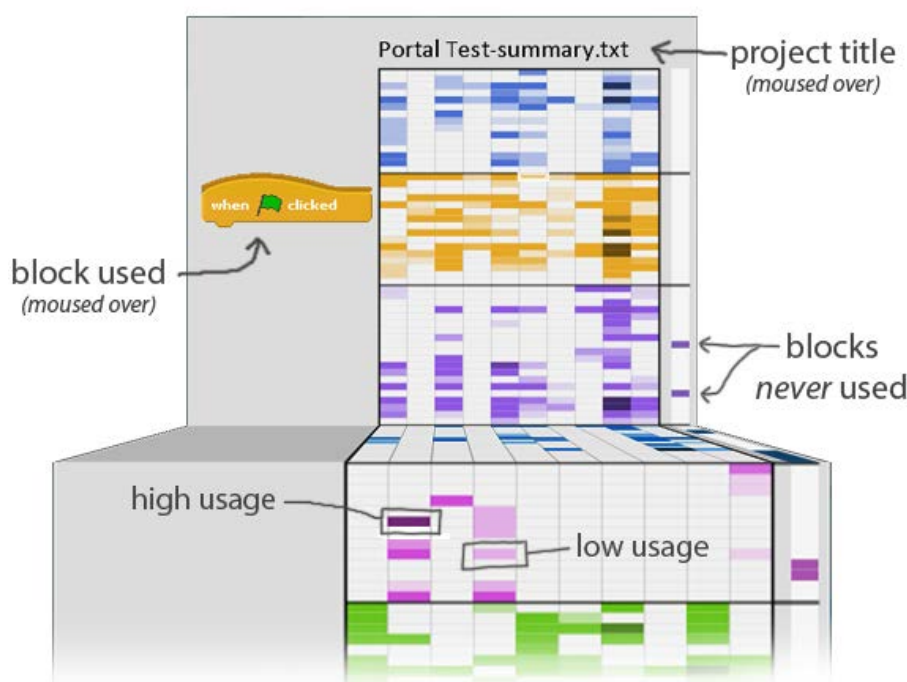


Figura 3.21. Visualización resultante del análisis de un conjunto de proyectos Scratch a través de la herramienta ‘*Scrape*’

Como fortaleza de esta aproximación a la evaluación del PC podemos destacar que, en la medida que los bloques de programación de Scratch se encuentran alineados con los ‘conceptos computacionales’ que propone el modelo MIT-Harvard, entonces este tipo de análisis provee información directa de los ‘conceptos computacionales’ que maneja un determinado sujeto. Especialmente interesante es el uso de este tipo de herramientas de análisis automático desde una perspectiva de evaluación formativa²⁵⁴, que sirva de retroalimentación para que el sujeto progrese en la programación de proyectos Scratch más complejos, y no tanto su uso meramente sumativo para ofrecer una calificación final a un portfolio.

Como limitación de esta aproximación, se puede señalar que está completamente orientada al análisis del producto, dejando de lado aspectos fundamentales del proceso seguido por el

²⁵³ <http://happyanalyzing.com/downloads/userAnalysis/>

²⁵⁴ Algo más adelante, veremos que este plus formativo es el que aporta una herramienta como ‘*Dr. Scratch*’ respecto a ‘*Scrape*’

estudiante al programar (p.e. ¿Qué tipo de ‘prácticas computacionales’ ha seguido el sujeto para producir el proyecto? ¿Ha remezclado o depurado el proyecto de una tercera persona, o lo ha hecho ‘desde cero’?... los bloques de programación que ha utilizado para producir su proyecto, ¿los usa porque los entiende? ¿Los usa porque los ha visto en proyectos de otros compañeros? ¿Los usa porque ha sido expuesto a programas-modelo similares en clase?). Otra limitación a señalar es que las herramientas de análisis automático como ‘*Scrape*’ sólo se focalizan en aspectos cuantitativos (presencia y frecuencia de aparición de cada uno de los tipos de bloques de programación), dejando de lado aspectos cualitativos de los proyectos (p.e. su calidad estética, su originalidad, su tipología...)

- **‘Aproximación #2: Entrevistas basadas en artefactos-objetos’** (*‘artifact-based interviews’*). Desde esta aproximación, se pide a los sujetos que seleccionen un par de entre sus proyectos Scratch, y sobre los mismos se les realiza una entrevista semiestructurada (véase Brennan & Resnick, 2012, p. 15-16), tratando de evaluar no sólo los ‘conceptos computacionales’ utilizados sino igualmente las ‘prácticas computacionales’ seguidas. Así, para cada proyecto: se comienza preguntando al sujeto sobre las motivaciones que originaron la programación del mismo; se continúa ejecutando el proyecto en pantalla para ver cómo funciona; entonces se pregunta al sujeto cómo fue evolucionando el proyecto a lo largo del proceso de programación (qué problemas encontró, cómo los resolvió, etc...); y finalmente se debate y reflexiona con el sujeto acerca del producto conseguido (de qué parte está más orgulloso, cómo se podría mejorar, etc...).

Esta aproximación presenta como fortaleza el superar algunas de las limitaciones de la aproximación anterior. Por ejemplo, con las entrevistas se hace evidente que el hecho de que aparezca un bloque de programación en un proyecto no es necesariamente un indicador de que el sujeto tenga una comprensión profunda sobre el ‘concepto computacional’ correspondiente, ni que lo sepa transferir a un contexto de programación diferente. Adicionalmente, este enfoque permite profundizar sobre las ‘prácticas computacionales’ seguidas por el sujeto mientras fue desarrollando el proyecto.

Como limitación, las ‘entrevistas basadas en artefactos-objetos’ son de aplicación individual y requieren de mucho tiempo para el evaluador. Además, tienen ciertos riesgos de sesgo como, por ejemplo, la discrecionalidad del sujeto a la hora de seleccionar los proyectos que se someterán a la entrevista, o las distorsiones de la memoria a la hora de relatar el proceso de programación seguido.

- **‘Aproximación #3: Escenarios-situaciones de diseño’** (*‘design scenarios’*). En esta tercera aproximación son los evaluadores los que proporcionan una serie de proyectos Scratch a los estudiantes, a modo de escenarios-situaciones iniciales que funcionan como puntos de partida para el proceso de diseño y programación. Concretamente, se les ofrece una serie de proyectos, de creciente complejidad, y se les pide: a) explicar qué es lo que hace el código del proyecto; b) describir cómo el proyecto podría ser ampliado; c) arreglar un ‘*bug*’ (error de código) del proyecto; y d) remezclar el proyecto, añadiéndole alguna nueva característica.

La fortaleza fundamental de esta aproximación reside en que evalúa *in situ* la fluidez del sujeto tanto en el manejo de ‘conceptos computacionales’ como de ‘prácticas computacionales’ (a diferencia de la aproximación anterior, basada en un relato de memoria). Como limitación, comparte con la aproximación anterior el alto consumo de tiempo exigido. Además, se corre el riesgo de que los proyectos, externamente elegidos, no conecten con los intereses del sujeto, subestimando sus capacidades computacionales.

En la Tabla 3.8 se sintetiza la conexión entre las 3 aproximaciones de evaluación con el ‘*Computational Thinking Framework*’ propuesto por los autores.

Tabla 3.8. Conexión entre las 3 aproximaciones de evaluación del PC y el ‘*Computational Thinking Framework*’ (Traducido de Brennan & Resnick, 2012)

	<i>Conceptos computacionales</i>	<i>Prácticas computacionales</i>	<i>Perspectivas computacionales</i>
Aproximación #1: <i>Análisis de portfolio-proyectos</i>	La presencia de bloques como indicador de la existencia de conceptos (sin matices)	No se cubre	No se cubre (se podría cubrir con la extensión del análisis a los comentarios que generan los proyectos en la plataforma)
Aproximación #2: <i>Entrevistas basadas en artefactos-objetos</i>	Se evalúa comprensión conceptual con matices, pero a partir de un conjunto limitado de proyectos	Se cubre, basado en experiencias propias y auténticas de diseño-programación; pero sujetas a las limitaciones de la memoria	Quizás se cubre, pero es difícil preguntarlo directamente
Aproximación #3: <i>Escenarios-situaciones de diseño</i>	Se evalúa comprensión conceptual con matices (y amplio rango de conceptos), pero a partir de un conjunto de proyectos seleccionados externamente	Se cubre, basado en una situación novedosa y en tiempo real, pero con proyectos externamente seleccionados	Quizás se cubre, pero es difícil observarlo directamente

Dado que ninguna de las 3 aproximaciones cubre suficientemente las 3 dimensiones del PC, se recomienda una combinación de ellas a la hora de evaluar el pensamiento computacional de los estudiantes.

b) Modelo Multinivel Bender-Urrea

Desde un punto de vista algo distinto, Walter Bender y Claudia Urrea proponen un modelo de evaluación multinivel, específicamente diseñado para valorar y hacer visible el impacto de las recientes reformas curriculares que están introduciendo la programación y el pensamiento computacional en los sistemas educativos (Bender & Urrea, 2015; Urrea & Bender, 2012). Para ello, proponen una serie de recomendaciones y pautas de innovación evaluativa a tres niveles:

- **Nivel micro** (*‘micro level’*): es el nivel de los estudiantes tomados individualmente. Se propone fundamentalmente que los estudiantes desarrollen portfolios digitales que incluyan sus proyectos de programación; y que éstos les sirvan de ayuda (también a profesores y padres) para tomar conciencia de su aprendizaje a través de la documentación sistemática de su trabajo, y de la reflexión subsiguiente sobre el mismo.
- **Nivel meso** (*‘mezzo level’*): es el nivel de aula o de centro. Se trata de ayudar a los profesores a que analicen y comprendan sistemáticamente el impacto del currículo de programación

introducido. Para ello se les proporcionan herramientas que analizan automáticamente datos derivados de las actividades de programación en línea de los estudiantes. Al estilo de lo visto anteriormente con la herramienta ‘*Scrape*’ para proyectos Scratch, estos autores desarrollan una analítica automatizada para proyectos realizados con una versión visual por bloques del lenguaje Logo (Bender & Urrea, 2015)

- **Nivel macro (‘macro level’):** es el nivel de sistema educativo nacional y global. Se propone crear repositorios en línea, que incluyan los objetos digitales creados por los estudiantes a través de distintos entornos de programación; para conectarlos entre sí (tanto objetos como estudiantes), generando un acervo cultural-digital propio.

3.3.2. Instrumentos de medida del pensamiento computacional

Contar con instrumentos validados de medida es algo necesario y valioso en cualquier área. Sin embargo, y por el momento, todavía existe un gran vacío de instrumentos de medida relativos al ‘pensamiento computacional’ (PC) que hayan pasado por un proceso completo de validación psicométrica (Mühling, Ruf, & Hubwieser, 2015). Tal y como afirman Buffum *et al.* (2015):

“Desarrollar medidas validadas del aprendizaje de los estudiantes es un área urgente de desarrollo para el campo, relativamente joven, de la educación en Ciencias de la Computación. Ello permitirá avanzar hacia una maduración de la disciplina, y equipararse al estado de otras como la Física y las Matemáticas, que ya desde hace tiempo han establecido sus pruebas de medida estandarizadas” (Buffum *et al.*, 2015, p. 622)

Tanto los profesores como los investigadores podrían beneficiarse de la existencia de un conjunto de instrumentos de medida validados del PC. Los profesores pueden utilizarlos para evaluar y ofrecer ‘*feed-back*’ a los estudiantes con una medida externa e independiente a los exámenes, habitualmente diseñados por los propios profesores y que, por tanto, pueden estar afectados de sesgos personales e incontrolables. Por su parte, los investigadores pueden utilizar estos instrumentos para evaluar la efectividad de distintas intervenciones, metodologías, e incluso currículums en el área de la educación en Ciencias de la Computación (Mühling, Ruf, & Hubwieser, 2015)

Dada la relativa novedad del ‘pensamiento computacional’ como constructo psicológico, y la falta aún de consenso sobre su definición, no es de extrañar que nos encontremos todavía en un estado incipiente en lo que se refiere a su medida a través de instrumentos validados y estandarizados (‘*tests*’)²⁵⁵. A este respecto, nos parece ilustrativo comparar el número de resultados que *Google Scholar* devuelve para el *input* de búsqueda “*computational thinking test*”, en comparación con otros instrumentos que, a priori, parecen afines: “*logical thinking test*”, “*critical thinking test*”, “*mathematical thinking test*”, y “*problem solving test*”. Tal y como puede apreciarse en la Tabla 3.9, el *input* “*computational thinking test*” es, con diferencia, el que menos resultados devuelve: sólo 20 resultados, y todos ellos del año 2013 hacia delante; todo lo cual parece un claro indicador del momento inicial en el que se encuentra la investigación al respecto, y el amplio camino por recorrer para equipararse con la medición psicométrica de otros constructos afines.

²⁵⁵ Sí que se encuentran algunos antecedentes históricos en los años 60, y en el contexto de la selección de personal para las nacientes empresas informáticas del momento como IBM, de ‘baterías de aptitudes de programación’ (‘*Aptitude Assessment Battery Programming*’), hoy día ya en desuso y descatalogadas. (véase, p.e., Wolfe, 1968)

Tabla 3.9. Comparativa de resultados de búsqueda entre el input “*computational thinking test*” y otros afines

<i>Input de búsqueda</i>	<i>Nº de resultados devueltos por Google Scholar (diciembre de 2015)</i>
“ <i>computational thinking test</i> ” ²⁵⁶	20 resultados ²⁵⁷
“ <i>mathematical thinking test</i> ”	57 resultados
“ <i>logical thinking test</i> ”	174 resultados
“ <i>critical thinking test</i> ”	2.890 resultados
“ <i>problem solving test</i> ”	4.290 resultados

Si revisamos este *corpus* de investigación reciente que aborda la problemática de la medida y evaluación del PC, podemos diferenciar inicialmente entre estudios realizados con población universitaria (habitualmente estudiantes de primeros cursos de carreras relacionadas con la Informática); y estudios realizados con población pre-universitaria (habitualmente estudiantes de finales de primaria, secundaria y bachillerato)

a) Estudios con población universitaria

Por ejemplo, en Sudáfrica, se ha desarrollado el ‘*Computational Thinking Framework (CT-F)*’, un marco de trabajo que sirve de referencia para diseñar y evaluar materiales y recursos de aprendizaje del PC (como, por ejemplo, ‘*Light-Bot*’²⁵⁸), y evaluar a los estudiantes (Gouws, Bradshaw & Wentworth, 2013a). Este grupo de investigación ha estudiado igualmente el rendimiento de estudiantes de primer año de universidad en su ‘*Test for Computational Thinking*’ (Gouws, Bradshaw & Wentworth, 2013b; Marais, & Bradshaw, 2015): las preguntas usadas en esta evaluación fueron seleccionadas de entre el banco de ítems liberados por la ‘*Computer Olympiad Talent Search*’²⁵⁹ una olimpiada informática cuyo objetivo es detectar y premiar el talento computacional, y que pretende orientar a los estudiantes sudafricanos con más aptitudes en PC hacia carreras técnicas. En la Figura 3.22 se muestra un ejemplo de ítem utilizado (“*Una rana se encuentra en el fondo de un pozo de 19 metros de profundidad. La rana salta 4 metros hacia arriba del pozo, pero entonces necesita 1 hora para recuperarse. Durante esa hora, la rana se desliza hacia abajo 2 metros. ¿Cuántas horas llevará a la rana salir del pozo?*”). Los autores sólo presentan resultados de rendimiento en el test de sus estudiantes, pero no un estudio de las propiedades psicométricas del mismo.

5.0.3 *A frog is at the bottom of a well which is 19 m deep. The frog jumps 4 m up the side of the well, but then needs an hour to recuperate. During this hour the frog slides back 2 m. How many hours will it take the frog to get out of the well?*

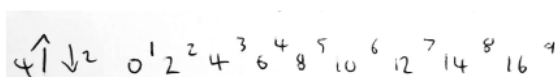


Figura 3.22. Ítem utilizado en el ‘*Test for Computational Thinking*’ (Gouws, Bradshaw & Wentworth, 2013b), perteneciente a la categoría de ‘Patrones y Algoritmos’ (la respuesta correcta es 8 horas)

²⁵⁶ <https://scholar.google.es/scholar?hl=es&q=%22computational+thinking+test%22&btnG=&lr=>

²⁵⁷ Todos ellos desde el año 2013 hacia adelante. Además, de entre los 20 resultados, ya aparecen 2 trabajos en los cuales ha tenido el privilegio de participar este doctorando (Moreno-León, Robles, & Román-González, 2015; Román-González, 2015a)

²⁵⁸ <https://lightbot.com/>

²⁵⁹ <http://www.olympiad.org.za/talent-search/>

En la Universidad de Kentucky, se han examinado las conexiones entre el PC y el pensamiento crítico en estudiantes universitarios, desarrollando su propio instrumento de medida del PC consistente en una combinación de preguntas cortas y de elección múltiple que trata de evaluar a los estudiantes en diversos conceptos computacionales; este instrumento aún no está validado ni resulta suficientemente fiable (Walden, Doyle, Garns & Hart, 2013).

Otra aproximación, algo distinta, proviene del grupo de investigación de la Universidad de Nebraska (Miller *et al.*, 2013, 2014; Shell *et al.*, 2014), que se propone como objetivo promover el desarrollo del pensamiento computacional de sus estudiantes de primer año universitario en el Grado de Informática (curso denominado habitualmente ‘CS1’ o ‘Computer Science I’). Su línea de investigación se basa en un enfoque innovador que integra el ‘pensamiento computacional’ y el ‘pensamiento creativo’ (*creative thinking*) en el marco de referencia de la Teoría de la Generatividad de Epstein (Epstein, Schmidt, & Warfel, 2008). Para ello, diseñan e implementan un conjunto de ejercicios de pensamiento creativo alineados-vinculados con conceptos del pensamiento computacional: los ‘Computational Creativity Exercises’ (CCE); con la premisa de que los estudiantes puedan aprovechar y transferir su ejercitación en habilidades de pensamiento creativo con dichos ejercicios para, posteriormente, ‘desbloquear’ su comprensión del pensamiento computacional. Los CCE tampoco han sido sometidos a un proceso de validación psicométrica.

También en el nivel CS1, denominación del primer año universitario en el Grado de Informática de los Estados Unidos, destacan los trabajos de Tew y Guzdial (2010, 2011). Estos autores, provenientes del Instituto Tecnológico de Georgia (*Georgia Tech*), desarrollan y validan el ‘Foundational CS1 Assessment Instrument’ (FCS1): un instrumento cuyo objetivo es evaluar la adquisición que los estudiantes CS1 hacen de los conceptos fundamentales de la computación, y que tiene la ventaja de ser independiente de cualquier pedagogía particular o lenguaje informático concreto que se utilice para dicho curso universitario CS1. Los conceptos computacionales abordados por el instrumento FCS1 son: operadores lógicos, condicionales simples y compuestos, bucles definidos e indefinidos, matrices, funciones simples y con parámetros, recursividad, variables, y orientación a objetos.

b) Estudios con población pre-universitaria

Desde la Universidad de California, se propone el instrumento ‘Fairy Assessment in Alice’, (Werner, Denner, Campe & Kawamoto, 2012), que se basa tanto en valorar objetos creados-programados por los estudiantes como en analizar sus respuestas a reactivos de programación prediseñados. En ambos casos, para tratar de medir la comprensión y uso de los estudiantes ‘middle school’ (6º Primaria a 2º ESO) de la abstracción, la lógica condicional, el pensamiento algorítmico y otros conceptos del PC que se aplican para resolver problemas. Sin embargo, es un instrumento diseñado *ad hoc* para ser utilizado en el contexto de aprendizaje del entorno de programación Alice²⁶⁰ (Graczyńska, 2010), y que no se ha sometido a un proceso de validación psicométrica.

El grupo de investigación de la Clemson University (Carolina del Sur), aporta una perspectiva complementaria (Daily, Leonard, Jörg, Babu, & Gundersen, 2014; Leonard *et al.*, 2015). Proponen una aproximación kinestésica al aprendizaje (*embodied learning*) y evaluación del pensamiento

²⁶⁰ <http://www.alice.org/index.php>

computacional con estudiantes de *'upper elementary and middle school'* (específicamente 5° y 6° de Primaria); para ello alternan actividades de programación de secuencias de movimiento (coreografías) en el entorno Alice, con la representación de esas mismas secuencias en un entorno físico-kinestésico. El instrumento de evaluación combina igualmente ambos entornos (Figura 3.23), pero no se reportan sus propiedades psicométricas. Los autores sí que informan de las virtudes específicas de esta aproximación kinestésica para atraer a las chicas hacia el aprendizaje de la programación.

Un enfoque complementario en la medida del crecimiento del PC a lo largo de la escuela secundaria ha sido estudiar el desarrollo entre los estudiantes del uso de vocabulario y terminología específica relacionados con la Informática (Grover, 2011)

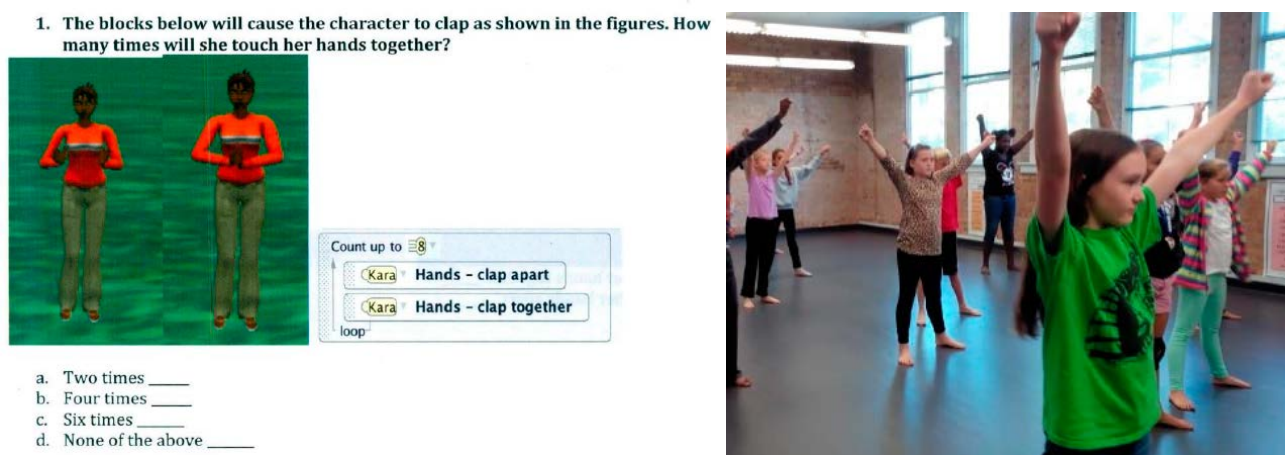


Figura 3.23. Aproximación kinestésica al aprendizaje (*'embodied learning'*) del PC. A la izquierda, un ejemplo de ítem utilizado en el instrumento diseñado por Leonard *et al.* (2015)

En Taiwán, recientemente se ha realizado una evaluación del PC a gran escala entre alumnos de secundaria, bachillerato y formación profesional, usando para ello una selección de 15 tareas-problemas extraídos del *'International Bebras Contest'*²⁶¹ (Lee, Lin & Lin, 2014). En su trabajo, los autores informan sobre el rendimiento de los estudiantes taiwaneses en las distintas tareas propuestas. Pero no se considera el conjunto de tareas como un instrumento de medida a validar, ni se estudian sus propiedades psicométricas. Retomaremos específicamente el concurso Bebras en el sub-epígrafe 3.3.2.1.

Otra línea de investigación muy interesante con estudiantes *'middle school'* (6° Primaria, 1° y 2° ESO) es la que aporta el grupo de la Universidad de Colorado. Trabajan con los estudiantes en el entorno de programación de videojuegos *'AgentSheets'*²⁶². En un primer grupo de estudios (Koh, Basawapatna, Bennett, & Repenning, 2010), identifican una serie de *'patrones de pensamiento computacional'* (*'Computational Thinking Patterns'* o CTP) que los jóvenes programadores abstraen y desarrollan durante el diseño de sus videojuegos: por ejemplo, el patrón *'colisión'* (*'collision'*) que es programado para que dos personajes del videojuego choquen; o el patrón *'transporte'*

²⁶¹ El *'International Bebras Contest'* es un concurso a nivel internacional sobre fluidez informática y computacional en niños, adolescentes y jóvenes. Tiene su origen en Lituania (<http://www.bebbras.org/>), aunque ya cuenta con filiales por toda Europa como, por ejemplo, la destacada de Reino Unido (<http://www.beaver-comp.org.uk/>).

²⁶² <http://www.agentsheets.com/>

(‘*transportation*’) que se pone en marcha para determinar que un personaje del juego lleve o arrastre algún objeto de un lugar a otro. En este contexto, diseñan el ‘*Computational Thinking Patterns Graph*’, una herramienta automática que analiza los videojuegos programados por los estudiantes, y que representa de manera gráfica en qué medida cada videojuego ha puesto en juego los diferentes ‘patrones de pensamiento computacional’ en comparación con un videojuego modelo (Figura 3.24)

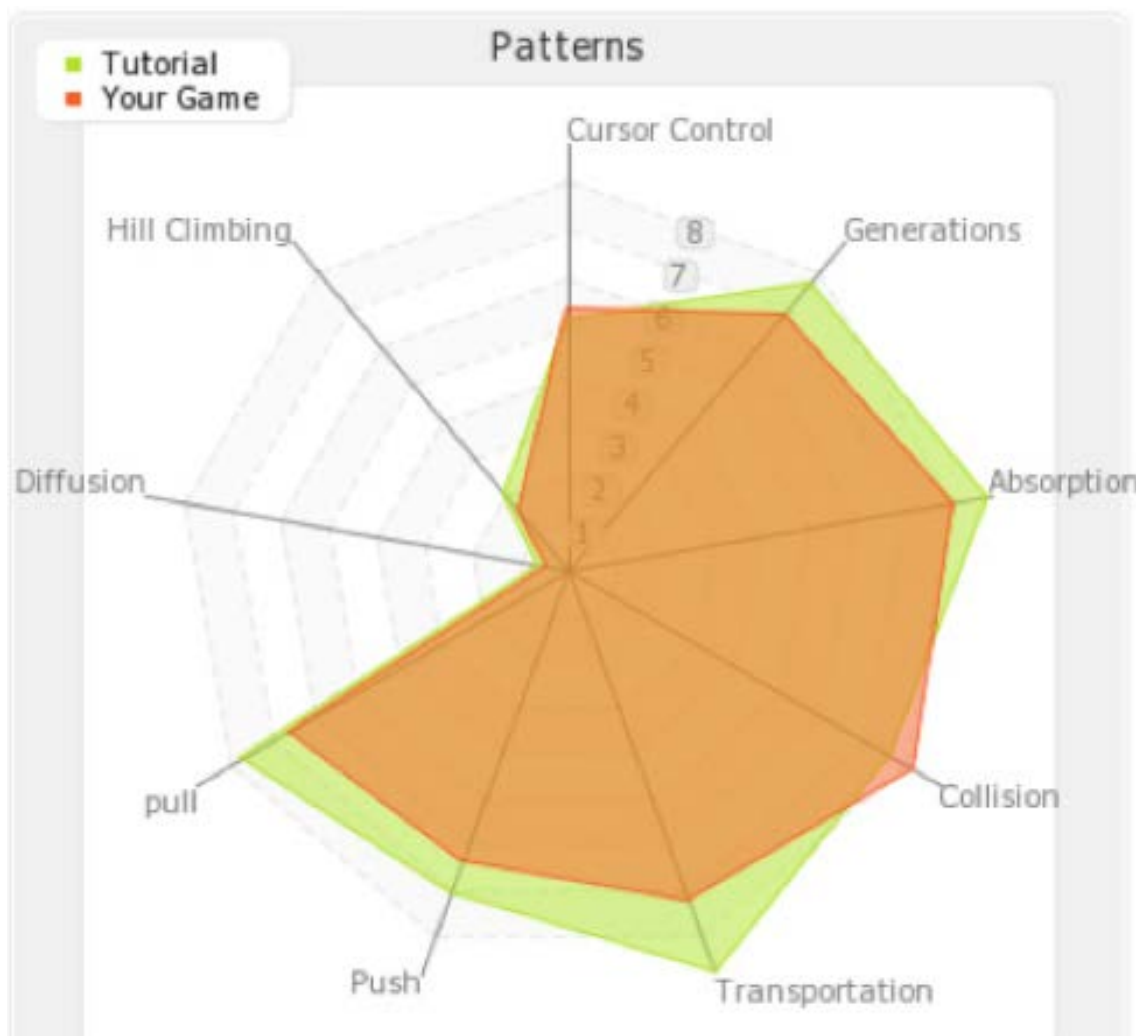


Figura 3.24. ‘*Computational Thinking Patterns Graph*’ desarrollado por la Universidad de Colorado

En un segundo grupo de estudios (Basawapatna, Koh, Repenning, Webb, & Marshall, 2011), los autores tratan de evaluar si los estudiantes son capaces de transferir los ‘patrones de pensamiento computacional’ adquiridos durante la programación de videojuegos, al nuevo contexto de programación de simulaciones científicas (p.e. si son capaces de transferir el patrón ‘colisión’ para programar una simulación del choque entre dos átomos; o si son capaces de transferir el patrón ‘transporte’ para programar una simulación de la circulación de la hemoglobina por el cuerpo humano a través de los glóbulos rojos). Para realizar esta evaluación desarrollan el instrumento de medida ‘*Computational Thinking Pattern Quiz*’, compuesto por una serie de preguntas en forma de vídeos, tras cuyo visionado el estudiante debe reconocer los patrones que han aparecido en el mismo. De este instrumento tampoco se reportan datos de fiabilidad o validez.

Así pues, se constata globalmente una ausencia notable, y preocupante, de instrumentos de medida del ‘pensamiento computacional’ que hayan sido sometidos a un proceso completo y sistemático de validación. Este hecho es remarcado recientemente por Buffum *et al.* (2015), de la Universidad de Carolina del Norte, cuando afirman que, en el campo de la educación en Ciencias de la Computación (*‘Computer Science Education’* o *‘CSE’*), no se han venido aplicando a sus instrumentos de medida y evaluación los criterios psicométricos exigidos de fiabilidad (en qué medida el instrumento es preciso y consistente, internamente y a lo largo de varias aplicaciones) y validez (en qué medida el instrumento mide realmente lo que afirma medir). Para afrontar dicho vacío, este grupo de investigación propone una guía práctica de 7 pasos para el diseño, desarrollo y validación de tests en el área de las Ciencias de la Computación, específicamente para poblaciones de estudiantes *‘middle school’* (6º Primaria, 1º y 2º ESO); y que cumple con los estándares de medición psicométrica fijados por la *American Educational Research Association* (AERA, APA, & NCME, 2014). A continuación, sintetizamos los 7 pasos de la guía (Buffum *et al.*, 2015), que nos han servido de referencia para el diseño y validación de nuestro ‘Test de Pensamiento Computacional’ (TPC) que será presentado en el Capítulo 6:

- i. **Identificar el propósito del test:** es importante enunciar explícitamente el objetivo del test.
- ii. **Definir el constructo de interés:** se debe definir operacionalmente el constructo que va a ser sometido a medición. Ello implica una definición del dominio del constructo, basado en la literatura científica previa.
- iii. **Preparar las especificaciones del test:** se debe decidir el formato del test (p.e. si será un test de elección múltiple o de respuesta abierta). Los autores recomiendan optar por un test de elección múltiple, dada: su mayor facilidad de corrección; su posibilidad de escalamiento y de aplicaciones colectivas a muestras de gran tamaño; su susceptibilidad a ser sometido a técnicas estadísticas de validación; su adecuación para poblaciones *‘middle school’* que pueden mostrarse en ocasiones poco colaborativas si se les solicita evocar una respuesta abierta. Si, finalmente, se opta por la construcción de un test de elección múltiple se dan las siguientes pautas: que cada ítem ofrezca 4 alternativas de respuesta; que la longitud del test sea de, al menos, 20 ítems; y que el tiempo de aplicación tome de referencia la duración de una sesión ordinaria de clase (p.e. si las clases son de 60 minutos, el tiempo máximo de aplicación no debería exceder los 45 minutos para posibilitar una buena gestión de todo el proceso: lectura de instrucciones, periodo de respuesta, recogida de los test).
- iv. **Generar los ítems candidatos del test:** se debe generar un conjunto inicial de ítems candidatos a formar parte del test. Dicho conjunto debe constituir una muestra suficiente y representativa del dominio total del constructo definido anteriormente. Se aconseja que el número de ítems candidatos sea alrededor del doble de la longitud final estimada (es decir, unos 40 ítems candidatos), dado que muchos de estos ítems serán desechados en el próximo paso.
- v. **Realizar una revisión formal de la validez de los ítems generados:** se recomienda que dicha revisión se realice a través de un panel de expertos (al menos 5), compuesto por una mezcla de profesores universitarios y de secundaria en el área de las Ciencias de la Computación. Cada experto, de manera independiente, debe valorar cada ítem candidato de

acuerdo a una serie de criterios, que pueden ser presentados como preguntas dicotómicas ('incluir el ítem / no incluir el ítem') o como escalas tipo Likert. También se recomienda que los expertos puedan expresar sugerencias y comentarios abiertos para la mejora de los ítems.

- vi. **Someter al test a una aplicación piloto con una muestra representativa de la población de interés:** se recomienda que dicha muestra sea de, al menos, 100 estudiantes. Igualmente, se sugiere que todos los sujetos, en esta situación *pretest*, tengan un nivel similar de exposición previa a la programación. Una vez recogidos los datos de la aplicación piloto, se recomiendan los siguientes análisis estadísticos: fiabilidad como consistencia interna, índice de dificultad por ítem, índice de discriminación por ítem, y correlaciones ítem-total (ITC). El test estará bien construido si sus ítems tienen: índices de dificultad que cubran un rango amplio (ítems fáciles, medios, y difíciles); índices de discriminación y correlaciones ítem-total superiores a 0,3.
- vii. **Iterativamente depurar y refinar el test, sometiéndolo a nuevas aplicaciones en cada ciclo:** a partir de los análisis estadísticos anteriores, se propone depurar y refinar el test en sucesivos ciclos o iteraciones (p.e. se propone eliminar los ítems con índices de discriminación y/o correlaciones ítem-total inferiores a 0,3). Cada iteración finaliza con la aplicación de la última versión depurada del test. El proceso iterativo es teóricamente infinito.

En el momento presente, tan sólo encontramos 2 ejemplos de instrumentos que hayan tomado en consideración las recomendaciones de Buffum *et al.* (2015): a) el '*Test for Measuring Basic Programming Abilities*' de la Universidad Técnica de Múnich (Mühling, Ruf, & Hubwieser, 2015); y b) el '*Commutative Assessment*' desarrollado en la Northwestern University (Weintrop & Wilensky, 2015a)

➤ '*Test for Measuring Basic Programming Abilities*' (Mühling, Ruf, & Hubwieser, 2015)

Este test está construido para población escolar de entre 10 y 15 años (Grados 7 a 10 del sistema educativo alemán, equivalentes a nuestro 1º - 4º de la ESO). Su objetivo es medir la habilidad de los estudiantes para ejecutar un programa dado, basado en las denominadas 'estructuras de control de flujo' (*flow control structures*); que se consideran en el corazón del 'pensamiento computacional' para estas edades:

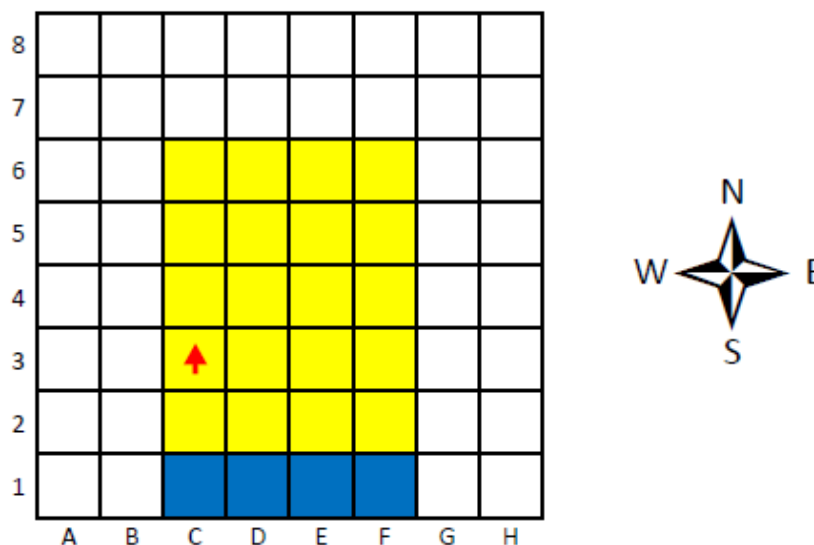
- ✓ Secuenciación: hacer una cosa, siguiendo los pasos que la componen uno tras otro
- ✓ Selección: hacer una cosa, u otra cosa
- ✓ Repetición: hacer una cosa, una y otra vez

Dichas 'estructuras de control de flujo' implican los siguientes conceptos computacionales, que conforman el dominio del constructo medido por el test:

- ✓ Secuencias de operaciones (direcciones básicas)
- ✓ Sentencias condicionales: tanto condicionales simples (sin alternativa: '*if*') como condicionales compuestos (con alternativa: '*if/else*')
- ✓ Bucles con un número fijo de iteraciones: 'repetir *x* veces' (*'repeat x times'*)
- ✓ Bucles con una condición de salida (*'conditional loop'*): por ejemplo, el bucle 'repetir hasta' (*'repeat until'*) o el bucle 'repetir mientras' (*'repeat while'*)
- ✓ Anidamiento de los anteriores conceptos computacionales entre sí.

El formato de diseño de los ítems se pretende que sea independiente de cualquiera de los entornos de aprendizaje de la programación existentes: se trata de una cuadrícula ('grid') en la cual una figura puede ser movida a partir de la combinación de los conceptos computacionales citados anteriormente. Cada ítem contiene dicha cuadrícula, una figura situada en alguna de las celdas de la misma, y un pequeño programa escrito en forma textual de 'pseudocódigo'²⁶³. El sujeto debe responder, para cada ítem, la celda en la que acabará la figura tras ejecutar el programa y su orientación (Figura 3.25):

Alex is standing in his old position C 3 and faces north.



Alex is now executing the following statements:

```
while IsYellow do
  Rightturn
  Step
  Leftturn
  Step
EndWhile
```

Please give the cell in which Alex now resides and the direction in which he faces:

Cell: _____

Direction: _____

Figura 3.25. Ejemplo de ítem del 'Test for Measuring Basic Programming Abilities' (Mühling, Ruf, & Hubwieser, 2015)

El test ha sido diseñado siguiendo la Teoría de Respuesta al Ítem (TRI o, en inglés, IRT) (DiBello & Stout, 2007), que asume la unidimensionalidad del rasgo latente medido y la independencia de los ítems entre sí. En el momento presente, este equipo de investigación se encuentra realizando aplicaciones piloto y depurando los ítems del Test para que se ajusten al modelo Rasch (Rasch, 1993); y, una vez cubierta esa fase, se proponen una re-aplicación en muestras de mayor tamaño para completar los estudios de validación del test definitivo y completo.

²⁶³ 'Pseudocódigo': lenguaje informático textual simplificado, que no sigue específicamente la sintaxis de ninguno de los lenguajes informáticos existentes. Es un lenguaje de programación diseñado para ser leído por el humano, no por la máquina.

➤ ‘Commutative Assessment’ (Weintrop & Wilensky, 2015a)

Este test está diseñado para población ‘high school’ (Grados 9 a 12 del sistema educativo estadounidense, equivalente a nuestro 3º ESO a 2º Bachillerato), y pretende medir la comprensión de los estudiantes de distintos conceptos computacionales, en función de si éstos se presentan a través de programas escritos en lenguajes ‘block-based’ (‘visuales por bloques’) o ‘text-based’ (‘textuales’). Esta comparación se considera muy relevante dado que, en estas edades y en el marco de la asignatura optativa de ‘Computer Science’ de los Estados Unidos, suele abordarse la transición entre la lectoescritura de código con lenguajes ‘visuales por bloques’ a lo propio con lenguajes ‘textuales’, que son los utilizados ya mayoritariamente en el ámbito universitario y profesional.

El test, compuesto por un total de 28 ítems, aborda los siguientes conceptos computacionales, que se consideran centrales en el desarrollo computacional para estas edades: ‘sentencias condicionales’; ‘bucles definidos/determinados’; ‘bucles indefinidos/indeterminados’; ‘funciones simples’; y ‘funciones con parámetros/variables’. Todos los ítems son de elección múltiple con 5 opciones de respuesta. Cada ítem consiste en un pequeño fragmento de código, que aleatoriamente puede aparecer escrito en el lenguaje *Snap!*²⁶⁴ (‘visual por bloques’) o en el lenguaje JavaScript (‘textual’), y el sujeto debe interpretar su funcionamiento. Por ejemplo, en la Figura 3.26 vemos las dos formas en las que puede presentarse una pregunta sobre ‘bucles definidos/determinados’ (“¿Cuántas veces será impresa la palabra ‘here’ cuando se ejecute el programa?” La respuesta correcta es “10 veces”)

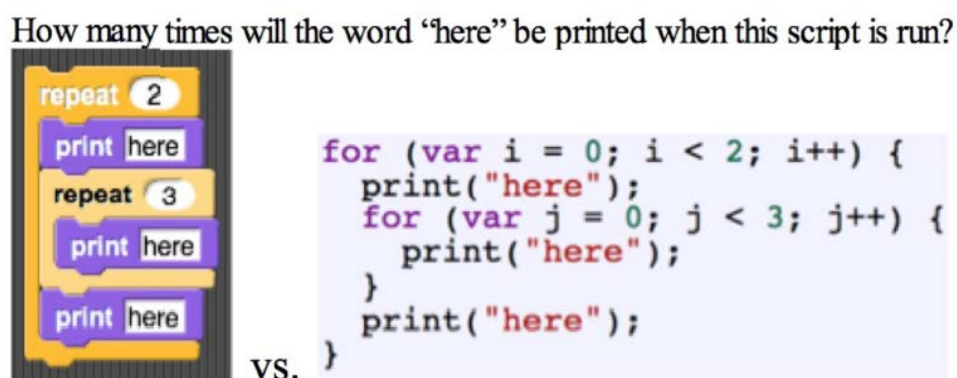


Figura 3.26. Ejemplo de ítem del ‘Commutative Assessment’ (Weintrop & Wilensky, 2015a)

En una primera aplicación piloto del test sobre 90 estudiantes, los autores informan de un rendimiento significativamente superior de los estudiantes en el instrumento cuando los ítems son mostrados en la modalidad ‘visual por bloques’ tanto en los ‘condicionales’, como en los ‘bucles’ y ‘funciones’. Estos resultados son consistentes con la hipótesis de que los lenguajes ‘visuales por bloques’ son más sencillos de interpretar que los lenguajes ‘textuales’; aunque aún dejan abierta la cuestión de si el aprendizaje con los primeros es una buena preparación para el aprendizaje de los segundos.

En un estudio complementario (Weintrop & Wilensky, 2015b), los autores presentan los resultados de las entrevistas realizadas a esos mismos 90 estudiantes sobre su percepción diferencial entre lenguajes ‘visuales por bloques’ y ‘textuales’, tras un curso de introducción a la programación de 10

²⁶⁴ <http://snap.berkeley.edu/>

semanas de duración en el que tuvieron contacto tanto con *Snap!* como con JavaScript. Los estudiantes mayoritariamente contestaron en el sentido de que los lenguajes ‘visuales por bloques’ son más sencillos que los ‘textuales’, por distintos motivos: que los bloques están escritos en un lenguaje mucho más próximo al lenguaje natural, y, por tanto, son más fáciles de leer; que la forma y color de los bloques ofrecen pistas visuales sobre cómo encajan unos con otros y, por tanto, es más sencillo ir componiendo el programa; que las librerías de bloques existentes liberan memoria al programador y facilitan su recuerdo de las funcionalidades disponibles, a diferencia de los lenguajes textuales en las que uno debe recordar sus detalles de sintaxis en todo momento durante la escritura. Por otro lado, los estudiantes también reconocen algunos inconvenientes de los lenguajes ‘visuales por bloques’, a saber, su menor potencia, su limitación a los tipos de bloques pre-existentes y el hecho de que no son los que utilizarán en entornos profesionales de programación.

En cualquier caso, tanto el ‘*Test for Measuring Basic Programming Abilities*’ (Mühling, Ruf, & Hubwieser, 2015) como el ‘*Commutative Assessment*’ (Weintrop & Wilensky, 2015a) se encuentran en pleno proceso de validación. Aún no disponemos de sus versiones públicas y definitivas con las que poder realizar un estudio de validez convergente con nuestro ‘Test de Pensamiento Computacional’ (TPC). Para ello, utilizaremos otras 2 medidas alternativas, que describimos a continuación: tareas extraídas del ‘*Bebras International Contest*’ (‘Concurso Internacional Bebras’); y análisis de proyectos Scratch con la herramienta de análisis automático ‘*Dr. Scratch*’, desarrollada por nuestros colegas de la Universidad Rey Juan Carlos.

3.3.2.1. *El Concurso Internacional Bebras*

El Concurso Internacional Bebras²⁶⁵ tiene su origen en Lituania en 2003. Su objetivo es promover el interés y la excelencia de los escolares de todo el mundo en el área de la Informática en general, y particularmente en lo que al pensamiento computacional se refiere (Cartelli, Dagiene, & Futschek, 2012; Dagiene & Futschek, 2008; Dagiene & Stupuriene, 2014). Desde su nacimiento, se ha ido extendiendo a lo largo de diferentes países de la zona OCDE (‘Organización para la Cooperación y el Desarrollo Económicos’) consolidando su carácter internacional; pasando de los 3.470 participantes lituanos del año 2004, a los 523.319 participantes provenientes de 21 países del año 2013.

La dinámica del concurso es la siguiente: en los meses de primavera, se celebra un simposio internacional al que asisten los respectivos organizadores locales del concurso de cada uno de los países participantes (habitualmente profesores de Ciencias de la Computación de nivel universitario o pre-universitario), y éstos proponen una serie de problemas o tareas (‘*Bebras tasks*’) para que sean incluidas en el concurso del año en curso. Durante el simposio se seleccionan y mejoran las tareas que se incluirán en el banco común o ‘*pool*’ definitivo de problemas para el concurso, alrededor de 400 tareas por año, que se celebra en el mes de noviembre. Posteriormente, cada comité nacional selecciona un subconjunto de tareas de dicho ‘*pool*’, las traduce a su idioma, y las aplica en su país. Al año siguiente se repite el proceso, renovándose el ‘*pool*’ de tareas disponibles.

Existe un conjunto de criterios establecidos que deben seguir las tareas Bebras para ser aceptadas (Dagiene & Futschek, 2008), según mostramos en la siguiente Tabla 3.10.

²⁶⁵ <http://www.bebbras.org/> [‘*Bebras*’ es ‘castor’ en lengua lituana; en inglés es ‘*beaver*’; representa la conducta inteligente, orientada a objetivos, persistente y creativa, que pone en juego el animal al acometer sus construcciones]

Tabla 3.10. Criterios para diseñar una adecuada tarea Bebras (Dagiene & Futschek, 2008)

<i>Las buenas tareas Bebras...</i>	<i>Explicación</i>
<i>Están relacionadas con las Ciencias de la Computación</i>	<i>El objetivo del concurso es promover el interés y desarrollo del pensamiento computacional entre los escolares</i>
<i>Permiten experiencias de aprendizaje</i>	<i>En la resolución de los problemas, los estudiantes deben aprender algo interesante, que les produzca interés y satisfacción</i>
<i>Cada tarea puede ser resuelta en alrededor de 3 minutos</i>	<i>3 minutos es el tiempo promedio para resolver una tarea</i>
<i>Cada tarea debe tener especificado un nivel de dificultad</i>	<i>1/3 de las tareas deben ser sencillas (Nivel A), 1/3 de las tareas deben ser intermedias (Nivel B), y 1/3 de las tareas deben ser difíciles (Nivel C)</i>
<i>Cada tarea debe estar dirigida a un grupo específico de edad</i>	<i>Actualmente, se distinguen los siguientes grupos de edad: Benjamines (equivalentes a nuestro 5º-6º Primaria); Cadetes (1º-2º ESO); Junior (3º-4º ESO); y Senior (1º-2º Bachillerato)</i>
<i>Deben ser independientes de cualquier currículum nacional en particular</i>	<i>Las tareas no están basadas en los contenidos curriculares concretos de Informática de ningún sistema educativo nacional; sino que se orientan a la resolución de problemas reales, y significativos para los estudiantes, a través de los principios y conceptos generales de la computación</i>
<i>Deben ser independientes de cualquier sistema informático en particular</i>	<i>Las tareas no deben basarse en (ni dar por hecho) ningún conocimiento previo particular sobre sistemas operativos, aplicaciones o lenguajes informáticos concretos.</i>
<i>Deben tener enunciados fácilmente comprensibles</i>	<i>La enunciación de la tarea debe ser tan sencilla como sea posible, utilizando, cuando sea factible: imágenes, ejemplos, una historia que contextualice y dé significatividad a la tarea, interactividad...</i>
<i>Cada tarea puede ser presentada y resuelta en una sola página-pantalla</i>	<i>Cada tarea no debe exceder los límites de una página-pantalla; tratando de evitar el 'scroll' con el ratón</i>
<i>Son resolubles a través de un ordenador, sin necesidad de lápiz y papel, ni de software o hardware adicional</i>	<i>Para su resolución, es suficiente con el pensamiento computacional de los estudiantes. No hace falta ni calculadora, ni otros medios adicionales</i>
<i>Son políticamente correctas</i>	<i>Las buenas tareas no contienen estereotipos de género, raciales o religiosos</i>
<i>Deberían ser divertidas</i>	<i>El concurso tiene carácter lúdico y su fin último es motivar a los estudiantes hacia el desarrollo de sus habilidades computacionales, como expresión máxima de la alfabetización digital en el siglo XXI</i>
<i>Deberían incluir imágenes</i>	<i>Las imágenes deben hacer la tarea más divertida. Pero no son meras ilustraciones, sino que deben servir de apoyo para el pensamiento computacional del estudiante.</i>
<i>Deberían incluir elementos interactivos</i>	<i>Aunque el formato de respuesta más habitual de las tareas Bebras es la elección múltiple entre alternativas ('multiple choice'), siempre que sea posible se tratará de formatear la tarea de manera que pueda ser resuelta de algún modo interactivo más allá de clicar sobre la opción de respuesta elegida</i>
<i>Deberían ofrecer 'feed-back' inmediato</i>	<i>Una vez completada la tarea, el estudiante debe recibir 'feed-back' inmediato para tener la certeza de haberlo hecho correctamente</i>

En lo relativo a su contenido, cada tarea Bebras puede ser asignada en, al menos, una de las siguientes categorías:

- ✓ **INF:** *‘Information comprehension’* (‘Comprensión de la información’). Concepción, comprensión y representación de la información (simbólica, numérica, gráfica). Sistemas de codificación y encriptación.
- ✓ **ALG:** *‘Algorithmic thinking’* (‘Pensamiento algorítmico’). Descripción y formalización de acciones según ciertas reglas. Fundamentos de programación.
- ✓ **USE:** *‘Using computer systems’* (‘Uso de sistemas informáticos’). Principios generales de interacción con los componentes informáticos: motores de búsquedas, directorios de archivos...
- ✓ **STRUC:** *‘Structures, patterns and arrangements’* (‘Estructuras, patrones y disposiciones’). Combinatoria, estructuras binarias (p.e. *‘binary trees’* o ‘árboles binarios’) , estructuras discretas (p.e. estructuras condicionales)
- ✓ **PUZ:** *‘Puzzles’*. Puzzles y juegos lógicos, mapas mentales.
- ✓ **SOC:** *‘ICT and Society’* (‘TIC y Sociedad’). Aspectos sociales, éticos, culturales, y legales, de las tecnologías de la información y la comunicación.

Además, en un reciente trabajo (Dagiene & Stupuriene, 2014), se trata de categorizar adicionalmente cada tarea Bebras en función de la taxonomía Bloom (1972), en su versión revisada para el contexto digital: *‘Remembering’* (‘recordar’); *‘Understanding’* (‘comprender’); *‘Applying’* (‘aplicar’); *‘Analysing’* (‘analizar’); *‘Evaluating’* (‘evaluar’); *‘Creating’* (‘crear’).

El Concurso Internacional Bebras finalmente se materializa en la aplicación de una prueba, habitualmente *‘on-line’*, compuesta del subconjunto de tareas finalmente escogidas y traducidas por cada país del *‘pool’* común de ese año, para cada uno de los grupos de edad. Las pruebas suelen tener una longitud de alrededor de 20 tareas (puede haber ligeras variaciones de un país a otro, y de un grupo de edad a otro), e incluyen tareas de diverso grado de dificultad. Las tareas adquieren en su mayoría el formato de ítems de elección múltiple. Por ejemplo, en la Figura 3.27 vemos la tarea *‘Ice Cream Machine’* (‘La Máquina de Helados’), extraída de la edición 2013 del Concurso Bebras: es una tarea dirigida al grupo de edad ‘Benjamines’ (5º-6º Primaria); adscrita a la categoría ‘ALG’ (pensamiento algorítmico); incluida en el nivel ‘comprender’ de la taxonomía de Bloom.

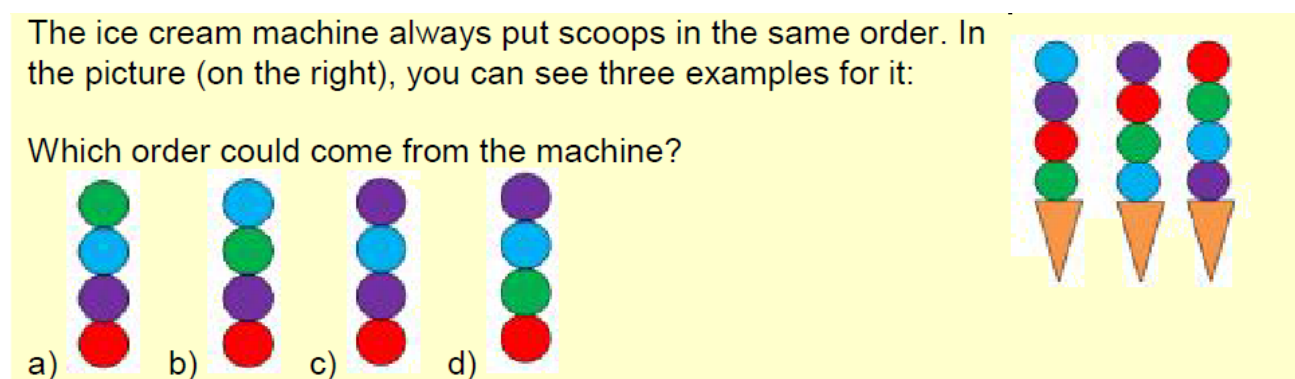
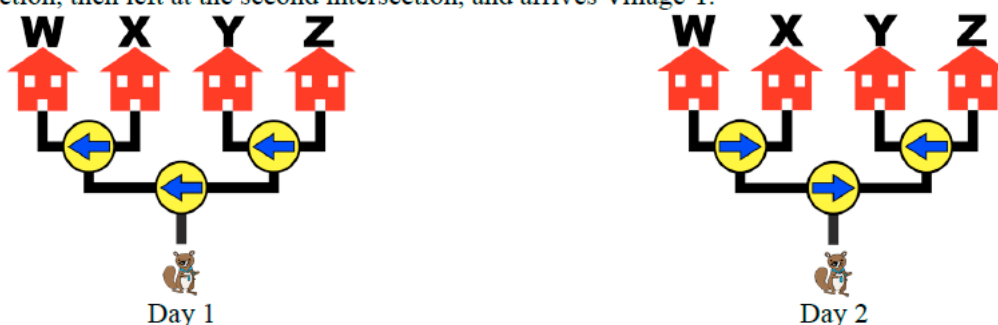


Figura 3.27. Tarea *‘Ice Cream Machine’* de la Edición 2013 del Concurso Internacional Bebras [Traducción: “*La Máquina de helados siempre pone las bolas en el mismo orden. En el dibujo de la derecha, puedes ver 3 ejemplos de ello. ¿Qué orden (patrón) podría salir de la máquina?*”]

En la Figura 3.28, se ilustra la tarea ‘*Visiting Friends*’ (‘Visitando amigos’), también extraída de la edición 2013 del concurso: es una tarea dirigida al grupo de ‘Juniors’ (3º-4º ESO); adscrita a las categorías ‘ALG’ (pensamiento algorítmico) y ‘STRUC’ (estructuras); incluida en el nivel ‘analizar’ de la taxonomía de Bloom.

Mr. Beaver has 4 friends living in different villages, and he plans to visit one of these friends every afternoon. Initially, all arrows point to the left road. When passing the intersection, Mr. Beaver would switch the arrow to the opposite direction. For example, on day 1, Mr. Beaver takes the road on the left at the first intersection, takes the left road on the second intersection, and reaches Village W. On day 2, Mr. Beaver turns right at the first intersection, then left at the second intersection, and arrives Village Y.



Which village will Mr. Beaver visit on day 30?

- A) Village W
- B) Village X
- C) Village Y
- D) Village Z

Figura 3.28. Tarea ‘*Visiting friends*’²⁶⁶ de la Edición 2013 del Concurso Internacional Bebras

Tal y como puede comprobarse, el enfoque general de las tareas Bebras es la solución de problemas ‘reales’ y significativos para los estudiantes, a través de la transferencia y proyección del pensamiento computacional sobre el contexto del problema. Éstos son independientes de cualquier software o hardware concreto, y pueden ser aplicados a sujetos sin ninguna experiencia previa en programación. Por todas estas características, se vienen señalando la tareas Bebras como un más que probable embrión para unas futuras pruebas PISA (‘*Programme for International Student Assessment*’) en el ámbito de las Ciencias de la Computación (Hubwieser & Mühling, 2014); lo que supondría un respaldo definitivo a la integración de esta disciplina de manera central en todos los currículum de la zona OCDE. Otro indicio de la más que probable expansión de las tareas Bebras al ámbito de la evaluación comparada es la progresiva adaptación que se está realizando de sus tareas para poder ser accesibles a toda la diversidad del alumnado; p.e., se recomienda consultar el reciente trabajo en el que se adaptan tareas Bebras para población escolar invidente (Jašková & Kováčová, 2015)

En cualquier caso, debemos recordar que el Concurso Internacional Bebras, por el momento, es un certamen de promoción del pensamiento computacional, no un instrumento de medida; entre otras cosas, porque no constituye un conjunto estable y determinado de ítems, sino una prueba que varía de año en año, y con ligeras modificaciones entre país y país. Sin embargo, su creciente expansión ha

²⁶⁶ Traducción: “El Sr. Castor tiene 4 amigos viviendo en pueblos diferentes, y planea visitar a uno de estos amigos cada tarde. Inicialmente, todas las flechas apuntan al camino de la izquierda. Cuando atraviesa la intersección, el Sr. Castor gira la flecha a la dirección opuesta. Por ejemplo, en el día 1, el Sr. Castor toma el camino de la izquierda en la primera intersección y otra vez el camino de la izquierda en la segunda intersección, llegando al pueblo W. En el día 2, el Sr. Castor toma el camino de la derecha en la primera intersección, y el de la izquierda en la segunda, llegando al pueblo Y ¿Qué pueblo visitará el Sr. Castor el día 30?”

despertado el interés de investigadores y psicómetras, que han comenzado a indagar sobre sus posibles virtudes como instrumento de medida del pensamiento computacional. Así, se han publicado recientemente estudios descriptivos sobre el nivel de ejecución de los estudiantes en las tareas Bebras correspondientes a ediciones del concurso celebradas en Alemania (Hubwieser & Mühling, 2014, 2015), Italia (Belletini *et al.*, 2015), Taiwán (Lee, Lin, & Lin, 2014) o Turquía (Kalelioğlu, Gülbahar, & Madran, 2015). En todos ellos se reportan rendimientos significativamente superiores en los chicos frente a las chicas, en la mayoría de las tareas.

Adicionalmente, hay un par de estos grupos de investigación (Belletini *et al.*, 2015; Hubwieser & Mühling, 2015) que están buscando subconjuntos de tareas Bebras que satisfagan los requerimientos de la Teoría de Respuesta al Ítem (TRI), a saber, que las tareas de dicho subconjunto formen una escala unidimensional y midan un mismo rasgo latente (*'single latent trait'*); de la cual se derive un futuro instrumento que pueda someterse a un proceso completo de validación. Son líneas de investigación aún por concluir.

3.3.2.2. *Un caso particular: Dr. Scratch*

Dr. Scratch²⁶⁷ es una aplicación web libre y de código abierto que permite analizar de forma simple y automática proyectos programados con Scratch²⁶⁸ (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010), así como obtener retroalimentación que puede usarse para mejorar las habilidades de programación y desarrollar el pensamiento computacional (PC) de los estudiantes. Dr. Scratch es una herramienta desarrollada por el grupo de investigación GSyC/LibreSoft²⁶⁹ de la Universidad Rey Juan Carlos (Moreno-León & Robles, 2014, 2015). Para analizar un proyecto con Dr. Scratch puede subirse a la web de la herramienta un archivo de extensión *.sb* o *.sb2*, ya que la herramienta soporta tanto proyectos de la versión Scratch 1.4 como 2.0, o bien se puede copiar-pegar directamente la URL del proyecto Scratch a analizar.

Cuando se analiza un proyecto Scratch, Dr. Scratch informa al usuario sobre el grado de desarrollo del pensamiento computacional demostrado en el mismo, asignando una puntuación total de PC. Al estar basado en *'Hairball'*²⁷⁰, Dr. Scratch detecta adicionalmente ciertos malos hábitos de programación o errores potenciales, como: el uso de nombres no significativos para los personajes; la repetición de código; existencia de código que no llega a ejecutarse; y/o la inicialización incorrecta de atributos de objetos.

Para asignar la puntuación total de PC al proyecto, Dr. Scratch infiere la competencia demostrada por el programador en cada una de las siguientes siete dimensiones: 'Abstracción y descomposición de problemas', 'Paralelismo', 'Pensamiento lógico', 'Sincronización', 'Nociones algorítmicas de control de flujo', 'Interactividad con el usuario' y 'Representación de la información'. La evaluación de la competencia en cada una de estas dimensiones sigue las reglas-algoritmos de la Tabla 3.11; que ha sido diseñada en base a la investigación previa sobre evaluación de proyectos Scratch (p.e., Seiter

²⁶⁷ <http://drscratch.org/>

²⁶⁸ <https://scratch.mit.edu/>

²⁶⁹ Grupo de investigación sobre Ingeniería del Software Libre, adscrito a la Escuela Técnica Superior de Ingeniería de Telecomunicación de la Universidad Rey Juan Carlos (<http://gscet.urjc.es/>)

²⁷⁰ *'Hairball'* es un analizador de código estático que detecta problemas potenciales en los programas, como código que nunca se ejecuta, mensajes que no recibe ningún objeto, o atributos que no se inicializan correctamente (Boe *et al.*, 2013)

& Foreman, 2013; Wilson, Hainey, & Connolly, 2012; Wolz, Hallberg, & Taylor, 2011), combinada con algunas propuestas de docentes de distintos niveles educativos que utilizan Scratch en sus clases.

Tabla 3.11. Criterios que aplica Dr. Scratch²⁷¹ para medir el nivel de competencia en las distintas dimensiones del PC

Dimensión del PC	Nivel de competencia: Se asigna en función de si en el proyecto Scratch analizado aparece...			
	Ninguno (0)	Básico (1 punto)	Intermedio (2 puntos)	Avanzado (3 puntos)
Abstracción y descomposición de problemas	-	Más de un programa y más de un objeto	Definición de bloques propios (funciones)	Uso de clones
Paralelismo	-	Dos programas en 'bandera verde'	Dos programas en 'tecla presionada', dos programas en 'al presionar' el mismo objeto	Dos programas en 'cuando reciba mensaje', crear clon, dos programas en 'cuando %s es > %', dos programas en 'cuando el escenario cambie a'
Pensamiento lógico	-	Si ('If')	Si/sino ('If/else')	Operaciones lógicas
Sincronización	-	Esperar	Enviar, cuando reciba mensaje, parar todos, parar programas, parar programas del objeto	Esperar hasta, cuando el escenario cambie a, enviar y esperar
Control de flujo	-	Secuencia de bloques	Repetir, por siempre	Repetir hasta
Interactividad con el usuario	-	Bandera verde	Tecla presionada, objeto presionado, preguntar y esperar, bloques de operaciones con ratón	Cuando %s es >%s, vídeo, audio
Representación de la información	-	Modificadores de propiedades de objetos	Operaciones con variables	Operaciones con listas

Para ilustrar cómo asigna Dr. Scratch un nivel de competencia en cada dimensión del PC, en la Figura 3.29 se muestran los 3 niveles de 'Pensamiento lógico', en función del tipo de bloques utilizados en el proyecto Scratch sometido a análisis.

En función de la puntuación total (o 'Mastery') de PC asignada al proyecto por Dr. Scratch, que puede oscilar de 0 a 21 puntos, la cantidad de información de 'feed-back' mostrada en la pantalla de resultados es distinta. Así, si el nivel de PC es bajo (≤ 10 puntos), se asume que el usuario es un programador novato y, en consecuencia, la herramienta solamente muestra información básica sobre las mejoras más importantes que pueden realizarse en el código del proyecto. Según va aumentando la puntuación, Dr. Scratch muestra más información de los proyectos analizados. Así, los usuarios avanzados reciben un informe de retroalimentación con toda la información disponible, tanto en términos de habilidades de PC que pueden desarrollar, como en lo relativo a los malos hábitos de programación que deben desterrar. La Figura 3.30 ilustra las diferencias en cantidad y complejidad de la información mostrada en pantalla en función de la puntuación de PC obtenida.

²⁷¹ Puede consultarse el código implementado en Dr. Scratch para ejecutar automáticamente estos algoritmos en: <https://github.com/jemole/hairball/blob/master/hairball/plugins/mastery.py>

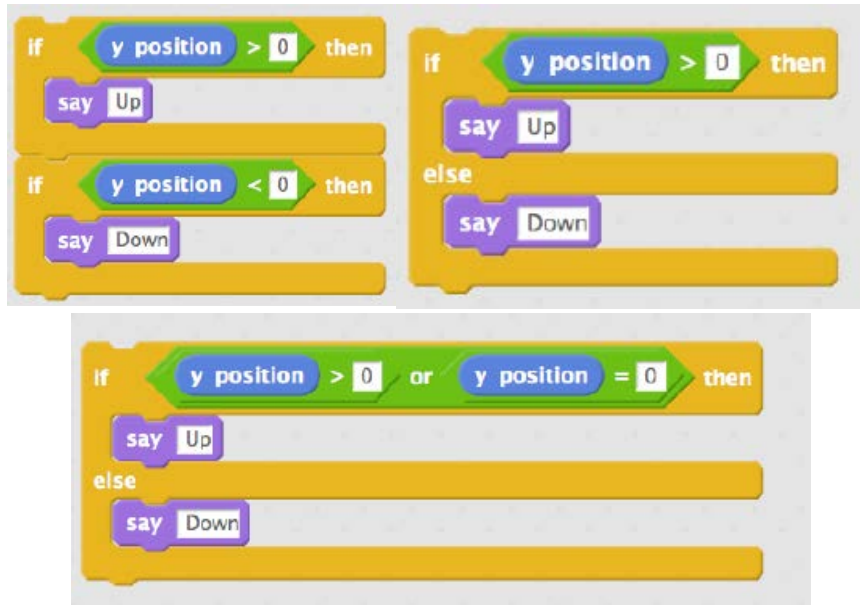


Figura 3.29. Diferentes niveles de competencia en la dimensión ‘Pensamiento lógico’: arriba a la izquierda, nivel básico (*‘If’*); arriba a la derecha, nivel intermedio (*‘If/else’*); abajo, nivel avanzado (*‘If/else’* con operadores lógicos)

The image displays two screenshots of the Dr. Scratch website, illustrating feedback based on the user's competency level.

Top Screenshot (BASIC Level):

- Score:** 7/21
- Level:** BASIC!
- Message:** "You're at the first of a great adventure... Continue that way!"
- Concept Progress:**

Concept	Points
Abstraction	2/3
Parallelization	1/3
Logic	0/3
Synchronization	1/3
FlowControl	1/3
UserInteractivity	1/3
DataRepresentation	1/3
- Want to level up?** Click here to go to your project.
- Issues:**
 - Parallelization
 - Logic
 - Synchronization
 - FlowControl
 - UserInteractivity
 - DataRepresentation

Bottom Screenshot (MASTER Level):

- Score:** 18/21
- Level:** MASTER!
- Message:** "You're the master of the universe!!!"
- Concept Progress:**

Concept	Points
Abstraction	2/3
Parallelization	3/3
Logic	3/3
Synchronization	3/3
FlowControl	3/3
UserInteractivity	2/3
DataRepresentation	2/3
- Want to level up?** Click here to go to your project.
- Issues:**
 - Abstraction
 - UserInteractivity
 - DataRepresentation
- Sprite naming details:**
 - Sprite4, Sprite2, Sprite8, Sprite5, Sprite31, Sprite36, Sprites53, Sprite48, Sprite50, Sprites41, Sprite42, Sprite43, Sprites47, Sprite52, Sprites1, Sprites48, Sprites44, Sprites45, Sprites46, Sprites7, Sprites1, Sprites12, Sprites21, Sprites14, Sprites17, Sprites20, Sprites18, Sprites13.
- Summary of issues:**
 - 6 duplicated scripts
 - 31 sprites naming
 - 0 dead code
 - 26 sprite attributes

Figura 3.30. Retroalimentación de Dr. Scratch en función del nivel de competencia demostrado

En resumen, Dr. Scratch es una aplicación web que permite a educadores y alumnos analizar automáticamente proyectos Scratch, el lenguaje de programación más utilizado globalmente en educación primaria y secundaria. Con Dr. Scratch, alumnos y educadores pueden comprobar si los proyectos se han programado correctamente, aprender de sus errores y recibir retroalimentación para mejorar su código, y desarrollar su capacidad de pensamiento computacional (PC). Otro de los objetivos de Dr. Scratch, además de ayudar al docente en las tareas de evaluación, es ser un estímulo para animar a los aprendices a seguir mejorando sus habilidades de programación. Para comprobar la efectividad de la herramienta en este sentido, en un reciente estudio en el que este doctorando ha tenido el privilegio de participar (Moreno-León, Robles, & Román-González, 2015), se organizaron talleres en 8 colegios con alumnos de entre 10 y 14 años en los que los estudiantes analizaron uno de sus proyectos Scratch con Dr. Scratch, leyeron la información del informe de resultados e intentaron mejorar sus proyectos usando los consejos ofrecidos por la herramienta. Al finalizar el taller, los alumnos mejoraron significativamente su puntuación de PC así como sus habilidades como programadores.

3.4. Síntesis: hacia una consolidación del pensamiento computacional como constructo psicopedagógico

A lo largo del presente capítulo ha quedado patente el estado aún incipiente del ‘pensamiento computacional’ (PC) como constructo psicopedagógico de estudio. Así, no existe una definición consensuada del PC, y se percibe una notable dispersión y confusión terminológica alrededor del mismo. Ello tiene que ver, en esencia, con que las Ciencias de la Computación han puesto tradicionalmente su foco en generar nuevas y mejores tecnologías, y no tanto en construir un *corpus* sistemático de conceptos para ser sometidos a transmisión y medición educativas, en especial en las etapas pre-universitarias; hecho que debemos contribuir a paliar si deseamos que las Ciencias de la Computación se integren con fuerza en el corazón de nuestros sistemas educativos.

Igualmente, en el campo de las intervenciones educativas para el desarrollo del PC se comprueba una gran variedad y heterogeneidad de las mismas: programas de duración variable, en contextos y poblaciones distintas, con metodologías diferentes, y en la mayoría de ocasiones investigadas a través de diseños débiles (estudios descriptivos o de casos). Situación parecida nos encontramos en el campo de la medida y evaluación del PC, existiendo una notable ausencia de instrumentos estandarizados, que hayan atravesado una completa validación, para ser aplicados sobre poblaciones escolares.

Así las cosas, creemos valioso y necesario: por una parte, diseñar y validar algún instrumento de medida del PC, dirigido a población escolar española; por otra parte, evaluar programas de desarrollo del PC en contextos escolares a través de diseños experimentales o *cuasi-experimentales* (que incluyan grupos ‘control’ y ‘de tratamiento’; con medidas estandarizadas *pre* y *post*). Ambas líneas pueden marcar el punto de inicio para situar la investigación educativa del PC en los niveles propios de otros constructos psicopedagógicos ya bien establecidos. A ello dedicaremos, respectivamente, los Capítulos 6 y 7 de esta tesis doctoral, ya en la parte empírica que nos disponemos a afrontar.

II. PARTE EMPÍRICA

La parte práctica de la presente tesis doctoral está compuesta por tres estudios empíricos.

En primer lugar, se desarrolla un estudio descriptivo-exploratorio (estudio no experimental) mediante metodología de encuesta sobre el evento ‘La Hora del Código’²⁷² (*‘The Hour of Code’*), promovido por la fundación Code.org²⁷³ y considerado como la mayor iniciativa a nivel mundial para la promoción y desmitificación del aprendizaje de código informático en las escuelas, y su impacto en España durante las ediciones de diciembre 2013 y diciembre 2014. En este estudio han participado un total de 29 centros educativos procedentes de múltiples puntos de la geografía española, acumulándose una muestra total de 1.662 estudiantes encuestados. Las conclusiones de este primer estudio descriptivo-exploratorio orientarán posteriormente el diseño de los otros dos estudios empíricos de esta tesis doctoral.

En segundo lugar, se procede a la validación de un instrumento, el Test de Pensamiento Computacional (TPC), específicamente diseñado para población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO), pero con aplicaciones complementarias en el último ciclo de Educación Primaria (5º Primaria – 6º Primaria) y en el segundo ciclo de la Educación Secundaria Obligatoria (3º ESO y 4º ESO). Para la validación de contenido del TPC hemos contado con el juicio de 20 expertos; y la versión definitiva del instrumento ha sido aplicada sobre un total de 1.251 estudiantes, de 24 centros educativos distintos. A partir de dicha aplicación hemos podido estudiar y establecer la fiabilidad del test así como sus diferentes indicios de validez: criterial (tanto con medidas concurrentes como predictivas), discriminante, convergente, y factorial. Todo este proceso de validación nos ha permitido utilizar posteriormente el TPC como herramienta de evaluación en el tercer estudio empírico, que se sintetiza en el siguiente párrafo.

En tercer lugar, se procede a la evaluación de programas de codigoalfabetización y desarrollo del pensamiento computacional. Se realiza una evaluación principal, en profundidad, sobre el programa-curso ‘K-8 Intro to Computer Science’ (Code.org, 2015a) implementado en 1º y 2º Ciclo de la ESO durante el 2º trimestre del año académico 14/15 en 12 centros educativos de la Comunidad Valenciana. Dicha evaluación principal se acompaña de una evaluación complementaria y de menor profundidad, realizada sobre el programa-curso ‘K-5 Computer Science Fundamentals (Course 2)’ (Code.org, 2015b) implementado en 3º Ciclo de Educación Primaria durante el 2º trimestre del año académico 14/15 en 1 centro educativo de la Comunidad de Andalucía (Sevilla). Finalmente, se presenta un breve estudio de casos de alumnos con altas capacidades computacionales de primer ciclo de la ESO que, adicionalmente al programa-curso ‘K-8 Intro to Computer Science’ seguido por sus compañeros de aula ordinaria, siguieron un programa-curso complementario de mayor dificultad a través de la plataforma Khan Academy: ‘Computer Programming’ (Khan Academy, 2015); mediante una metodología de ‘clase invertida’ (*‘flipped classroom’*) (Tourón & Santiago, 2015)

²⁷² <https://hourofcode.com>

²⁷³ <https://code.org/>

CAPÍTULO 4

OBJETIVOS Y METODOLOGÍA

4.1. Objetivos

Distinguimos entre objetivos generales y objetivos específicos.

4.1.1. Generales

Los objetivos generales de esta parte empírica son:

- Objetivo 1 (**O₁**): Describir el impacto del evento ‘La Hora del Código’ (a partir de ahora, siguiendo la abreviatura inglesa habitual, ‘**HoC**’) en España a lo largo de sus dos primeras ediciones: HoC-2013 y HoC-2014.
- Objetivo 2 (**O₂**): Explorar evidencias empíricas en dicho evento ‘**HoC**’ que den apoyo y fundamento a nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.
- Objetivo 3 (**O₃**): Diseñar y validar un instrumento que mida el nivel de desarrollo del pensamiento computacional, o ‘Test de Pensamiento Computacional’ (**TPC**), en población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO).
- Objetivo 4 (**O₄**): Evaluar el programa-curso ‘*K-8 Intro to Computer Science*’ ([Code.org, 2015a](#)), desde el marco fijado por nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.
- Objetivo 5 (**O₅**): Evaluar el programa-curso ‘*K-5 Computer Science Fundamentals (Course 2)*’ ([Code.org, 2015b](#)), desde el marco fijado por nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.
- Objetivo 6 (**O₆**): Estudiar casos de sujetos con alta capacidad computacional en el contexto del programa-curso ‘*Computer Programming*’ ([Khan Academy, 2015](#)); desde el marco fijado por nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.

Los objetivos **O₁** y **O₂** son abordados en el **Capítulo 5** (primer estudio empírico); el objetivo **O₃** es abordado en el **Capítulo 6** (segundo estudio empírico); y los objetivos **O₄**, **O₅** y **O₆** son abordados en el **Capítulo 7** (tercer estudio empírico).

4.1.2. Específicos

Los objetivos específicos se enuncian de manera explícita en los capítulos en los que se abordan respectiva y sucesivamente los objetivos generales anteriores. Así:

- Los objetivos generales **O₁** y **O₂** se abordan en el **Capítulo 5** (primer estudio empírico), y sus objetivos específicos se enuncian explícitamente en el [apartado 5.2](#).
- El objetivo general **O₃** se aborda en el **Capítulo 6** (segundo estudio empírico), y sus objetivos específicos se enuncian explícitamente en el [apartado 6.2](#).
- El objetivo general **O₄** se aborda en el **Capítulo 7** (tercer estudio empírico), y sus objetivos específicos se enuncian explícitamente, a modo de hipótesis específicas, en el [epígrafe 7.1.2](#).
- El objetivo general **O₅** se aborda en el **Capítulo 7** (tercer estudio empírico), y sus objetivos específicos se enuncian explícitamente, a modo de hipótesis específicas, en el [epígrafe 7.2.2](#).
- El objetivo general **O₆** se aborda en el **Capítulo 7** (tercer estudio empírico), y sus objetivos específicos se enuncian explícitamente en el [epígrafe 7.3.2](#).

4.2. Metodología

En consonancia con el principio general de complementariedad metodológica ([García Hoz, 1991](#); [Pérez Juste, 2000](#)), se aplican metodologías distintas en función de los objetivos generales planteados. Estos objetivos se abordan a través de una sucesión de tres estudios empíricos concatenados. A continuación, catalogamos los tres estudios en función de su metodología; para ello nos apoyamos en las taxonomías y terminologías, por orden de aparición, de [Montero y León \(2007\)](#), [Colás y Buendía \(1992\)](#), y [García Llamas \(2003\)](#).

- El **primer estudio**, que aborda los objetivos generales **O₁** y **O₂**, puede clasificarse como *estudio descriptivo de poblaciones mediante encuestas con muestras probabilísticas*; siendo éstas, *muestras independientes sucesivas* ([Montero & León, 2007](#)). Las encuestas combinan la aproximación *cuantitativa* (preguntas cerradas) y *cualitativa* (preguntas abiertas). A su vez, dentro del primer estudio, se puede diferenciar que:
 - El objetivo general **O₁** se aborda desde una perspectiva *longitudinal*, comparando la evolución de las dos *muestras independiente sucesivas* (HoC-2013 y HoC-2014)
 - El objetivo general **O₂** se aborda desde una perspectiva *transversal*, acumulando ambas muestras independientes sucesivas como una sola cohorte de población bianual, que es posteriormente analizada.
- El **segundo estudio**, que aborda el objetivo general **O₃**, puede clasificarse como *estudio instrumental* ([Montero & León, 2007](#)) de diseño, construcción y validación de una prueba psicométrica; desde la ‘Teoría Clásica de los Tests’ (TCT) ([Muñiz, 1992](#); [Navas Ara, 2001](#)). El proceso de desarrollo de nuestro ‘Test de Pensamiento Computacional’ (TPC) sigue la guía metodológica de [Buffum et al. \(2015\)](#) para la construcción y validación de instrumentos de medida en el área de las Ciencias de la Computación con poblaciones ‘*middle school*’ (de 6º Primaria a 2º ESO); guía que está alineada, a su vez, con los estándares para la medición psicométrica educativa fijados por la *American Educational Research Association* (AERA) y la *American Psychological Association* (APA) (ver [AERA, APA, & NCME, 2014](#))

- El **tercer estudio** está, a su vez, dividido en tres sub-estudios interrelacionados:
- El primer sub-estudio, que aborda el objetivo general **O₄**, aplica distintas metodologías en función del tipo de hipótesis específicas sometidas a contraste:
 - ✓ Las hipótesis específicas enunciadas en el marco de la directiva de *adecuación* son contrastadas a través de un *diseño cuasi-experimental de dos grupos no equivalentes* (Colás y Buendía, 1992); que incluye medidas psicométricas *pretest* y *postest*, tanto en el grupo de tratamiento como en el grupo de control, habiéndose constituido dichos grupos de manera natural. En terminología de Montero y León (2007), hablaríamos de un *cuasi-experimento pre-post, de dos grupos, uno de cuasi-control*.
 - ✓ Las hipótesis específicas enunciadas en el marco de la directiva de *viabilidad* son contrastadas a través de una metodología *descriptivo-cuantitativa apoyada en encuestas* (preguntas cerradas) e indicadores objetivos de seguimiento (*'tracking'*) del curso K-8 (Code.org, 2015a)
 - ✓ Las hipótesis específicas enunciadas en el marco de la directiva de *relevancia* son contrastadas a través de una metodología *interpretativo-cualitativa, apoyada en encuestas* (preguntas abiertas).
 - El segundo sub-estudio, que aborda el objetivo general **O₅**, aplica distintas metodologías en función del tipo de hipótesis específicas sometidas a contraste:
 - ✓ La hipótesis específica enunciada en el marco de la directiva de *adecuación* es contrastada a través de un *diseño pre-experimental de un solo grupo con pretest y postest* (Colás y Buendía, 1992); que sólo incluye grupo de tratamiento (no hay grupo de control). En terminología de Montero y León (2007), hablaríamos de un *cuasi-experimento débil pre-post, un grupo*.
 - ✓ La hipótesis específica enunciada en el marco de la directiva de *viabilidad* es contrastada a través de una metodología *descriptivo-cuantitativa*, apoyada en indicadores objetivos de seguimiento (*'tracking'*) del curso K-5 (Code.org, 2015b)
 - El tercer sub-estudio, que aborda el objetivo general **O₆**, sigue una metodología de *estudio de casos múltiples* (García Llamas, 2003) o *estudio de caso, múltiple* (Montero & León, 2007); apoyado en medidas de distinto tipo en función del objetivo específico abordado:
 - ✓ El objetivo específico enunciado en el marco de la directiva de *adecuación* es abordado a través del análisis de las medidas psicométricas de los sujetos-casos.

- ✓ El objetivo específico enunciado en el marco de la directiva de *viabilidad* es abordado a través de los indicadores de seguimiento (*'tracking'*) de los sujetos-casos en el curso *'Computer Programming'* (Khan Academy, 2015)
- ✓ El objetivo específico enunciado en el marco de la directiva de *relevancia* es abordado a través del análisis de las producciones (programas y código informático escrito) de los sujetos-casos en el curso *'Computer Programming'*

4.2.1. Plan de análisis de datos

Para la realización efectiva de los objetivos generales, se lleva a cabo el siguiente plan de análisis de datos, que diferenciamos para cada uno de los tres estudios empíricos:

- **En el primer estudio empírico** (*estudio descriptivo mediante encuestas sobre el evento 'La Hora del Código'*):
 - **Con respecto a los datos cuantitativos:** se realizan análisis descriptivos (frecuencias, porcentajes, tendencia central, variabilidad y asimetría); análisis de asociación entre variables nominales (coeficientes de contingencia C); análisis de correlación entre variables al menos de cuasi-intervalo (correlaciones de Pearson r); análisis inferenciales de diferencias entre categorías (Chi-cuadrado χ^2), de diferencias de medias entre dos grupos (t de Student) o entre n -grupos (análisis de varianza ANOVA con F de Fisher); análisis de tamaños del efecto (d de Cohen); y distintos análisis multivariante (análisis de varianza multivariado MANOVA, análisis de componentes principales con rotación Varimax, análisis de *cluster*, análisis de regresión múltiple, y análisis discriminante). Todo ello acompañado de sus correspondientes tablas (tablas de frecuencia, tablas de contingencia y tablas resumen), y gráficas de representación (gráficas de barras, gráficas de sectores, histogramas, diagramas de dispersión, y otras representaciones construidas *ad hoc* para ilustrar los resultados obtenidos).
 - **Con respecto a los datos cualitativos:** se realizan análisis de contenido textual sobre las respuestas a las preguntas abiertas de las encuestas; con desarrollo inductivo de categorías de análisis, a partir de un sistema de codificación abierta (García Llamas, 2003)
- **En el segundo estudio empírico** (*estudio instrumental del Test de Pensamiento Computacional o TPC*): para la validación de contenido del instrumento, mediante juicio de expertos, se calculan coeficientes de correlación intra-clase (α) con el fin de determinar la fiabilidad inter-jueces. Para establecer la psicometría general del TPC, se realizan análisis descriptivos sobre la puntuación total en el mismo (frecuencias, porcentajes, tendencia central, variabilidad, asimetría, curtosis y baremación por percentiles), y análisis inferenciales sobre diferencias de rendimiento en el TPC según distintas variables (t de student y ANOVA con F de Fisher); así como el cálculo de los tamaños del efecto asociados a dichas diferencias (d de Cohen). Para el análisis de ítems del TPC, se calculan 'índice de dificultad' (p_i), 'correlación ítem-total' (*'item-total correlation'* o ITC), e 'índice de discriminación' (D). Para establecer la fiabilidad del TPC se calcula el estadístico Alfa de Cronbach de consistencia interna (α), y el coeficiente ρ_{xx} de estabilidad temporal. Para la validez criterial y

convergente de nuestro instrumento, se calculan coeficientes de correlación de Pearson (r) entre las puntuaciones totales en el TPC con los distintos criterios y medidas alternativas del pensamiento computacional, respectivamente. Para la validez discriminante del TPC se calcula la Curva COR de potencia diagnóstica con respecto al criterio dicotómico, que será justificado, de ‘alta capacidad computacional’. Finalmente, para establecer la validez factorial del TPC se realizará un análisis factorial exploratorio (rotación oblicua Promax)

➤ **En el tercer estudio empírico** (*evaluación de programas, con estudio de casos subsiguiente*), podemos diferenciar, a su vez, por cada uno de los sub-estudios que lo forman:

- En el primer sub-estudio (*evaluación cuasi-experimental K-8*):
 - ✓ Para contrastar las hipótesis específicas en el marco de la directiva de *adecuación*, esto es, el efecto del curso K-8 ([Code.org, 2015a](#)) sobre distintas variables dependientes, se realizan análisis de covarianza (ANCOVA con F de Fisher) sobre el *postest* de dichas variables, en función de la condición ‘experimental-tratamiento’ o ‘control’ de los grupos, y controlando las diferencias iniciales en *pretest* (covariable). Los ANCOVA se complementan con pruebas t de diferencias de medias para muestras independientes (para establecer la equivalencia entre grupos, en *pretest* y *postest*), y para muestras relacionadas-emparejadas (para establecer la ganancia *pre-post*, tanto del grupo ‘experimental’ como ‘control’). Se calculan igualmente los tamaños del efecto (d de Cohen) asociados a la F del ANCOVA y a las distintas pruebas t de Student. También se presentan gráficos resumen del diseño *cuasi-experimental* para cada una de las variables dependientes analizadas.
 - ✓ Para contrastar las hipótesis específicas en el marco de la directiva de *viabilidad*, se realizan análisis descriptivos (frecuencias, porcentajes, tendencia central y variabilidad)
 - ✓ Para contrastar las hipótesis específicas en el marco de la directiva de *relevancia*, se realiza análisis de contenido textual (frecuencias y categorías) sobre las respuestas a las preguntas abiertas de las encuestas.
- En el segundo sub-estudio (*evaluación pre-experimental K-5*): la hipótesis específica en la directiva de *adecuación* se contrasta mediante una prueba t para muestras relacionadas; y la hipótesis específica en la directiva de *viabilidad* se contrasta mediante análisis descriptivos (frecuencias, porcentajes, tendencia central y variabilidad)
- En el tercer sub-estudio (*estudio de casos de ‘alta capacidad computacional’*): se realizan análisis descriptivos sobre las puntuaciones psicométricas de los casos y sobre sus estadísticas de desempeño (*‘tracking’*) en Khan Academy, y un análisis de contenido sobre el código informático escrito por los sujetos-casos en dicha plataforma.

Globalmente, para los análisis *cuantitativos* se ha utilizado Excel 2013 y SPSS 22.0; y para los análisis *cualitativos* de contenido textual se ha utilizado Atlas.ti 7.0.

CAPÍTULO 5

PRIMER ESTUDIO: LA HORA DEL CÓDIGO EN ESPAÑA

5.1. Antecedentes

Tal y como vimos en la parte teórica de esta tesis doctoral, concretamente en el Capítulo 2, [sub-epígrafe 2.2.3.1](#), ‘La Hora del Código’²⁷⁴ (*The Hour of Code*, o simplemente ‘**HoC**’) es el mayor evento a nivel mundial para la promoción del aprendizaje de la programación informática (coloquialmente ‘*coding*’) en las escuelas. A dicho sub-epígrafe remitimos al lector para una información más detallada sobre el evento.

5.2. Objetivos específicos

Recordemos previamente que el presente estudio descriptivo-exploratorio tiene dos grandes objetivos generales:

- **Objetivo 1 (O₁):** Describir el impacto del evento ‘La Hora del Código’ (a partir de ahora, siguiendo la abreviatura inglesa habitual, ‘**HoC**’) en España a lo largo de sus dos primeras ediciones: HoC-2013 y HoC-2014
- **Objetivo 2 (O₂):** Explorar evidencias empíricas en dicho evento ‘**HoC**’ que den apoyo y fundamento a nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.

Dentro del objetivo general **O₁** nos proponemos los siguientes objetivos específicos:

- **O_{1a}:** Describir el impacto del evento a lo largo de las ediciones HoC-2013 y HoC-2014, en función del área geográfica de procedencia de los centros educativos participantes.
- **O_{1b}:** Describir el impacto del evento a lo largo de las ediciones HoC-2013 y HoC-2014, en función de la titularidad de los centros educativos participantes.
- **O_{1c}:** Describir el impacto del evento a lo largo de las ediciones HoC-2013 y HoC-2014, en función de la tipología de los centros educativos participantes.

Dentro del objetivo general **O₂** nos proponemos los siguientes objetivos específicos:

- ❖ En lo relativo a los **profesores/as** organizadores del evento ‘La Hora del Código’ en sus respectivos centros, nos planteamos los siguientes (**O_{2p*}**):
 - ✓ En el marco de la hipótesis directiva de **adecuación**:
 - **O_{2Pa}:** Explorar qué aptitudes y habilidades considera el profesorado que desarrolla la programación informática o ‘*coding*’.

²⁷⁴ <https://hourofcode.com>

- **O_{2Pb}**: Explorar si el profesorado considera el ‘coding’ como especialmente apropiado para algún tipo específico de estudiante.
 - **O_{2Pc}**: Explorar si el profesorado considera que hay algún tipo de correlación entre el nivel de ejecución de los estudiantes en tareas relativas al ‘coding’ y su rendimiento en otras áreas curriculares.
- ✓ En el marco de la hipótesis directiva de **viabilidad**:
- **O_{2Pd}**: Explorar el grado de conocimiento del profesorado sobre los recursos, plataformas-tutoriales en línea, existentes para el aprendizaje del ‘coding’.
 - **O_{2Pe}**: Explorar por qué tipo concreto de plataformas-tutoriales en línea se han decidido los profesores para ser utilizados durante el evento HoC.
 - **O_{2Pf}**: Explorar si el profesorado considera viable que a través de una actividad de tan solo 1 hora se puede introducir a los estudiantes en el mundo y la práctica del ‘coding’.
 - **O_{2Pg}**: Explorar con qué apoyos y resistencias de la comunidad educativa se ha encontrado el profesorado a la hora de organizar el evento HoC en su centro.
- ✓ En el marco de la hipótesis directiva de **relevancia**:
- **O_{2Ph}**: Explorar si el profesorado considera el aprendizaje de la programación o ‘coding’ como una competencia relevante para el futuro de sus estudiantes como ciudadanos.
 - **O_{2Pi}**: Explorar si el profesorado considera el aprendizaje de la programación o ‘coding’ como una competencia relevante para el futuro de sus estudiantes como trabajadores.
- ✓ Adicionalmente, y con fines de índole más descriptiva sobre el evento HoC, nos planteamos recabar informaciones de los profesores para:
- **O_{2Pj}**: Describir en detalle qué actividades se planificaron inicialmente en cada centro.
 - **O_{2Pk}**: Describir en detalle qué actividades se llevaron finalmente a cabo.
 - **O_{2Pl}**: Recopilar los aspectos más positivos derivados de la celebración del evento.
 - **O_{2Pm}**: Recopilar los aspectos más negativos derivados de la celebración del evento.
 - **O_{2Pn}**: Recopilar anécdotas o hechos excepcionales derivados de la celebración del evento.
 - **O_{2Pñ}**: Sondar la intención del profesorado de proseguir con actividades y currículos de aprendizaje del ‘coding’ más allá de la celebración del evento.

- ❖ En lo relativo a los **estudiantes** participantes en el evento ‘La Hora del Código’, nos planteamos los siguientes objetivos específicos (**O_{2E*}**)
 - ✓ Desde un punto de vista meramente descriptivo:
 - **O_{2Ea}**: Determinar cuántos estudiantes han oído hablar de la programación informática o ‘coding’ con anterioridad al evento.
 - **O_{2Eb}**: Determinar cuántos estudiantes han tenido experiencia previa con la programación informática o ‘coding’ con anterioridad al evento.
 - **O_{2Ec}**: Determinar con qué lenguajes informáticos han tenido contacto los estudiantes con experiencia previa en programación.
 - ✓ En el marco de la hipótesis de **adecuación**:
 - **O_{2Ed}**: Determinar en qué medida los estudiantes consideran que la actividad de ‘coding’ realizada durante el evento ha servido para desarrollar distintas aptitudes y habilidades.
 - ✓ En el marco de la hipótesis de **viabilidad**:
 - **O_{2Ee}**: Determinar en qué medida los estudiantes se perciben ‘auto-eficaces’ en tareas de ‘coding’ tras la celebración del evento.
 - **O_{2Ef}**: Determinar qué porcentaje de estudiantes pudo finalizar la actividad de ‘coding’ propuesta para el evento en su centro.
 - **O_{2Eg}**: Explorar la posible relación entre las dos cuestiones anteriores y el rendimiento académico habitual del estudiante en distintas áreas.
 - **O_{2Eh}**: Determinar en qué medida los estudiantes se sienten motivados para proseguir el aprendizaje del ‘coding’ tras el evento.
 - **O_{2Ei}**: Determinar qué porcentaje de estudiantes se muestra dispuesto a seguir aprendiendo ‘coding’ a través de plataformas-tutoriales en línea tras el evento.
 - **O_{2Ej}**: Determinar qué plataformas-tutoriales en línea de aprendizaje del ‘coding’ son más conocidos por los estudiantes.
 - ✓ En el marco de la hipótesis de **relevancia**:
 - **O_{2Ek}**: Explorar si los estudiantes consideran el aprendizaje de la programación o ‘coding’ como una competencia relevante para su futuro como ciudadanos.
 - **O_{2El}**: Explorar si los estudiantes consideran el aprendizaje de la programación o ‘coding’ como una competencia relevante para su futuro como profesionales.

5.3. Método

Para acometer los anteriores objetivos, llevamos a cabo un estudio descriptivo-exploratorio mediante metodología de encuesta; o más técnicamente, en la denominación de Montero y León (2007), un *estudio descriptivo de poblaciones mediante encuestas con muestras probabilísticas*; siendo éstas, *muestras independientes sucesivas*.

La recogida de datos se realiza en dos oleadas sucesivas, diciembre de 2013 (edición HoC-2013) y diciembre de 2014 (edición HoC-2014); con la intención de acumular una muestra representativa del territorio nacional y de tamaño suficiente. Como veremos en el siguiente epígrafe 5.3.1., la muestra productora de datos en la HoC-2013 es distinta que la muestra productora en la HoC-2014 (muestras independientes sucesivas), y su tratamiento será diferente en función del objetivo general abordado:

- El objetivo general O_1 (y sus objetivos específicos derivados) se aborda desde una perspectiva *longitudinal*, comparando la evolución de las dos muestras independiente sucesivas (HoC-2013 y HoC-2014)
- El objetivo general O_2 (y sus objetivos específicos derivados) se aborda desde una perspectiva *transversal*, acumulando ambas muestras independientes sucesivas como una sola cohorte de población bianual, con el objeto de elevar la potencia estadística de nuestros análisis posteriores.

5.3.1. Muestra

5.3.1.1. Muestra de centros

Como población se considera el conjunto total de centros educativos españoles que se registran para participar en las ediciones HoC-2013 y HoC-2014 del evento ‘La Hora del Código’. A dicha población se puede acceder cada año a través de la URL: <https://hourofcode.com/es/events/all/es>

Por un procedimiento de muestreo aleatorio ‘por lista’ se contacta por correo electrónico con un subconjunto de centros pertenecientes a dicha población, y se les invita a participar en la investigación. De esta muestra invitada, sólo una parte acaba aceptando y produciendo datos. En la Tabla 5.1 se detalla el número de centros registrados, invitados y aceptantes para cada una de las dos ediciones objeto de estudio.

Tabla 5.1. Centros registrados, invitados y aceptantes por edición de ‘La Hora del Código’

	Centros Educativos					
	Registrados	Invitados	Aceptantes ²⁷⁵	% Inv. / Reg.	% Acep. / Reg.	% Acep. / Inv.
HoC-2013	96	23	10	23,95 %	10,41 %	43,47 %
HoC-2014	490	54	19	11,02 %	3,87 %	35,18 %
Total	586	77	29	13,13 %	4,94 %	37,66 %

En las siguientes Tablas 5.2 y 5.3 se detallan los centros educativos invitados a participar en la investigación, respectivamente para las ediciones HoC-2013 y HoC-2014. En la medida que esta muestra de centros invitados es aleatoria con respecto del total de la población (conjunto de centros que se registraron en el evento), la descripción de dicha muestra de centros invitados podrá dar una imagen adecuada del conjunto en diversas variables relevantes (área geográfica de procedencia de los centros, titularidad de los centros, tipología de los centros...); dando así respuesta a los objetivos específicos enunciados más arriba $O_{1a} \dots O_{1c}$. Con ello iniciaremos el apartado 5.4 de ‘Resultados’.

²⁷⁵ Se considera ‘Aceptante’ cualquier centro que haya producido datos en alguno de los tres cuestionarios aplicados en este estudio.

Tabla 5.2. Muestra de centros invitados a la edición HoC-2013 (N = 23 centros)

Área Geográfica	Comunidad Autónoma	Centro Educativo (Web)	Población (Provincia)	Etapas Educativas impartidas ²⁷⁶	Titularidad
Centro	Castilla-La Mancha	IES El Greco (http://ieselgreco.es/)	Toledo	S, B y FP	Público
	Comunidad de Madrid	American School of Madrid (http://www.asmadrid.org/)	Pozuelo de Alarcón	I, P, S y B	Privado
		Colegio Sagrado Corazón – Fuencarral (http://www.sagradocorazonfuencarral.com/)	Madrid	I, P, S y B	Concertado
		Colegio Tajamar (http://tajamar.es/)	Madrid	I, P, S, B y FP	Concertado
Noroeste	Asturias	CP Cabueñes (http://web.educastur.princast.es/cp/cabuenes/joomla/)	Gijón	I y P	Público
	País Vasco	American School of Bilbao (http://www.asob.es/)	Berango (Vizcaya)	I, P, S y B	Privado
		Lauro Ikastola Sociedad Cooperativa (http://www.lauroikastola.eus/)	Loiu (Vizcaya)	I, P, S y B	Concertado
Este y Baleares	Cataluña	American School of Barcelona (http://www.a-s-b.com/)	Esplugues de Llobregat (Barcelona)	I, P, S y B	Privado
		Aula Escola Europea (http://www.aula-ee.com/escola/)	Barcelona	I, P, S y B	Privado
		Benjamin Franklin International School (http://www.bfischool.org/)	Barcelona	I, P, S y B	Privado
		Escola Projecte (http://www.escolaprojecte.org/)	Barcelona	I, P y S	Concertado
		Escola Vilorai (http://www.vilorai.com/)	Barcelona	I, P, S y B	Concertado
		Jesuites Casp (http://www.casp.fje.edu/)	Barcelona	I, P, S y B	Concertado
		St. Paul's School (http://www.stpauls.es/)	Barcelona	I, P, S y B	Privado
		American School of Valencia (https://www.asvalencia.org)	Puzol (Valencia)	I, P, S y B	Privado

²⁷⁶ I = Infantil; P = Primaria; S = Secundaria; B = Bachillerato; FP = Formación Profesional

Área Geográfica	Comunidad Autónoma	Centro Educativo (Web)	Población (Provincia)	Etapas Educativas impartidas ²⁷⁶	Titularidad
		IES L'Almadrava (http://www.iesalmadrava.com/)	Benidorm (Alicante)	S, B y FP	Público
		IES L'Eliana (http://iesleliana.edu.gva.es/)	L'Eliana (Valencia)	S, B y FP	Público
		IES Penyagolosa (http://mestreacasa.gva.es/web/iespenyagolosa/)	Castellón de la Plana	S y B	Público
		IES San Antonio de Benagéber (http://iesbenageber.edu.gva.es/)	San Antonio de Benagéber (Valencia)	S y B	Público
		IES Tavernes Blanques (http://mestreacasa.gva.es/web/iestavernesblanques)	Tavernes Blanques (Valencia)	S, B y FP	Público
Sur y Canarias	Andalucía	IES Francisco Romero Vargas (http://www.iesromerovargas.es/)	Jerez de la Frontera (Cádiz)	S, B y FP	Público
		IES Pedro Soto de Rojas (http://www.iespedrosoto.es/)	Granada	S, B y FP	Público
	Canarias	Colegio Nuryana (http://nuryana.com/)	La Laguna (Santa Cruz de Tenerife)	I, P, S y B	Privado

Tabla 5.3. Muestra de centros invitados a la edición HoC-2014 (N = 54 centros)

Área Geográfica	Comunidad Autónoma	Centro Educativo (Web)	Población (Provincia)	Etapas Educativas impartidas ²⁷⁷	Titularidad
		Casa de Juventud de Las Rozas (http://www.rozasjoven.es/)	Las Rozas	NF	Público
		CEIP Lope de Vega (http://www.ceiplopedevega.es/colegio.htm)	Madrid	I y P	Público
Centro	Comunidad de Madrid	Colegio Alameda de Osuna (http://www. colegio-alameda.com/)	Madrid	I, P, S y B	Privado
		Colegio Balmes (http://www.educa.madrid.org/web/cc.balmes.mostoles/)	Móstoles	I, P y S	Concertado
		Colegio Corazón Inmaculado (http://www.corazoninmaculado.es/)	Madrid	I, P, S y B	Concertado
		Colegio Educrea (http://www.grupoeducra.com/)	Villalbilla	I, P, S y B	Concertado

²⁷⁷ I = Infantil; P = Primaria; S = Secundaria; B = Bachillerato; FP = Formación Profesional; NF = Educación No Formal

Área Geográfica	Comunidad Autónoma	Centro Educativo (Web)	Población (Provincia)	Etapas Educativas impartidas ²⁷⁷	Titularidad	
Noroeste		Colegio Estudio (http://colegio-estudio.es/)	Madrid	I, P, S y B	Privado	
		Colegio Mirasur (http://www.colegiomirasur.com/)	Pinto	I, P, S y B	Privado	
		Humanitas Bilingual School (http://www.humanitastrescantos.com/)	Tres Cantos	I, P y S	Concertado	
		IES Europa (http://www.educa2.madrid.org/web/centro.ies.europa.mostoles)	Móstoles	S, B y FP	Público	
		IES Ramiro de Maeztu (http://www.educa.madrid.org/web/ies.ramirodemaetztu.madrid/)	Madrid	S y B	Público	
	Asturias		CP Cabueñes (http://web.educastur.princast.es/cp/cabuenes/joomla/)	Gijón	I y P	Público
			IESO El Sueve (http://www.iesoelsueve.es/)	Arriondas	S	Público
			The English School of Asturias (http://www.colegioingles.com/)	Pruvia de Abajo	I, P, S y B	Privado
	Cantabria		IES Leonardo Torres Quevedo (http://www.iestorresquevedo.net/)	Santander	S y B	Público
			CEIP Campo Charro (http://cpcampocharro.centros.educa.jcyl.es)	Salamanca	I y P	Público
	CEIP León Felipe (http://ceip-leonfelipe.centros.educa.jcyl.es/)		Valladolid	I y P	Público	
	CIFP Juan de Colonia (http://cifpjuandecolonia.centros.educa.jcyl.es/)		Burgos	FP	Público	
	Castilla y León		Colegio San Agustín (http://www.colegiosanagustin.net/)	Valladolid	I, P, S y B	Concertado
			Colegio Santísima Trinidad (http://www.amordediospinilla.es/)	Zamora	I, P y S	Concertado
	Galicia		IES Delicias (http://www.iesdelicias.com/)	Valladolid	S y B	Público
CEIP Noalla-Telleiro (http://www.edu.xunta.es/centros/ceipdenoalla/)			Sanxenxo (Pontevedra)	I y P	Público	
CEIP Ponte dos Brozos (http://www.edu.xunta.es/centros/ceippontebrozos/)			Arteixo (La Coruña)	I y P	Público	

Área Geográfica	Comunidad Autónoma	Centro Educativo (Web)	Población (Provincia)	Etapas Educativas impartidas ²⁷⁷	Titularidad
		CEP Xosé María Brea Segade (http://www.edu.xunta.es/centros/cepbreasegade/)	Rianxo (La Coruña)	P	Público
		IES Castro Aobre (http://www.edu.xunta.es/centros/iescastroalobrevilagarcia/)	Vilagarcía de Arousa (Pontevedra)	S, B y FP	Público
		IES Ramón Otero Pedrayo (http://www.edu.xunta.es/centros/iesoteropedrayocoruna/)	La Coruña	S	Público
		IES Rosalía de Castro (http://iesrosalia.net/)	Santiago de Compostela (La Coruña)	S y B	Público
		IES Tomiño (http://www.edu.xunta.es/centros/iesdetomino/)	Tomiño (Pontevedra)	S y B	Público
	Aragón	CEIP Hermanos Argensola (http://www.web.hermanosargensola.es/)	Zaragoza	I y P	Público
		CEIP La Jota (http://www.colegiolajota.com/)	Zaragoza	I y P	Público
		CEIP Escola Agustí Gifre (http://www.xtec.cat/ceip-agusti-gifre/)	Sant Gregori (Gerona)	I y P	Público
		CEIP Escola Àngels Alemany i Boris (http://ceipangels.blogspot.co.uk/)	Lloret de Mar (Gerona)	I y P	Público
		CEIP Escola Antoni Brusi (http://www.antonibrusi.cat/)	Barcelona	I y P	Público
Este y Baleares		CEIP Escola El Gegant del Rec (http://gegantdelrec.blogspot.co.uk/)	Salt (Gerona)	I y P	Público
	Cataluña	CEIP Escola Esteve Carles (http://blocs.xtec.cat/estvecarles/)	Lloret de Mar (Gerona)	I y P	Público
		CEIP Escola L'Estelada (http://ceiplestelada.org/)	Cànoves (Barcelona)	I y P	Público
		CEIP Escola Montfalgars (http://escolamontfalgars.cat/)	Gerona	I y P	Público
		CEIP Escola Mossèn Joan Batlle (http://blocs.xtec.cat/mjbatlle/)	Blanes (Gerona)	I y P	Público
		CEIP Escola Pere Torrent (http://blocs.xtec.cat/escolaperetorrent15/)	Lloret de Mar (Gerona)	I y P	Público

Área Geográfica	Comunidad Autónoma	Centro Educativo (Web)	Población (Provincia)	Etapas Educativas impartidas ²⁷⁷	Titularidad
		Col.legi La Mercè (http://www.lamercemartorell.cat/)	Martorell (Barcelona)	I, P, S y B	Privado
		Complex Cultural i Esportiu Montessori-Palau (http://www.montessori-palau.net/)	Gerona	I, P, S y B	Privado
		IES Institut de Terrassa (http://web.iesterrassa.cat/)	Terrasa (Barcelona)	S, B y FP	Público
		Cambridge House Community College (http://www.cambridgehouse.es/)	Rocafort (Valencia)	I, P, S y B	Privado
		CEIP Carles Salvador (http://ampacarllessalvador.es/)	Valencia	I y P	Público
	Comunidad Valenciana	English School Los Olivos (http://www.los-olivos.es/)	Godella (Valencia)	I, P, S y B	Privado
		IES Fuente de San Luis (http://iesfuentesanluis.edu.gva.es/nova/index.php)	Valencia	S, B y FP	Público
		IES La Torreta (http://ies-latorreta.es/)	Elche (Alicante)	S, B y FP	Público
		IES Ximén d'Urrea (http://ximendurrea.edu.gva.es/iesximen)	L'Alcora (Castellón)	S, B y FP	Público
	Región de Murcia	Colegio San Jorge (http://colegiosanjorge.es/)	La Alcayna (Murcia)	I, P, S, B y FP	Privado
		Colegio La Presentación (http://www.lapresentacion.com/malaga/)	Málaga	I, P, S y B	Concertado
	Andalucía	Colegio Portocarrero (http://www.colegioportocarrero.net/)	Roquetas de Mar (Almería)	I, P, S, B y FP	Concertado
Sur y Canarias		IES Ciudad de Dalías (http://www.iesdalias.es/)	Dalías (Almería)	S	Público
	Canarias	CEIP Laurisilva (http://www.gobiernodecanarias.org/educacion/3/WebC/laurisilva/)	Las Palmas de Gran Canaria (Las Palmas)	I y P	Público
	Ceuta	IES Abyla (http://www.iesabyla.com/)	Ceuta	S, B y FP	Público

En la siguiente Tabla 5.4 se detallan los 29 centros educativos aceptantes de la investigación, indicando la edición en la que fueron invitados a participar en el estudio, así como en qué cuestionarios produjeron datos (estos cuestionarios se describirán en detalle en el [epígrafe 5.3.2](#) dedicado a ‘Instrumentos’)

Tabla 5.4. Muestra de centros educativos aceptantes en ambas ediciones HoC (N = 29)

Cuestionario profesores - pre	Cuestionario profesores - post	Cuestionario estudiantes	Edición invitada
American School of Bilbao	American School of Bilbao	American School of Bilbao	HoC-2013
(The) American School of Madrid	(The) American School of Madrid	(The) American School of Madrid	HoC-2013
American School of Valencia	American School of Valencia	American School of Valencia	HoC-2013
Aula Escola Europea	-	-	HoC-2013
Benjamin Franklin International School	Benjamin Franklin International School	Benjamin Franklin International School	HoC-2013
CP Cabueñes	CP Cabueñes	-	HoC-2013
IES Francisco Romero Vargas	IES Francisco Romero Vargas	-	HoC-2013
IES L'Almadrava	-	-	HoC-2013
IES Penyagolosa	IES Penyagolosa	IES Penyagolosa	HoC-2013
IES San Antonio de Benagéber	IES San Antonio de Benagéber	IES San Antonio de Benagéber	HoC-2013
-	-	Casa de Juventud de Las Rozas	HoC-2014
CEIP Escola Agustí Gifre	CEIP Escola Agustí Gifre	CEIP Escola Agustí Gifre	HoC-2014
CEIP Escola Antoni Brusi	-	-	HoC-2014
CEIP Escola Pere Torrent	CEIP Escola Pere Torrent	CEIP Escola Pere Torrent	HoC-2014
-	-	CEIP La Jota	HoC-2014
CEIP Laurisilva	CEIP Laurisilva	CEIP Laurisilva	HoC-2014
CEIP Lope de Vega	CEIP Lope de Vega	CEIP Lope de Vega	HoC-2014
Colegio Educrea	-	Colegio Educrea	HoC-2014
-	-	Colegio Mirasur	HoC-2014
-	-	Colegio San Agustín	HoC-2014
-	-	Colegio San Jorge	HoC-2014
English School Los Olivos	-	English School Los Olivos	HoC-2014
Humanitas Bilingual School	-	-	HoC-2014
IES Abyla	IES Abyla	IES Abyla	HoC-2014
IES Castro Alobre	-	IES Castro Alobre	HoC-2014
-	-	IES Fuente de San Luis	HoC-2014
-	-	IES María Blasco	HoC-2014
IES Ramiro de Maeztu	-	IES Ramiro de Maeztu	HoC-2014
IES Ximén d'Urrea	IES Ximén d'Urrea	IES Ximén d'Urrea	HoC-2014
22 centros	14 centros	23 centros	29 centros

5.3.1.2. *Muestra de profesores*

Tal y como se detalla en la siguiente Tabla 5.5, un total de 23 profesores (N=23) procedentes de 22 centros educativos distintos (se recogieron dos respuestas del CEIP Escola Pere Torrent procedentes de 2 profesores distintos, ID_17 e ID _23, ambos coordinadores del evento HoC en el centro) contestaron al cuestionario *pre*.

 Tabla 5.5. Relación de sujetos que contestaron al cuestionario *pre* para profesores

Edición HoC	ID Sujeto ²⁷⁸	Nombre y apellidos	Centro Educativo	Provincia	Titularidad	Etapas educativas implantadas en el Centro ²⁷⁹ :
HoC 2013	1	John Asensi	American School of Bilbao	Vizcaya	Privado	I, P, S, B
	2	Lidia Stabile	IES San Antonio de Benagéber	Valencia	Público	S, B
	3	Antonio Coloma	IES L'Almadrava	Alicante	Público	S, B, FP _M , FP _S
	4	Samuel Landete	American School of Valencia	Valencia	Privado	I, P, S, B
	5	José A. Gómez	IES Francisco Romero Vargas	Cádiz	Público	S, B, FP _M , FP _S
	6	Lisa Howells	Benjamin Franklin International School	Barcelona	Privado	I, P, S, B
	7	Alan MacLean	(The) American School of Madrid	Madrid	Privado	I, P, S, B
	8	Jordi Losantos	Aula Escola Europea	Barcelona	Privado	I, P, S, B
	9	Joaquín López	IES Penyagolosa	Castellón	Público	S, B
	10	Laura Carrascal	CP Cabueñes	Asturias	Público	I, P
HoC 2014	11	Jorge Lobo	CEIP Lope de Vega	Madrid	Público	I, P
	12	Miguel Señor	IES Abyla	Ceuta	Público	S, B, FP _M , FP _S
	13	Marta Figueras	CEIP Escola Antoni Brusi	Barcelona	Público	I, P
	14	Carlos Campos	IES Ximén d'Urrea	Castellón	Público	S, B, FP _M , FP _S
	15	Elsa García	CEIP Laurisilva	Las Palmas	Público	I, P
	16	M ^a Rosa Ayats	CEIP Escola Agustí Gifre	Gerona	Público	I, P
	17	Carlos Martín	CEIP Escola Pere Torrent	Gerona	Público	I, P
	18	Paul F. Love	English School Los Olivos	Valencia	Privado	I, P, S, B
	19	Iván Alfaro	Colegio Educrea	Madrid	Concertado	I, P, S, B
	20	Arcadio García	IES Ramiro de Maeztu	Madrid	Público	S, B
	21	Sandra Gómez	Humanitas Bilingual School	Madrid	Concertado	I, P, S
	22	Susana Oubiña	IES Castro Alobre	Pontevedra	Público	S, B, FP _M
	23	Gerard Gutierrez	CEIP Escola Pere Torrent	Gerona	Público	I, P

De estos 23 profesores: 10 profesores (43,5%) corresponden a la edición HoC-2013, y 13 profesores (56,5%) a la edición HoC-2014; 15 son hombres (65,2%) y 8 son mujeres (34,8%). Con respecto a las características de los centros educativos de procedencia de dichos 23 profesores: 12 profesores (52,2%) contestan desde el área geográfica 'Este y Baleares' (6 desde Cataluña y 6 desde la Comunidad Valenciana), 5 profesores (21,7%) provienen del área 'Centro' (todos ellos de la Comunidad de Madrid), 3 profesores (13,0%) provienen del área 'Noroeste' (1 de Asturias, 1 de Galicia y 1 del País Vasco), y los 3 profesores restantes (13,0%) lo hacen desde el área 'Sur y Canarias' (1 de Andalucía, 1 de Ceuta y 1 de las Islas Canarias). Además, 15 profesores (65,2%)

²⁷⁸ Los ID se asignaron por orden cronológico de llegada de las respuestas.

²⁷⁹ I: Infantil; P: Primaria; S: Secundaria; B: Bachillerato; FP_M: Formación Profesional de Grado Medio; FP_S: Formación Profesional de Grado Superior.

contestan desde centros públicos, 6 (26,1%) desde centros privados, y los 2 restantes (8,7%) desde centros concertados. Finalmente, señalar que 8 profesores (34,8%) lo son de IES (Institutos Públicos de Educación Secundaria, Bachillerato y/o FP), 8 profesores (34,8%) lo son de centros ‘Multietapa’ (Privados o Concertados), y los 7 restantes (30,4%) son profesores, más exactamente maestros, en CEIP (Colegios Públicos de Educación Infantil y/o Primaria).

De entre los 23 profesores que contestaron al cuestionario *pre*, un total de 14 profesores (N=14) procedentes de 14 centros educativos distintos contestaron al cuestionario *post* (lo cual supone un 39,13% de mortalidad experimental). Concretamente, los 14 sujetos que contestaron a ambos cuestionarios son los ID: 1, 2, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 16, y 23.

5.3.1.3. Muestra de estudiantes

En la Tabla 5.6 se muestra el número de estudiantes registrados²⁸⁰ por cada edición del evento ‘La Hora del Código’, así como la muestra finalmente productora de datos válidos a través del ‘Cuestionario para estudiantes’.

Tabla 5.6. Número de estudiantes registrados y finalmente productores de datos por edición

	Estudiantes registrados	Muestra válida de estudiantes	% Válidos / Registrados
Hora del Código 2013	11.340	335	2,95 %
Hora del Código 2014	57.880	1.327	2,29 %
Total Acumulado	69.220	1.662	2,40 %

Tal y como se detalla en la anterior Tabla 5.6, a dicho cuestionario contestaron un total de 1.662 estudiantes (N=1.662); de los cuales 335 (20,2%) fueron recogidos en la edición HoC-2013, y 1.327 (79,8%) en la edición HoC-2014.

Con respecto a la distribución por sexo del conjunto de la muestra, se detalla en la Tabla 5.7. Como puede observarse, la muestra está bien balanceada por sexos (52,6% de chicos frente a 47,4% de chicas), indicativo de que la celebración del evento HoC fue transversal a todo tipo de aulas (y no sólo a las aulas de la asignatura optativa de ‘Informática’ o similar, en la que, como veremos en los Capítulos 6 y 7, la proporción aproximada por género que aparece es de 66%-33% a favor de los chicos).

Tabla 5.7. Distribución por sexo de la muestra de estudiantes

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Chico	846	50,9	52,6	52,6
	Chica	761	45,8	47,4	100,0
	Total	1607	96,7	100,0	
Perdidos	Sistema	55	3,3		
Total		1662	100,0		

²⁸⁰ Estrictamente los que se registran en el evento son los centros, no los estudiantes. Pero, al registrarse, los centros indican de manera orientativa qué número de sus estudiantes participarán en el evento; dicho número es el que aquí tomamos en consideración.

Con respecto a la distribución por edad del conjunto de la muestra, se detalla en las Tablas 5.8 y 5.9, y en el histograma de la Figura 5.1. Es una distribución no normal ($Z_{k-s} = 0,110$; $p_{(Z_{k-s})} = 0,000 \ll 0,05$), con media $\bar{X} = 13,01$ años, y desviación típica $s = 2,337$. Aunque el rango es muy amplio, abarcando de 8 a 20 años, alrededor del 80% de la muestra se concentra en el intervalo de entre 10 y 15 años (último ciclo de Educación Primaria y Educación Secundaria Obligatoria), que será donde focalizaremos precisamente el segundo y el tercer estudio empírico de esta tesis doctoral (Capítulos 6 y 7)

Tabla 5.8. Estadísticos descriptivos relativos a la edad de la muestra de estudiantes

N	Válido	1637
	Perdidos	25
Media		13,01
Mediana		13,00
Moda		12
Desviación estándar		2,337
Varianza		5,460
Asimetría		,193
Error estándar de asimetría		,060
Curtosis		-,540
Error estándar de curtosis		,121
Mínimo		8
Máximo		20

Tabla 5.9. Distribución por edad de la muestra de estudiantes

		Edad (en años)			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	8 ó menos	18	1,1	1,1	1,1
	9	53	3,2	3,2	4,3
	10	203	12,2	12,4	16,7
	11	214	12,9	13,1	29,8
	12	238	14,3	14,5	44,3
	13	224	13,5	13,7	58,0
	14	204	12,3	12,5	70,5
	15	231	13,9	14,1	84,6
	16	137	8,2	8,4	93,0
	17	90	5,4	5,5	98,5
	18	13	,8	,8	99,3
	20 ó más	12	,7	,7	100,0
	Total	1637	98,5	100,0	
	Perdidos	Sistema	25	1,5	
Total		1662	100,0		

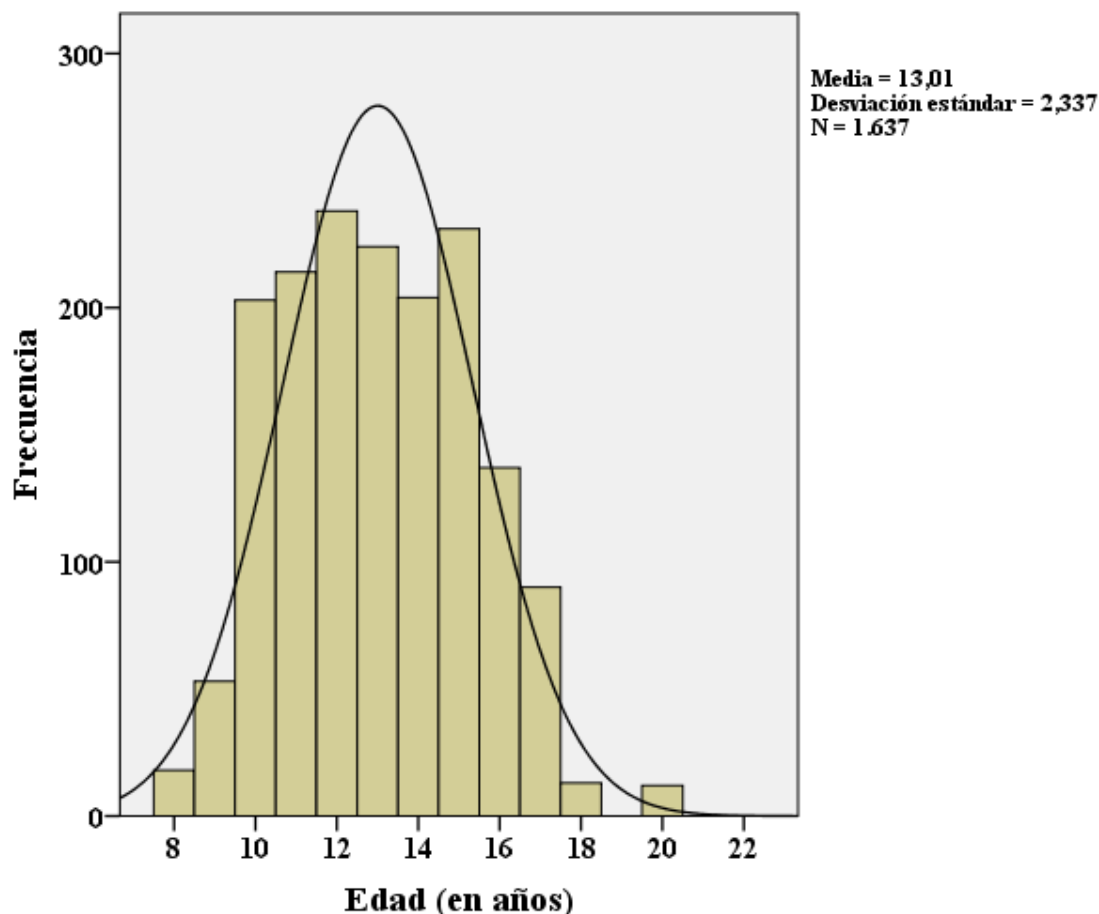


Figura 5.1. Histograma con la distribución por edad de la muestra de estudiantes

Con respecto a la distribución de la muestra de estudiantes en función de la ‘etapa educativa’ que están cursando, se detalla en la Tabla 5.10. Tal y como puede apreciarse, el centro de gravedad de la muestra está en Educación Secundaria (con prolongaciones hacia el último ciclo de Primaria).

Tabla 5.10. Distribución de la muestra de estudiantes según etapa educativa

		Etapa Educativa			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Primaria	505	30,4	30,7	30,7
	Secundaria	966	58,1	58,8	89,5
	Bachillerato	157	9,4	9,6	99,1
	Formación Profesional	15	,9	,9	100,0
	Total	1643	98,9	100,0	
Perdidos	Sistema	19	1,1		
Total		1662	100,0		

Tal y como puede verse en la Tabla 5.11 y Figura 5.2, la muestra de estudiantes proviene de 23 centros educativos distintos. Cada centro aporta, como valor promedio, alrededor de 70 estudiantes a la investigación. Por encima de este valor promedio, destacan las contribuciones de los siguientes 8 centros: Benjamin Franklin International School, CEIP La Jota, CEIP Lope de Vega, Colegio Mirasur, Colegio San Jorge, IES Castro Alobre, IES María Blasco, e IES San Antonio de Benagéber.

Tabla 5.11. Centros educativos de procedencia de la muestra de estudiantes

Centro Educativo		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	American School of Bilbao	45	2,7	2,8	2,8
	(The) American School of Madrid	1	,1	,1	2,8
	American School of Valencia	39	2,3	2,4	5,2
	Benjamin Franklin International School	117	7,0	7,2	12,4
	Casa de Juventud de Las Rozas	24	1,4	1,5	13,9
	CEIP Escola Agustí Gifre	3	,2	,2	14,1
	CEIP Escola Pere Torrent	10	,6	,6	14,7
	CEIP La Jota	111	6,7	6,8	21,5
	CEIP Laurisilva	18	1,1	1,1	22,6
	CEIP Lope de Vega	125	7,5	7,7	30,3
	Colegio Educrea	1	,1	,1	30,3
	Colegio Mirasur	249	15,0	15,3	45,6
	Colegio San Agustín	21	1,3	1,3	46,9
	Colegio San Jorge	98	5,9	6,0	52,9
	English School Los Olivos	40	2,4	2,5	55,4
	IES Abyla	45	2,7	2,8	58,2
	IES Castro Aobre	79	4,8	4,9	63,0
	IES Fuente de San Luis	25	1,5	1,5	64,6
	IES María Blasco	106	6,4	6,5	71,1
	IES Penyagolosa	15	,9	,9	72,0
	IES Ramiro de Maeztu	17	1,0	1,0	73,0
	IES San Antonio de Benagéber	431	25,9	26,5	99,5
	IES Ximén d'Urrea	8	,5	,5	100,0
Total	1628	98,0	100,0		
Perdidos	Sistema	34	2,0		
Total		1662	100,0		

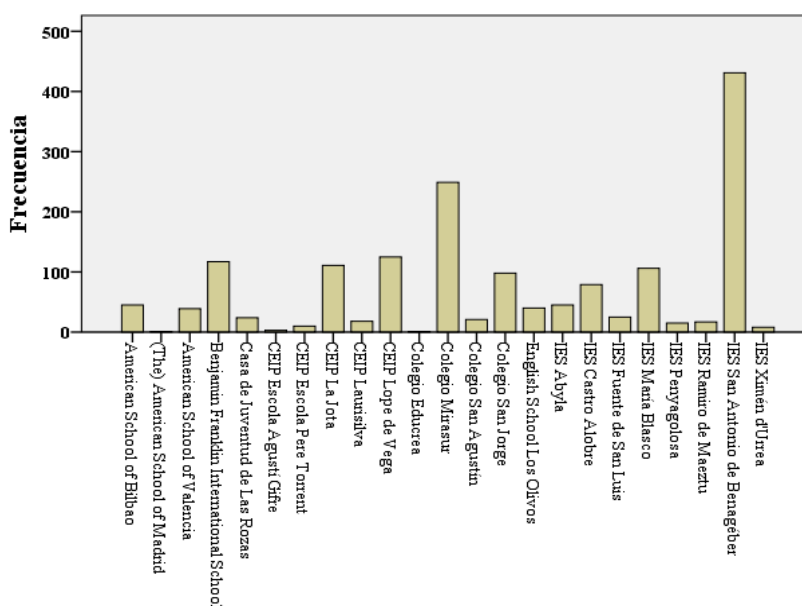


Figura 5.2. Centros educativos de procedencia de la muestra de estudiantes

Respecto a la distribución de los estudiantes en función del ‘Área Geográfica’, ‘Comunidad Autónoma’, ‘Titularidad’ y ‘Tipología’ de sus respectivos centros, se detalla en las Tablas 5.12 a 5.15. Tal y como puede observarse, casi el 75% de la muestra de estudiantes proviene de la Comunidad Valenciana y Cataluña (área Este), o de la Comunidad de Madrid (área Centro); reflejo del desequilibrio territorial que veremos al analizar el conjunto de centros invitados a la investigación. En la muestra hay representación suficiente de estudiantes procedentes de centros públicos (tanto de CEIP como de IES) y de centros privados-concertados (Multietapa).

Tabla 5.12. Distribución de la muestra de estudiantes por área geográfica

		Área Geográfica			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Este y Baleares	1003	60,3	61,6	61,6
	Centro	417	25,1	25,6	87,2
	Noroeste	145	8,7	8,9	96,1
	Sur y Canarias	63	3,8	3,9	100,0
	Total	1628	98,0	100,0	
Perdidos	Sistema	34	2,0		
Total		1662	100,0		

Tabla 5.13. Distribución de la muestra de estudiantes por Comunidad Autónoma

		Comunidad Autónoma			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Comunidad Valenciana	664	40,0	40,8	40,8
	Comunidad de Madrid	417	25,1	25,6	66,4
	Cataluña	130	7,8	8,0	74,4
	Aragón	111	6,7	6,8	81,2
	Región de Murcia	98	5,9	6,0	87,2
	Galicia	79	4,8	4,9	92,1
	Ceuta	45	2,7	2,8	94,8
	País Vasco	45	2,7	2,8	97,6
	Castilla y León	21	1,3	1,3	98,9
	Canarias	18	1,1	1,1	100,0
	Total	1628	98,0	100,0	
Perdidos	Sistema	34	2,0		
Total		1662	100,0		

Tabla 5.14. Distribución de la muestra de estudiantes por titularidad de su centro

		Titularidad			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Público	1017	61,2	62,5	62,5
	Privado	589	35,4	36,2	98,6
	Concertado	22	1,3	1,4	100,0
	Total	1628	98,0	100,0	
Perdidos	Sistema	34	2,0		
Total		1662	100,0		

Tabla 5.15. Distribución de la muestra de estudiantes por tipología de su centro

Tipología		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	IES – Inst. Públ. Secundaria, Bachiller. y/o FP	726	43,7	44,6	44,6
	Centro Privado o Concertado Multietapa	611	36,8	37,5	82,1
	CEIP - Colegio Público Infantil y/o Primaria	267	16,1	16,4	98,5
	Centro Municipal de Educación No Formal	24	1,4	1,5	100,0
	Total	1628	98,0	100,0	
Perdidos	Sistema	34	2,0		
Total		1662	100,0		

5.3.2. Instrumentos

Para el presente estudio empírico, se han diseñado y aplicado los siguientes 3 instrumentos (todos ellos en lengua española e inglesa, dado que algunos de los centros participantes son bilingües): cuestionario *pre* para profesores; cuestionario *post* para profesores; y cuestionario para estudiantes. Los vemos con detalle en los siguientes sub-epígrafes.

5.3.2.1. Cuestionario pre para profesores

El “CSEW²⁸¹: Cuestionario para profesores/as (previo)” es un cuestionario que se aplica ‘on-line’ a través de un formulario diseñado en Google Drive²⁸². Está dirigido al profesor en cada centro participante responsable de la coordinación del evento ‘La Hora del Código’ y es cumplimentado antes del inicio del mismo, habitualmente a lo largo de la semana previa (Figura 5.3)

Este cuestionario se construye según las siguientes directrices:

- Dada la novedad del tema de investigación y el leve contacto que se tiene con muchos de los profesores (en muchas ocasiones se limita a un cruce previo de correos electrónicos), al comienzo del cuestionario se explicita la finalidad general de la investigación para buscar la concienciación y complicidad del profesor. A continuación, se presentan unas breves instrucciones y se solicitan datos personales básicos al encuestado.
- En una segunda página se presentan 10 preguntas abiertas, dado el carácter inicial y exploratorio de este cuestionario, que se redactaron en consonancia con los objetivos específicos enunciados más arriba:
 - Preguntas 1...3 para cubrir respectivamente los objetivos O_{2Pa}...O_{2Pc}
 - Preguntas 4...7 para cubrir respectivamente los objetivos O_{2Pd}...O_{2Pg}
 - Preguntas 8...9 para cubrir respectivamente los objetivos O_{2Ph}...O_{2Pi}
 - Pregunta 10 final de síntesis descriptiva para cubrir el objetivo O_{2Pj}

²⁸¹ Recordamos que CSEW son las siglas de ‘Computer Science Education Week’, semana a lo largo de la cual se celebra el evento ‘La Hora del Código’ (HoC)

²⁸² <http://goo.gl/xCRO26>



Figura 5.3. Código QR que enlaza al ‘CSEW: Cuestionario para profesores (previo)’ (Román-González, 2013d)

En el **Anexo A** se reproduce igualmente el contenido exacto de este cuestionario, para su consulta detallada por parte del lector interesado.

5.3.2.2. Cuestionario post para profesores

El “*CSEW: Cuestionario para profesores/as (post)*” es igualmente un cuestionario que se aplica ‘*on-line*’ a través de un formulario diseñado en Google Drive²⁸³. Está dirigido al profesor en cada centro participante responsable de la coordinación del evento ‘La Hora del Código’ y es cumplimentado después de la finalización del mismo, normalmente en la semana siguiente a su celebración (Figura 5.4). Este cuestionario consta de 5 preguntas abiertas (**Preguntas 1...5**), que se redactaron consistentemente para cubrir los objetivos específicos enunciados más arriba (**O_{2Pk}...O_{2Pñ}**). En el **Anexo B** se reproduce el contenido completo de este cuestionario.



Figura 5.4. Código QR que enlaza al ‘CSEW: Cuestionario para profesores (post)’ (Román-González, 2013e)

5.3.2.3. Cuestionario para estudiantes

El “*CSEW: Cuestionario para estudiantes*” es igualmente un cuestionario que se aplica ‘*on-line*’ a través de un formulario diseñado en Google Drive²⁸⁴. Está dirigido a los estudiantes participantes en el evento ‘La Hora del Código’ y es cumplimentado justo tras finalizar la actividad de introducción a la programación, habitualmente en los últimos 5 minutos de la hora de clase en la que se celebra (Figura 5.5)

Este cuestionario consta de un total de 12 preguntas, siendo una combinación de preguntas de elección múltiple, de selección múltiple (casillas de verificación), de escalas tipo Likert, y de preguntas abiertas. Las preguntas se redactaron consistentemente para cubrir los objetivos específicos enunciados más arriba:

- **Preguntas 1...3** para cubrir respectivamente los objetivos **O_{2Ea}...O_{2Ec}**
- **Pregunta 4** para cubrir el objetivo **O_{2Ed}**

²⁸³ <https://goo.gl/RRV9E5>

²⁸⁴ <https://goo.gl/c3j5xq>

- **Preguntas 5...10** para cubrir respectivamente los objetivos $O_{2Ee} \dots O_{2Ej}$
- **Preguntas 11...12** para cubrir respectivamente los objetivos $O_{2Ek} \dots O_{2EI}$



Figura 5.5. Código QR que enlaza al ‘CSEW: Cuestionario para estudiantes’ (Román-González, 2013f)

En el **Anexo C** se reproduce el contenido completo de este cuestionario, para su consulta detallada por parte del lector interesado.

5.3.3. Procedimiento

El procedimiento seguido para este estudio empírico se sintetiza en los siguientes pasos:

- Se accede al listado de ‘centros educativos registrados’ en el evento HoC²⁸⁵, en las ediciones de diciembre de 2013 y diciembre de 2014.
- Por un procedimiento de muestreo aleatorio ‘por lista’, se selecciona un subconjunto de dichos ‘centros registrados’, a los que se invita a participar en la investigación (‘centros invitados’)
- Se establece contacto con cada ‘centro invitado’ a través de un correo electrónico dirigido al coordinador TIC del mismo, habitualmente responsable del desarrollo del evento HoC.
- Del conjunto de ‘centros invitados’, sólo un subconjunto responde afirmativamente a participar en la investigación: son los ‘centros aceptantes’.
- Al responsable del evento HoC en cada ‘centro aceptante’ se le remite con anterioridad al mismo: el cuestionario *pre* para profesores (para ser cumplimentado por dicho profesor responsable en la semana previa al evento); y el cuestionario para estudiantes (para ser cumplimentado por los alumnos justo al finalizar la actividad HoC, en los últimos 5 minutos de la clase)
- En la semana posterior al evento HoC, se remite a cada ‘centro aceptante’ el cuestionario *post* para profesores, para ser cumplimentado por la misma persona que el *pre*.
- Una vez finalizada la recogida de cuestionarios descargamos las respuestas desde nuestras bases de datos asociadas en *Google Drive*. Los datos cualitativos (respuestas a las preguntas abiertas) se vuelcan sobre archivos de texto de extensión *.rtf*, y se analizan con el software de análisis cualitativo Atlas.ti²⁸⁶ versión 7. Los datos cuantitativos son exportados, y depurados, en un archivo de extensión *.sav*, y se analizan con el software de análisis cuantitativo SPSS²⁸⁷ versión 22.

²⁸⁵ <https://hourofcode.com/es/events/all/es>

²⁸⁶ <http://atlasti.com/es/>

²⁸⁷ <http://www-01.ibm.com/software/es/analytics/spss/>

5.4. Resultados

Dado el gran volumen de resultados obtenidos, se presentarán en el cuerpo del texto sólo las figuras y tablas principales. Las figuras y tablas complementarias se trasladan y presentan en el [Anexo D](#).

5.4.1. Relativas a las características de los centros

Dentro del objetivo general O_1 , el primer resultado destacable es que, tal y como vimos algo más arriba en la Tabla 5.1, el número de centros registrados en España en el evento de ‘La Hora del Código’ se incrementó desde 96 centros en 2013 a 490 centros en 2014; es decir, el número de centros registrados se multiplicó por algo más de 5 en tan solo un año.²⁸⁸

Más concretamente, y para dar respuesta a los objetivos específicos $O_{1a} \dots O_{1c}$, en la medida que los centros educativos invitados a la investigación fueron muestreados de manera aleatoria (‘por lista’) del total de la población (= total de centros educativos registrados); entonces, un análisis de las características de los 77 centros educativos invitados será representativa del conjunto, sirviendo así para una descripción general del fenómeno. De los 77 centros invitados, 23 corresponden a la edición HoC-2013 y 54 a la edición HoC-2014.

En la Tabla 5.16 (y Figura D.1 del anexo) se detalla la distribución por área geográfica²⁸⁹ de procedencia de los 77 centros educativos invitados:

Tabla 5.16. Área geográfica de procedencia de los centros invitados a la investigación

	Frecuencia	Porcentaje	Porcentaje acumulado
Este y Baleares	34	44,2	44,2
Noroeste	20	26,0	70,1
Centro	15	19,5	89,6
Sur y Canarias	8	10,4	100,0
Total	77	100,0	

Acercando más el foco, en la Tabla 5.17 (y Figura D.2 del anexo) se detalla la Comunidad Autónoma de procedencia de los 77 centros educativos invitados a la investigación:

Tabla 5.17. Comunidad Autónoma de procedencia de los centros educativos invitados a la investigación

	Frecuencia	Porcentaje	Porcentaje acumulado
Cataluña	19	24,7	24,7
Comunidad de Madrid	14	18,2	42,9
Comunidad Valenciana	12	15,6	58,4
Galicia	7	9,1	67,5

²⁸⁸ En las semanas previas al depósito de esta tesis doctoral, hemos tenido noticia del número de centros registrados en España en la edición HoC-2015, que ha ascendido a 1581 centros. Ello supone que el número de centros registrados en nuestro país se ha multiplicado por más de 15 veces en tan sólo 2 años (de diciembre de 2013 a diciembre de 2015)

²⁸⁹ Se han codificado las siguientes áreas geográficas: Este y Baleares (incluye Aragón, Cataluña, Comunidad Valenciana, Región de Murcia e Islas Baleares); Noroeste (incluye Galicia, Asturias, Cantabria, País Vasco, Navarra, La Rioja y Castilla y León); Centro (Comunidad de Madrid y Castilla La Mancha); y Sur y Canarias (Extremadura, Andalucía, Ceuta, Melilla e Islas Canarias)

	Frecuencia	Porcentaje	Porcentaje acumulado
Castilla y León	6	7,8	75,3
Andalucía	5	6,5	81,8
Asturias	4	5,2	87,0
Aragón	2	2,6	89,6
Canarias	2	2,6	92,2
País Vasco	2	2,6	94,8
Cantabria	1	1,3	96,1
Castilla-La Mancha	1	1,3	97,4
Ceuta	1	1,3	98,7
Región de Murcia	1	1,3	100,0
Total	77	100,0	

Analizamos a continuación si las anteriores distribuciones han sufrido una variación significativa entre la edición HoC-2013 y HoC-2014. Así, en la Figura 5.6 (y Tabla D.1 del anexo) se presenta la distribución por área geográfica de los centros invitados, diferenciada para ambas ediciones. Aplicando una prueba de Chi-Cuadrado (y Coeficiente de Contingencia), se encuentra una variación no significativa de la distribución por área geográfica entre las dos ediciones ($\chi^2 = 3,543$; $p_{(\chi^2)} = 0,315 > 0,05$; $C = 0,210$; $p_{(C)} = 0,315 > 0,05$).

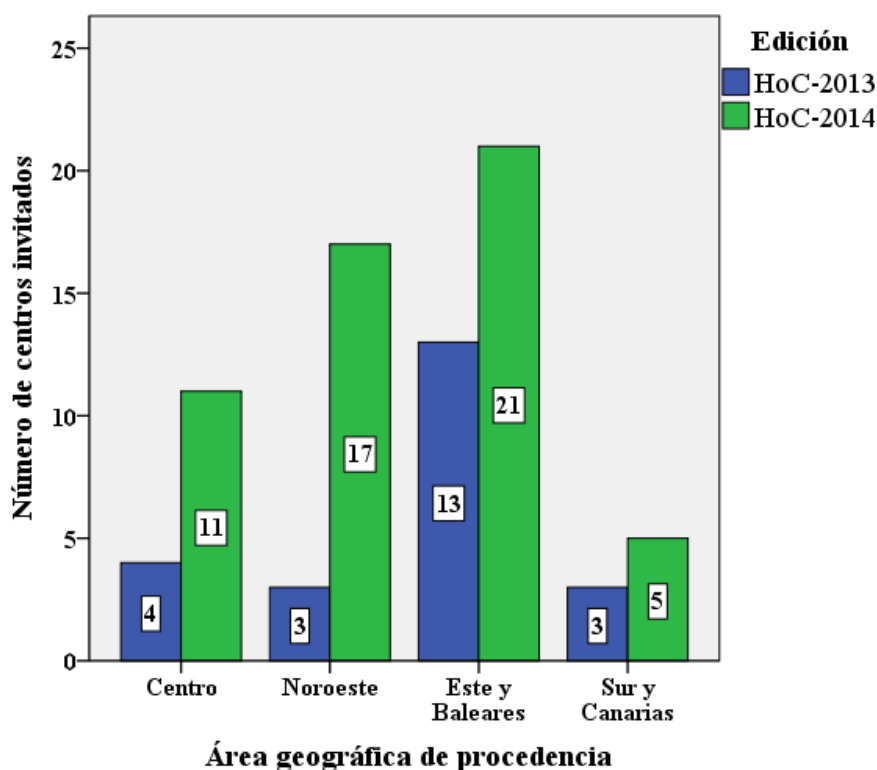


Figura 5.6. Distribución de los centros invitados por área geográfica, diferenciada para ambas ediciones

Hacemos lo propio para la Comunidad Autónoma de procedencia de los centros invitados en la Figura 5.7 (y Tabla D.2 del anexo), encontrándose igualmente una variación no significativa de la distribución entre las dos ediciones ($\chi^2 = 18,626$; $p_{(\chi^2)} = 0,135 > 0,05$; $C = 0,441$; $p_{(C)} = 0,135 > 0,05$).

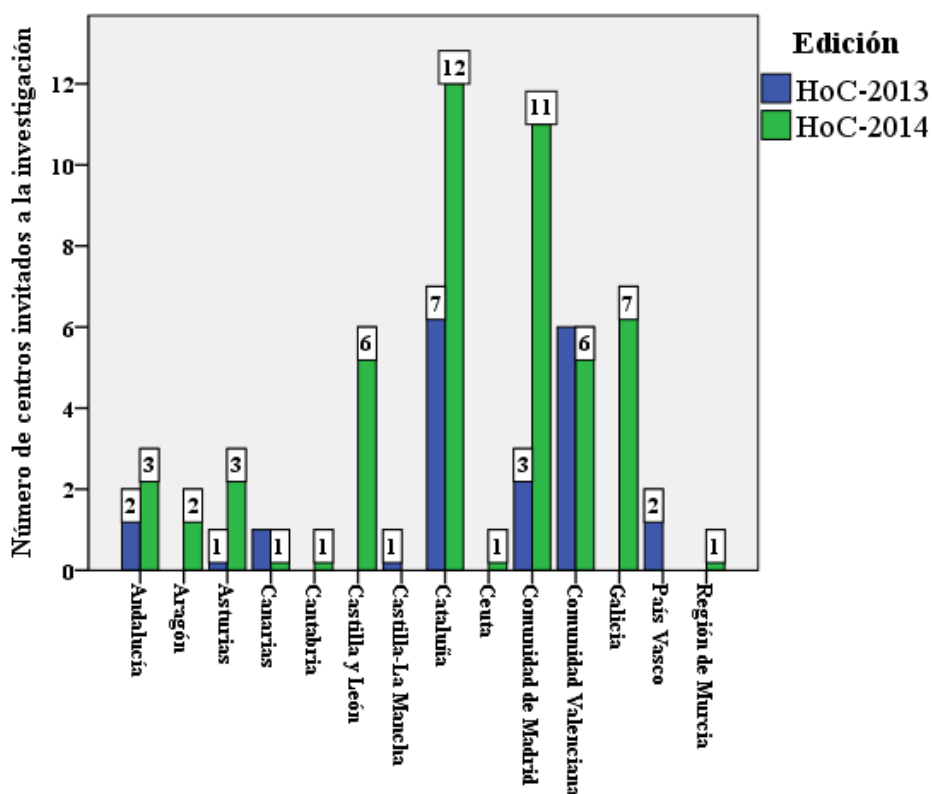


Figura 5.7. Distribución de los centros invitados por Comunidad Autónoma, diferenciada para ambas ediciones

Con respecto a la titularidad de los centros educativos invitados a la investigación (titularidad pública, concertada, o privada), se muestra en la Tabla 5.18 (y Figura D.3 del anexo). Encontramos en este caso una variación, en el límite de la significatividad, de la distribución de centros por titularidad a lo largo de ambas ediciones ($\chi^2 = 5,857$; $p_{(\chi^2)} = 0,053 \approx 0,05$; $C = 0,266$; $p_{(C)} = 0,053 \approx 0,05$). En la Figura 5.8 (y Tabla D.3 del anexo) se muestra dicha variación.

Tabla 5.18. Titularidad de los centros educativos invitados a la investigación

	Frecuencia	Porcentaje	Porcentaje acumulado
Público	46	59,7	59,7
Privado	17	22,1	81,8
Concertado	14	18,2	100,0
Total	77	100,0	

Finalmente, en cuanto a la tipología²⁹⁰ de los centros invitados, se presenta en la Tabla 5.19 (y la Figura D.4 del anexo). Encontramos, aquí sí claramente, una variación significativa de la distribución de centros por tipología a lo largo de ambas ediciones ($\chi^2 = 10,343$; $p_{(\chi^2)} = 0,016 < 0,05$; $C = 0,344$; $p_{(C)} = 0,016 < 0,05$). En la Figura 5.9 (y Tabla D.4 del anexo) queda ilustrada de manera evidente dicha variación.

²⁹⁰ Por tipología entendemos si el centro educativo pertenece a alguna de las siguientes categorías (exhaustivas y mutuamente excluyentes): CEIP (Colegio Público de Educación Infantil y/o Primaria); IES (Instituto Público de Educación Secundarias y/o Bachillerato y/o Formación Profesional); Centro Privado o Concertado Multietapa (en el mismo centro se imparten, al menos, las etapas de Primaria y Secundaria); Centro de Educación No Formal (p.e. Casas de Juventud, Centros de Día, Asociaciones...)

Tabla 5.19. Tipología de los centros educativos invitados a la investigación.

	Frecuencia	Porcentaje	Porcentaje acumulado
Centro Privado o Concertado Multietapa	31	40,3	40,3
IES - Instituto Público Secundaria, Bachillerato y/o FP	24	31,2	71,4
CEIP - Colegio Público Infantil y/o Primaria	21	27,3	98,7
Centro Municipal de Educación No Formal	1	1,3	100,0
Total	77	100,0	

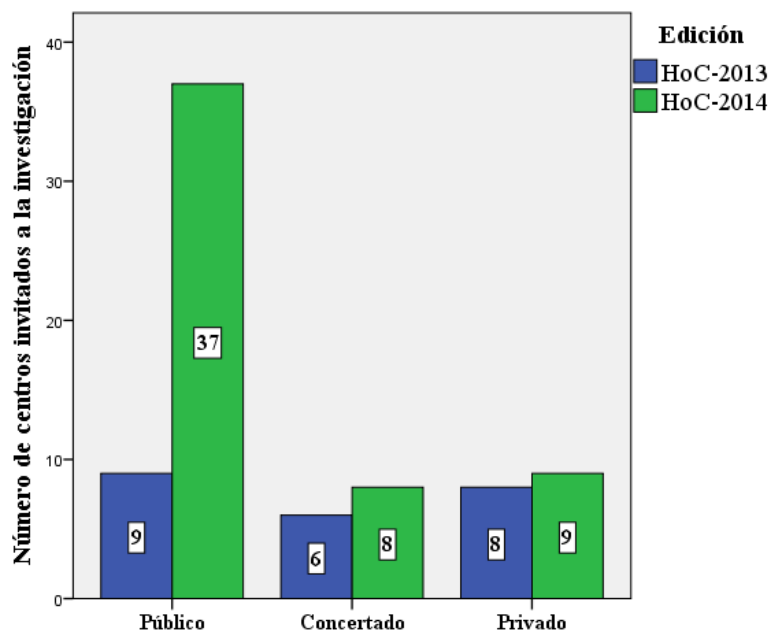


Figura 5.8. Distribución de los centros invitados por titularidad, diferenciado por cada edición HoC

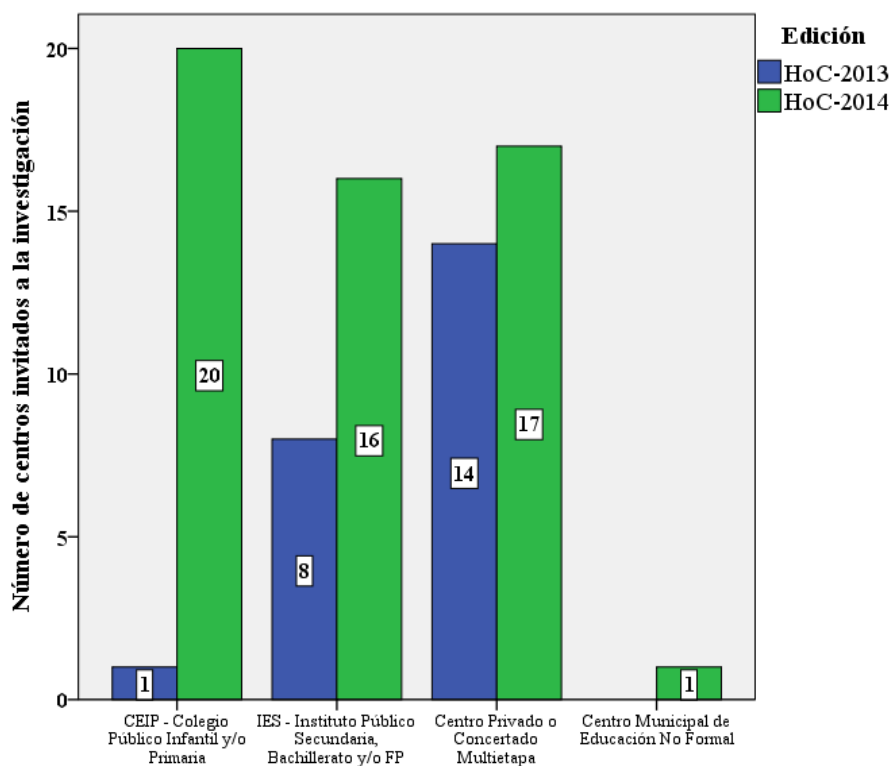


Figura 5.9. Distribución de los centros invitados por tipología, diferenciado por cada edición HoC

A modo de resumen, en la Tabla 5.20 se muestran sintéticamente las pruebas estadísticas aplicadas para comprobar la significatividad de las variaciones sufridas, en las distintas variables consideradas, por la distribución de centros invitados a lo largo de las dos ediciones HoC. No se encuentra significatividad en las variables ‘Área geográfica’ y ‘Comunidad Autónoma’ de procedencia del centro invitado; en el límite de la significatividad para la variable ‘Titularidad’; y sí claramente significativa la variación en la distribución de centros por ‘Tipología’ a lo largo de ambas ediciones.

Tabla 5.20. Resumen de significatividad de las variaciones en la distribución de centros invitados a la investigación, a lo largo de las ediciones HoC

	Grados de libertad (gl)	Chi-Cuadrado (χ^2)	Signific. $P(\chi^2)$	Coefficiente de Contingencia (C)	Signific. $P(C)$	¿Signific. a un nivel de confianza del 95%?
Área Geogr.	3	3,543	0,315	0,210	0,315	No
CC.AA.	13	18,626	0,135	0,441	0,135	No
Titularidad	2	5,857	0,053	0,266	0,053	En el límite
Tipología	3	10,343	0,016	0,344	0,016	Sí

5.4.2. Relativos al cuestionario *pre* para profesores

Pasamos ahora a dar respuesta a los objetivos $O_{2Pa} \dots O_{2Pj}$, que se abordan a través del “CSEW: Cuestionario para profesores/as (previo)” (Anexo A).

A través de la Figura 5.10 se puede acceder a un documento²⁹¹ con el contenido literal y completo de las respuestas de los 23 profesores a las 10 preguntas abiertas que componen el “CSEW: Cuestionario para profesores/as (previo)” (más la ‘Pregunta 0’ inicial de control), y que cubren respectivamente los objetivos $O_{2Pa} \dots O_{2Pj}$ de la investigación.



Figura 5.10. Código QR al documento con el contenido completo de las respuestas al cuestionario *pre* para profesores

En lo relativo a la **Pregunta 0** del cuestionario [“Cargo/s que ocupa usted en el Centro. Indique, además de su docencia, si tiene alguna otra responsabilidad en su centro (p.ej. Coordinador TIC, Coordinador de Etapa, Jefe de Departamento, Dirección, etc...)”]: la respuesta típica de los sujetos que ejercen en Secundaria es la de ser, además de profesores, jefes del Departamento de Informática y Coordinadores TIC de sus respectivos centros, especialistas encargados de la integración de la tecnología en el mismo (“*Technology Integration Specialist*” en palabras del sujeto ID_7) y de la formación continua del profesorado en este aspecto (“...*coordinating ICT across the campus and providing professional development for teachers*” en palabras del sujeto ID_6). En el caso de los sujetos que ejercen en Primaria, el perfil es más variado: además de ser muchos Coordinadores TIC,

²⁹¹ Disponible en <https://goo.gl/5Kxy2x>

hay quien es “*Tutora 4º y Especialista de Música*” (ID_13), o afirma “*imparto Matemáticas, Conocimiento del Medio en 6º y TIC’s en 6º y 5º*” (ID_15), o “*Maestra Inglés y maestra informática 3º y 4º (de Primaria)*” (ID_16). Además, hay 2 sujetos excepcionales que no son estrictamente personal docente de su respectivo centro educativo, pero que fueron los coordinadores del evento ‘La Hora del Código’ en el mismo: uno es “*antiguo alumno del centro*” (ID_20) y otro sujeto (ID_21) relata “*yo no trabajo en el centro. Mi hija estudia en el centro. Yo fui quien le presentó al centro la iniciativa del ‘Hour of Code’ y de forma voluntaria daré el día 9 ‘The Hour of Code’ a los alumnos de primaria y el día 15 lo daré a alumnos de la ESO*”; lo cual sugiere una buena interpenetración y permeabilidad del fenómeno entre los colectivos familia-escuela.

En lo relativo a la **Pregunta 1** del cuestionario [“*¿Qué aptitudes y habilidades considera usted que desarrolla el aprendizaje de la programación informática (‘coding’)?*”], que cubre el objetivo de investigación **O_{2Pa}**: las respuestas más frecuentes se centran alrededor del pensamiento lógico, en sus distintas formulaciones (“*pensamiento analítico, lógico, matemático*”... “*desarrolla su capacidad lógica*”... “*logical skills*”... “*razonamiento*”... “*desarrollo del pensamiento lógico-matemático*”... “*pensamiento abstracto*”... “*promover el pensamiento crítico y analítico del alumnado*”... “*creo que desarrolla el pensamiento abstracto*”... “*the ability to think linearly*”... “*pensamiento analítico*”... “*desarrolla el pensamiento lógico, estructurado*”), y con aplicaciones sobre conceptos matemáticos (“*deepens understanding of certain mathematical concepts, sequencing and organizing information*”); e igualmente se repiten con frecuencia respuestas alrededor de la resolución de problemas, también a través de distintas formulaciones (“*les estructura la forma de pensar, les enseña a pensar y aumenta sus habilidades para solución de problemas*”... “*habilidades en la resolución de problemas*”... “*planificación, evaluar resultados*”... “*precisión en la elaboración de una solución; pensamiento estratégico; autonomía en la elaboración y autocorrección de la solución alcanzada*”... “*división de problemas*”... “*resolución de problemas; organización de ideas*”... “*nos ayuda a ordenar, pautar y pensar cómo conseguir llegar a un fin, o problema o reto*”... “*para planificar diferentes tareas y conseguir objetivos y metas previstas*”... “*the ability to break a large problem down into smaller ones*”... “*permite poner en forma de secuencia una tarea; es decir ejercita el paso de un pensamiento global, problema a resolver a buscar una secuencia ordenada que te permite resolver la tarea*”... “*aprenden a buscar soluciones alternativas (Think out of the box)²⁹²*”)

Aparecen también respuestas, aunque en menor medida, que afirman que la programación informática o ‘coding’ desarrolla, por ejemplo: aptitudes perceptivas espacio-temporales (“*la distribución espacial*”... “*la inteligencia espacial*”... “*la organización espacial, la organización temporal*”); aptitudes atencionales-motivacionales (“*concentración*”... “*autoaprendizaje y autonomía*”... “*mejora de la autonomía e iniciativa personal*”... “*considerar el error y la copia como motor de aprendizaje*”); creatividad (“*creatividad*”... “*estética, expresión artística*”... “*desarrolla la creatividad del alumno que se adueña del comportamiento del programa y por lo tanto le motiva de forma intrínseca a la creación de sus propios programas*”... “*creatividad*”... “*podrán desarrollar juegos, expresar emociones y sentimientos, así*”)

²⁹² “*Think out of the box*”: traducido al español como “Pensar fuera de la caja”, es una metáfora que significa pensar diferente, de manera no convencional o desde una nueva perspectiva. Esta frase casi siempre se refiere al pensamiento novedoso o creativo

como, reflejar pensamientos e ideas de una forma interactiva y creativa”); trabajo en equipo (“trabajo colaborativo”... “promover el aprendizaje entre iguales, el intercambio de estrategias y conocimientos entre ellos”... “el trabajo colaborativo”); actitudes prosociales (“habilidades como la responsabilidad social al compartir proyectos”); e incluso “inglés”. Y también las posibilidades del aprendizaje del ‘coding’ para la atención de la diversidad (“ofrecer actividades y propuestas a alumnos desmotivados, alumnado de altas capacidades y de NEE”)

Finalmente, llama la atención que el término ‘pensamiento computacional’ sólo es nombrado explícitamente en una ocasión, por el sujeto ID_22 (“*se trabaja el pensamiento o lógica computacional*”), sugiriendo que, por el momento, es un término que no está extendido entre el colectivo docente (a diferencia del colectivo académico-investigador). Ahora bien, sí aparecen respuestas en la línea que propugna el enfoque del ‘pensamiento computacional’, tal y como vimos en la parte teórica de esta tesis, defendiendo que el ‘coding’ no debe ser un fin en sí mismo sino un medio para desarrollar una amplia gama de aptitudes y habilidades, transferibles a diversos campos (“*usar la tecnología no como instrumento sino como contexto de situaciones de aprendizaje*”... “*la clave para lograr todo el potencial que tiene la programación informática para desarrollar distintas aptitudes y capacidades es que el docente la entienda como una herramienta multidisciplinar y la utilice dentro de su programación didáctica con distintos enfoques, dependiendo del objetivo que se proponga en cada momento*”)

En lo relativo a la **Pregunta 2** del cuestionario [*“¿Considera usted el aprendizaje del ‘coding’ especialmente adecuado para algún tipo específico de estudiante? (p.ej. para los chicos frente a las chicas, para los alumnos de alta capacidad frente a los alumnos de capacidad media, para los estudiantes de ESO frente a los estudiantes de Primaria, etc...) ¿Por qué?”*], que cubre el objetivo de investigación **O_{2P1}**: hay un consenso generalizado alrededor del ‘no’ o, dicho de otra forma, de la adecuación del ‘coding’ para todo tipo de alumnos (“*no, creo que el coding es adecuado para todos los alumnos, sean chicos o chicas*”... “*creo que el coding es necesario para todo tipo de alumnado*”... “*no, todos los estudiantes están cualificados, dentro de las posibilidades de su edad, para aprender a programar igual que lo están para aprender a escribir o comprender matemáticas*”... “*I think learning about coding and at least some introductory coding is appropriate for any and all users of technology*”... “*creo que el aprendizaje de la programación es aconsejable para alumnos de todo tipo y edad; en mi opinión se trata de un conocimiento transversal, útil para todo tipo de alumnos, y que se puede hacer atractivo para todos ellos*”... “*no, eso sería como decir que la lectoescritura o el cálculo o cualquier otra área es especialmente adecuada para un tipo específico de estudiante; como en cualquier otro campo, a unos estudiantes se les da mejor que a otros, a unos les gusta más que otros... pero es positivo para todos*”... “*no creo que sea especialmente adecuado para un grupo concreto de alumnado*”... “*I have found that it is suitable for all ages, levels and both genders!*”... “*en mi opinión es apto para todas las edades, sexos y diferentes capacidades*”).

En menor medida, aparece también cierto consenso alrededor de que: el aprendizaje del ‘coding’ puede iniciarse en edades tempranas, incluso contemplando la etapa de Educación Infantil (“*cuanto antes empiecen a programar, el desarrollo de las aptitudes y habilidades antes mencionadas será mayor*”... “*sería bueno que empezaran a aprender a programar en los últimos cursos de primaria*”... “*las actividades relacionadas con la programación informática se pueden desarrollar*

desde la Educación Infantil, pero, obviamente, no pueden ser las mismas para alumnos de cuatro años que para alumnos de catorce”... “desde educación infantil el alumnado responde a estos aprendizajes con habilidad, seguridad y motivación”... “es importante remarcar que cuanto más pequeños más entusiasmo y menos miedo a enfrentar; los pequeños enfrentan este aprendizaje sin complejos y con una habilidad sorprendente; de manera que un trabajo prematuro en este tipo de aprendizaje puede ser mucho beneficioso para ellos en su proceso educativo posterior”); la virtud que tiene el aprendizaje de la programación informática para adaptarse al ritmo de cada estudiante (“en especial porque resulta muy fácil de adaptar a los distintos ritmos de trabajo de los alumnos, que es una de las principales dificultades con las que se encuentran los maestros hoy en día”... “permite que cada uno siga su propio ritmo”); la advertencia de que no siempre los alumnos que destacan en las asignaturas tradicionales son los que destacan en el ‘coding’ (“no depende de la inteligencia cognitiva sino de otro tipo de inteligencia que la puede tener o no cualquiera”... “el alumnado con más capacidades en las áreas habituales no siempre es el más interesado ni el más hábil”); o el peligro de que las chicas no opten por aprender programación por los estereotipos negativos de género asociados (“more males tend to take programming classes partly because the female students don't like the ‘nerd’ stigma that is attached to programmers”... “actualmente existe una brecha muy grande entre las chicas que se dedican a la computación y los chicos; lo que habría que hacer es involucrar más a las chicas en este mundo, quitarles la ideas tanto a unos como a otros de que la programación/computación es cuestión de chicos”)

En lo relativo a la **Pregunta 3** del cuestionario [*¿Espera usted algún tipo de correlación entre el rendimiento habitual del estudiante en algún/as área/s curricular/es y su nivel de motivación y ejecución en la actividad de la CSEW? (p.ej. posible correlación entre el rendimiento en matemáticas y su futuro desempeño en la CSEW)*], que cubre el objetivo de investigación **O_{2Pc}**: aparecen moderadamente algunas respuestas que correlacionan el rendimiento en el área de matemáticas (a veces acompañadas de otras áreas) con el nivel de ejecución en tareas de ‘coding’, y con la matización de que estudiantes con bajo rendimiento habitual en matemáticas pueden también motivarse y divertirse con el ‘coding’, aunque su desempeño no sea excelente (*“it is likely that students who demonstrate strength in mathematics, may also show strength related to the coding activities; however, there may also be students who respond to the visual component of coding on the computer who may show greater strength when coding then in their classroom”... “I think there is a positive correlation between math and programming, however, some students who are weak at math also enjoy programming”... “especialmente en las matemáticas y también en las ciencias experimentales al fomentar su autonomía”... “creo que los alumnos con notas medias más altas, especialmente en matemáticas y música, tendrán más facilidad para la ejecución de la actividad, pero no creo que el rendimiento vaya a influir en el nivel de motivación”... “en cuanto a áreas, las matemáticas, pero nuestra programación favorece todas las competencias básicas”... “posiblemente será así, cuánta más capacidad matemática se tenga, más rápido y fácil será la ejecución de la actividad”... “correlación con matemáticas, artística”... “en la ESO se podría esperar un mejor rendimiento en aquellos que obtienen buenos resultados en las áreas de matemáticas y física y química con facilidad”*).

En otro sentido se expresa algún profesor que narra cómo alumnos de bajo rendimiento consiguen incrementarlo a raíz de una alta motivación en tareas de programación: *“tengo la experiencia*

personal de alumnos con bajo rendimiento en otras áreas y alta motivación en ciencias de la computación, incluso, que esta motivación, con el tiempo, les ayuda a mejorar su rendimiento general cuando ven la necesidad de las matemáticas, la comprensión lectora o el inglés en el ámbito de la informática y más concretamente en la programación”

En lo relativo a la **Pregunta 4** del cuestionario [“*Existen distintas plataformas-tutoriales ‘on-line’ (p.ej. ‘Scratch’) para el aprendizaje del ‘coding’ ¿Cuáles de ellas conoce? ¿Cuáles de ellas le parecen más apropiadas para cada edad?*”], que cubre el objetivo de investigación **O_{2Pd}**: los profesores encuestados reportan una amplia variedad de recursos ‘on-line’ para el aprendizaje del ‘coding’. Ordenados orientativamente según la edad de uso referida por los profesores para cada recurso, se señalan para Educación Infantil y comienzo de Educación Primaria: ‘Kodable’²⁹³, ‘Lightbot’²⁹⁴, ‘Bee-Bot app’²⁹⁵, ‘ScratchJr’²⁹⁶, ‘Daisy the Dinosaur’²⁹⁷ o ‘The Foos’²⁹⁸; para finales de Educación Primaria y comienzos de Educación Secundaria: ‘Blockly’²⁹⁹, ‘Hopscotch’³⁰⁰, ‘Code.org’ (tutorial de *Angry Birds*)³⁰¹, ‘Tynker’³⁰², ‘Alice’³⁰³, ‘Kodu’³⁰⁴ y ‘Scratch’³⁰⁵ (de éste último se destaca repetidamente su gran versatilidad para ser utilizado con estudiantes de distintas edades y distintos niveles de experiencia previa en programación); para finales de Educación Secundaria y Bachillerato: ‘App Inventor’³⁰⁶, ‘Code Combat’³⁰⁷, ‘TouchDevelop’³⁰⁸, ‘Codecademy’³⁰⁹, ‘Code School’³¹⁰, ‘Khan Academy’³¹¹ (de éste último trío de recursos se destaca repetidamente por los profesores que son sólo aptos para alumnos independientes, de estilo autónomo).

Adicionalmente, los profesores refieren otros recursos conocidos como ‘CS Unplugged’³¹², kit de juegos de aprendizaje de conceptos computacionales sin utilizar ordenador; u otros recursos que permiten conectar el código con el mundo físico a través de la robótica y la electrónica: ‘Lego Mindstorms’³¹³, ‘Moway’³¹⁴, ‘BitBloq’³¹⁵ o ‘Arduino’³¹⁶

²⁹³ <https://www.kodable.com/>

²⁹⁴ <http://lightbot.com/>

²⁹⁵ <https://itunes.apple.com/es/app/bee-bot/id500131639?mt=8>

²⁹⁶ <http://www.scratchjr.org/>

²⁹⁷ <https://itunes.apple.com/es/app/daisy-the-dinosaur/id490514278?mt=8>

²⁹⁸ <http://thefoos.com/>

²⁹⁹ <https://blockly-games.appspot.com/>

³⁰⁰ <https://www.gethopscotch.com/>

³⁰¹ <https://studio.code.org/hoc/1>

³⁰² <https://www.tynker.com/>

³⁰³ <http://www.alice.org/>

³⁰⁴ <http://www.kodugamelab.com/>

³⁰⁵ <https://scratch.mit.edu/>

³⁰⁶ <http://appinventor.mit.edu/explore/>

³⁰⁷ <https://codecombat.com/>

³⁰⁸ <https://www.touchdevelop.com/>

³⁰⁹ <https://www.codecademy.com/>

³¹⁰ <https://www.codeschool.com/>

³¹¹ <https://www.khanacademy.org/computing/computer-programming>

³¹² <http://csunplugged.org/>

³¹³ <http://www.lego.com/en-us/mindstorms/>

³¹⁴ <http://moway-robot.com/>

³¹⁵ <http://bitbloq.bq.com>

³¹⁶ <https://www.arduino.cc/>

En lo relativo a la **Pregunta 5** del cuestionario [“Para planificar las actividades de la CSEW, ¿ha comparado distintas plataformas-tutoriales ‘on-line’ tal y como se sugiere en <http://csedweek.org/learn?> ¿Por cuál/es se ha decidido? ¿Por qué?”], que cubre el objetivo de investigación **O_{2Pe}**: para el tramo de edad de entre 10-15 años (que, como vimos en la Figura 5.1 es el mayoritario en participación en ‘La Hora del Código’, representando sobre el 80% del total de estudiantes) el tutorial más utilizado es el propuesto por Code.org³¹⁷ con los personajes del videojuego ‘Angry Birds’³¹⁸, por varias razones (“la actividad de ‘Angry Birds’ programada creo que se adapta muy bien al tiempo de una hora, está muy bien explicada, está en castellano desde el principio, es repetitiva en los pasos iniciales, totalmente gráfica... además es simpática”... “la de Angry Birds por ser más atractiva para los estudiantes”... “Angry Birds tutorials are the most robust”... “efectivamente, he comparado distintas actividades y la que me ha parecido más fácil y atractiva para los alumnos/as que realizarán la actividad es la de los bloques de Angry Birds”... “para el alumnado de 4º, 5º y 6º (Primaria) he optado por Angry Birds porque pienso que conectará directamente con su interés al ser protagonistas unos personajes que ya conocen, por el nivel de dificultad”... “si he preferido Angry Birds porque a priori resulta de interfaz más atractiva y de resolución más sencilla”... “me decidí por el de <http://hourofcode.com/code> porque era el más ‘popular’ en cuanto a los personajes (Angry Birds) para enganchar a los chic@s”). Este tutorial consiste en un conjunto de 20 pantallas, cada una de las cuales plantea un problema cerrado o puzle de programación, que el estudiante debe resolver secuenciando el algoritmo correcto con los bloques de código de los que dispone (Figura 5.11)

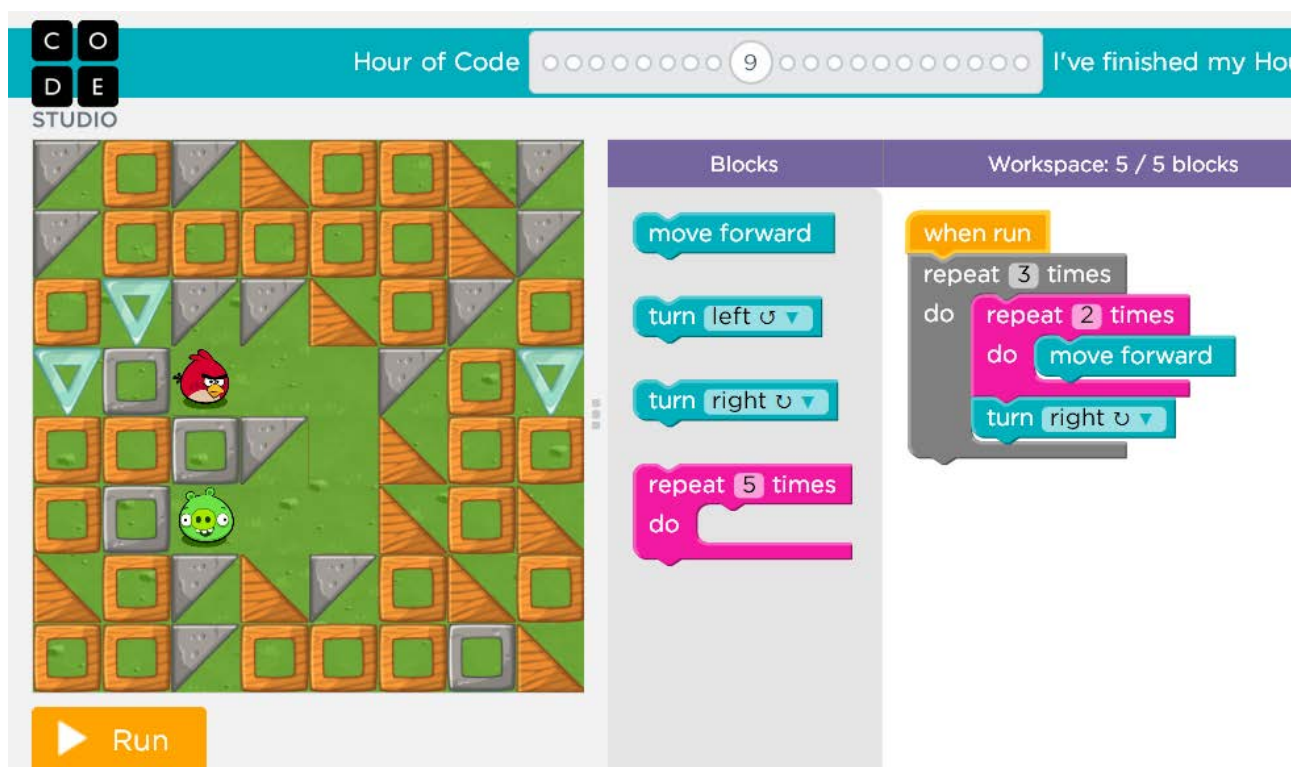


Figura 5.11. Captura de una de las pantallas del tutorial propuesto por Code.org, utilizando los personajes de ‘Angry Birds’. A la izquierda se plantea el problema o puzle de programación (llevar al pájaro hasta el cerdo por el camino señalado); a la derecha se resuelve el puzle secuenciando un algoritmo con los bloques de los que se dispone.

³¹⁷ <https://studio.code.org/hoc/1>

³¹⁸ <https://www.angrybirds.com/>

Para alumnos menores de 10 años, el tutorial más utilizado durante la HoC es *'Lightbot'* (Figura 5.12), referenciado algo más arriba, por su sencillez (*"Empezaremos por las más simples como el Lightbot..."*), que consiste igualmente en retos cerrados o puzzles de programación, pero que en este caso pueden resolverse mediante flechas (sin necesidad de lectura)

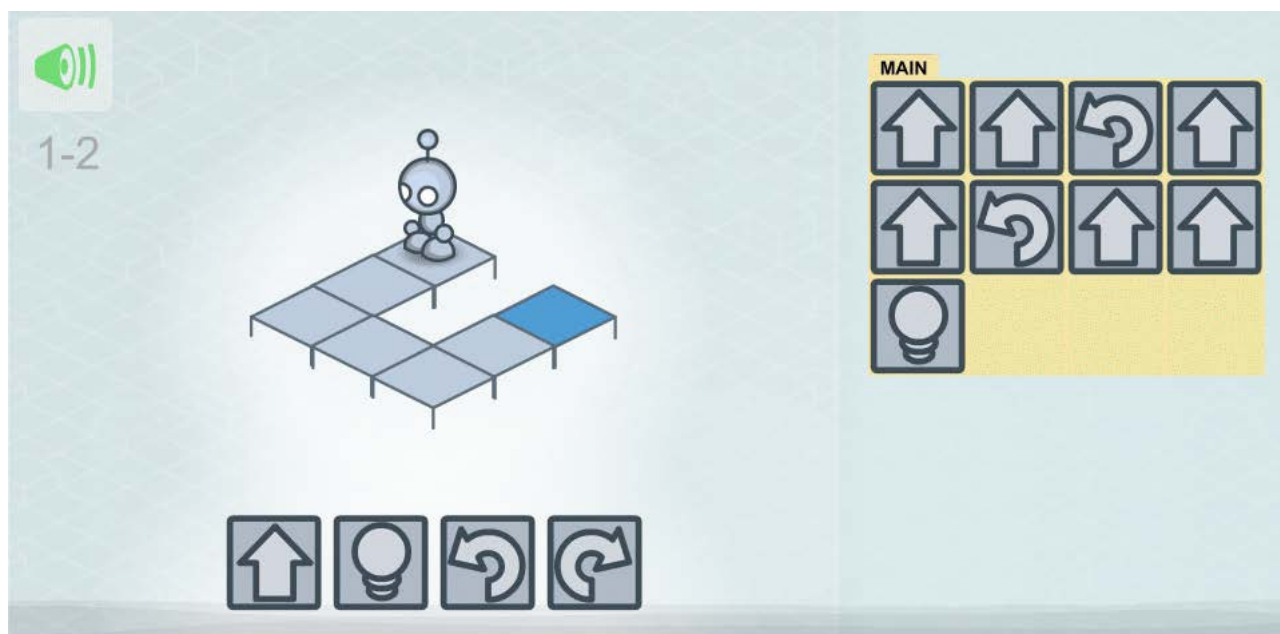


Figura 5.12. Captura de una de las pantallas del tutorial *'Lightbot'* para la HoC. A la izquierda se plantea el puzzle de programación (llevar al robot hasta la casilla azul); a la derecha, la secuencia de instrucciones que lo resuelve.

Y en el caso de alumnos de Secundaria con algo de experiencia previa en programación, los tutoriales más utilizados son los de Scratch³¹⁹; en este caso, son proyectos abiertos (*'open-ended'*) en los que el estudiante parte de un *'lienzo en blanco'* y, con ayuda de un tutorial, crea una animación, historia interactiva o juego a través del lenguaje de programación visual de Scratch. Algún profesor reportó al respecto la utilización de la metodología *'live coding'* (*"voy a hacer una clase de 'live coding' con Scratch haciendo un juego de laberinto"*), en la cual el docente proyecta en la pizarra cómo va programando en tiempo real una animación o juego en Scratch, mientras el resto sigue su ejemplo como referencia para programar el suyo propio.

En lo relativo a la **Pregunta 6** del cuestionario [*"Entendemos que al incorporar a su centro a la CSEW ('An Hour of Code') considera viable que con una actividad de una hora se pueda introducir a los estudiantes en el mundo del 'coding', ¿es así? ¿Cómo es esto posible?"*], que cubre el objetivo de investigación **O_{2PF}**: la práctica totalidad de los profesores está de acuerdo en señalar que con una actividad de una hora es posible introducir a los estudiantes en el mundo y práctica del *'coding'*, desde una perspectiva lúdica y como forma de incentivar su curiosidad para proseguir con un aprendizaje más profundo del mismo (*"sí, ya que pueden ver cómo se 'programa' y algunos de ellos (no todos) que les falta el 'empujoncito' pueden iniciarse en los recursos online (que posiblemente no conocen)..."* *"one hour of code is to give the students of taste of coding; it can also help them to decide whether they want to take the programming electives offered at our school"...* *"con una hora podemos generar un cierto interés, que esperamos que se extienda por toda la comunidad educativa*

³¹⁹ <https://scratch.mit.edu/hoc/>

del centro”... “creo que les puede despertar la curiosidad”... “es posible introducir a los estudiantes en el mundo del coding, será suficiente para que unos pocos se interesen, y para que algunos de esos pocos pidan o busquen información de forma más seria”... “una actividad de una hora sirve para generar una sensación en el alumno; un interés que después debe ser conducido hacia un trabajo en profundidad haciendo uso de las plataformas mencionadas”... “lo que hacemos es darles el ‘caramelo’, que lo prueben y ya es cuestión de seguir fomentando desde el cole (lo ideal) o desde casa el gusto por la programación”... “como programadora y madre pienso (y la cara y emoción de los chicos del martes pasado me lo confirman) que sí se puede introducir en una hora a los estudiantes en el mundo de la programación”... “a mis alumnos le gusta la programación; esta actividad les abre otras posibilidades que no aún no conocen y les seducen”)

También se señala la importancia del evento para dar una idea a los estudiantes, y demás comunidad educativa, de la accesibilidad de aprender ‘código’ a través de los nuevos lenguajes visuales (“*con iniciativas como esta se permite que la gente se dé cuenta de que programar es más accesible de lo que parece, ya que en la actualidad hay una gran gama de recursos que simplifican el proceso*”); y para desmitificar su extrema dificultad (“*como la propia organización Code.org dice, es perfectamente posible aproximarse a los fundamentos básicos de la programación con su tutorial de una hora; de esta forma se desmitifica la creencia de que la programación informática es sólo para gurús*”).

Por último, se resalta la importancia de que, con los nuevos lenguajes y plataformas de aprendizaje del ‘coding’, se generen de manera inmediata productos que el estudiante puede disfrutar y compartir al momento (“*es posible que vean en qué consiste desarrollar un mini-programa o pequeño juego y de ahí imaginar lo que es una programación más a fondo*”... “*con Scratch en una hora se puede hacer un juego sencillo con los alumnos ya que no tienen que escribir y los resultados aparecen en la pantalla inmediatamente sin tener que compilar y ejecutar*”)

En lo relativo a la **Pregunta 7** del cuestionario [“*Durante la preparación de las actividades de la CSEW, ¿qué apoyos está encontrando entre la comunidad educativa de su centro? ¿Qué resistencias?*”], que cubre el objetivo de investigación **O_{2Pg}**: en general, los profesores reportan gran cantidad de apoyos por parte de la comunidad educativa (“*apoyo total, tanto de dirección que les gustó mucho la idea, me consta que lo comentaron con inspección y que también gustó, como del resto de compañeros a los cuales les he tenido que pedir que cedan a sus alumnos durante las horas programadas para que todo el alumnado del centro no sólo los alumnos de informática participaran en esta actividad*”... “*the admin was very supportive of this week*”... “*apoyo total por parte del equipo directivo y de los padres más implicados en las actividades del colegio, principalmente socios de la AMPA; apoyo también por parte de profesores con los que tengo más relación personal y que confían en mi criterio al proponer una actividad*”... “*apoyo total por parte del director y buena acogida y colaboración por parte de los compañeros involucrados*”); comunidad educativa que se muestra sensible al valor del ‘coding’ en la sociedad actual (“*creo que todo el mundo entiende el valor profesional de la programación y las perspectivas de futuro que tiene*”... “*they are supportive as they understand the importance of programming in today’s society!*”), y sus beneficios para el desarrollo de aptitudes valiosas (“*he recibido muchos apoyos por parte de los profesores que entienden que la programación ayuda a organizar la mente y al razonamiento abstracto*”)

Prácticamente no se indican resistencias (*“resistencia: ninguna”... “no hemos detectado ninguna resistencia al proyecto”... “no ha habido (que yo sepa) retencencias”... “resistencia ninguna”... “no he tenido resistencias en mi centro”... “ninguna resistencia por parte de la comunidad educativa”... “ninguna resistencia; los dos profesores de informática así lo decidimos y no hubo ningún problema, al contrario”... “no ha habido ninguna resistencia”*), si acaso algún reparo de otros profesores por cierto temor o inseguridad ante el ‘coding’ (*“las resistencias son por parte de profesores que piensan que va a ser difícil o que afirman no tener tiempo para organizar más eventos o actividades”*), o más bien alguna dificultad derivada de insuficiencias técnicas del centro (*“la parte negativa es el poco ancho de banda disponible en el centro, lo cual ralentiza la conexión a la red”*)

En lo relativo a la **Pregunta 8** del cuestionario [*“¿Considera ‘aprender a programar’ una competencia relevante para el futuro personal (como ciudadanos) de sus estudiantes? ¿Por qué?”*], que cubre el objetivo de investigación **O_{2Ph}**: prácticamente la totalidad de los profesores declaran que sí, por distintas razones. Por un lado, razones que tienen que ver con la omnipresencia de lo digital en el mundo que les ha tocado vivir, y que les espera en el futuro (*“sí, totalmente necesaria porque (...) desarrollarán su capacidad de adaptación a todos los cambios tecnológicos que se están produciendo y se van a producir... “sí, porque cada día dependemos más de los ordenadores y para prácticamente cualquier faceta de la vida se usará un programa”... “considero que es interesante en tanto que las nuevas tecnologías están adquiriendo una presencia fundamental en todos los ámbitos de la vida; me parece esencial que todas las personas tengan unos conocimientos básicos de programación que les hagan más eficientes a la hora de comprender el funcionamiento de los ordenadores”... “totalmente de acuerdo; nuestros alumnos son digitales y por lo tanto los tenemos que prepararlos para su total integración en la sociedad”... “sí, ya que se verán enfrentados al uso de ordenadores en su día a día”*).

También aparecen justificaciones relativas a las aptitudes y habilidades cognitivas de alto nivel que pueden desarrollar los estudiantes al aprender ‘coding’ (*“sí, como he dicho antes permite ver la ‘resolución de problemas’ de otra manera”... “el pensamiento computacional se puede aplicar en todos los ámbitos de la vida, por lo que habiéndolo desarrollado mediante el aprendizaje de la programación les permitirá afrontar situaciones o solucionar problemas de muy diversa índole de una forma más eficaz”... “definitivamente sí; creo que aprender a resolver problemas por medio de la programación informática puede ayudarles a ser más reflexivos y analíticos en su vida, en la toma de decisiones y al afrontar otros retos”... “sí, debe considerarse una herramienta que ayude a resolver problemas; programar para aprender”... “yes, it is essential; specially in Spain where I am concerned about the methods used to study; there does not appear to be a method of problem-solving, which is essential for university and further study”... “totalmente; aunque el día de mañana no se dediquen a la computación, lo que aprenden programando les va a servir para muchas otras áreas de su vida: van a aprender a ser estructurados, lógicos, ordenados, creativos”*)

En menor medida, los profesores esgrimen razones de índole creativa (*“sí, porque hay muchos dispositivos digitales y es bueno saber crear contenido”... “además al potenciar la creatividad y de apropiación de la creación, puede permitir a los alumnos el desarrollo de sus potencialidades como seres humanos en cualquier ámbito”... “considero que dentro de un entorno de programación, los alumnos/as podrán desarrollar juegos, expresar emociones y sentimientos, así como, reflejar*

pensamientos e ideas de una forma interactiva y creativa”); o de raíz crítica, en el sentido de hacer a los estudiantes conscientes de lo que hay ‘tras las pantallas’ para no asumir de manera irreflexiva su funcionamiento (“I think all students should have at least one programming class, as it gives them some perspective on computing and what exists behind the screen; perhaps this will help in thinking more clearly about information accessed via the computers and increase critical thinking skills that would hopefully contribute to being a more engaged citizen”... “por otra parte, tendrán una visión más crítica como consumidores de tecnología, además de que podrán, en algunos casos, crear herramientas adaptadas a sus necesidades”). Finalmente, también aparece una respuesta en términos de aprender a programar como un nuevo alfabetismo (“sí, en este momento creo que es como saber leer o escribir o interpretar un plano”)

En lo relativo a la **Pregunta 9** del cuestionario [*“¿Considera ‘aprender a programar’ una competencia relevante para el futuro profesional (como trabajadores) de sus estudiantes? ¿Por qué?”*], que cubre el objetivo de investigación **O_{2PI}**: a excepción de una terna de profesores que limitan la relevancia del ‘coding’ para algunas profesiones específicas (*“si trabajan en un área relacionada con la informática, sí; si no pues no creo que les sirva de mucho saber programar”... “sólo en ciertas profesiones (ingeniería, ciencias)”... “siempre que quieras dedicarte más a este sector”*); el resto de docente están de acuerdo en afirmar la relevancia del ‘coding’ para el futuro profesional de sus estudiantes: bien sea por la creciente demanda de trabajadores con perfiles que incluyan conocimientos de programación (*“van a surgir nuevos puestos de trabajo que requieren conocimientos en ‘coding’ y la tendencia de formación en estas áreas no parece que vaya a cubrir todas las necesidades futuras”... “también ya que cada vez son necesarios más programadores”... “es la profesión del futuro”... “sí claro; muchos de los trabajos implican conocimiento de lenguas e informática; las máquinas se programan”*), o bien por considerarse que las habilidades desarrolladas con el ‘coding’ pueden transferirse a la inmensa mayoría de trabajos de la sociedad digital (*“sí, ya que en cualquier trabajo se le van a presentar problemas no esperados y van a tener que resolverlos”... “cada día dependemos más de los ordenadores y es difícil imaginar un trabajo dentro de 10 años que no utilice un programa de algún modo; automatizar tareas es fundamental para ser productivo y gracias a la programación puede conseguirse”... “sí, hasta en Excel hay ya fórmulas que parecen programas; si programas, haces análisis, diseño, codificación y prueba”... “I think all students should learn some basic level of coding, as I see it as a form of literacy; many jobs require use of computing and some understanding of a variety of tools; the most skill and information an individual has will likely correlate to future opportunities in their professional life”... “sí, en algún momento en su carrera profesional deberán hacer algo relacionado directa o indirectamente con la programación”... “el pensamiento computacional podrá aplicarse a resolver situaciones o problemas del entorno laboral; además, actualmente el uso de ordenadores y aplicaciones informáticas está a la orden del día en casi cualquier trabajo, y va a ir en aumento; la diferencia entre saber cómo utilizar una aplicación informática determinada o saber cómo funciona dicha aplicación puede ser abismal; y no digamos si le sumamos la posibilidad de adaptar o crear una aplicación para cubrir las necesidades específicas de una determinada tarea”... “el desarrollo de software está cada vez más presente en áreas que no están exclusivamente relacionadas con la informática; en este sentido, tener un contacto con el coding, les puede favorecer en un futuro profesional”... “sí, porque mejorará sus capacidades de resolución de*

problemas y porque les permitirá crear aquellas aplicaciones que puedan necesitar y no tengan a su disposición en el futuro”)

Finalmente, en lo relativo a la **Pregunta 10** del cuestionario [*“Por favor, describa la/s actividad/es que tiene previsto realizar durante la CSEW: ¿cuántos estudiantes? ¿De qué edades? ¿Con qué recursos? ¿Con qué metodología? ¿Habrá algún tipo de evaluación? Puede incluir en su descripción igualmente cualquier otro comentario (o enlaces) que desee hacer al respecto”*], que cubre el objetivo de investigación **O_{2P}**; las descripciones completas, algunas bastante extensas, pueden leerse al final del documento en la ‘nube’³²⁰, que hemos enlazado con el código QR de la Figura 5.10. Baste aquí reflejarse que, en líneas generales, las actividades planificadas se apoyan en una multitud de recursos y tutoriales ‘on-line’, adaptados para cada edad, como los referidos algo más arriba; que, en muchos centros la actividad involucra de manera masiva a todo el estudiantado (*“participa la totalidad del centro: 330 estudiantes”...“vamos a participar con 24 grupos de Enseñanza Secundaria Obligatoria, unos 600 alumnos”...“all age groups in secondary; years 7 to 13 will be undertaking 1 class of programming this Week”*), dándose al evento una amplia difusión, tanto intramuros como extramuros, a través de la cartelería, videos y ‘merchandising’ diverso que se proporciona desde la web de ‘La Hora del Código’³²¹ (*“hemos colgado en la página web información sobre el evento y colocado dentro del centro, los carteles que proporcionaba la organización; durante esta semana en los descansos se proyectará en una pantalla de recepción y se escuchará por la megafonía del centro uno de los vídeos que explicaban el evento; en las clases de informática les estamos explicando en que va a consistir este evento; los alumnos van a traducir y comentaremos después en clase las distintas frases que hay en los posters que hemos colgado por el centro; también hemos enviado por email a los padres copia de la carta que habíamos colgado en la web”*).

En la mayor parte de los centros, el evento se celebra en el aula/s de informática, equipada con PC’s fijos, por la cual van pasando los distintos grupos (*“aulas informática de 24 ordenadores cada una”...“en la sala de ordenadores del centro”...“disponemos de aula de informática y proyector”...“esto se hará en el aula de informática”*) aunque también encontramos casos que utilizan carritos móviles con portátiles o tabletas (*“tablets Samsung e iPad”*)

La metodología usada es siempre activa, aprendiendo cada estudiante a su propio ritmo y de forma interactiva con el tutorial que va siguiendo en la pantalla (*“3/4 min de presentación de la plataforma y posterior realización de las actividades”...“dedicaría 10 minutos a explicar el porqué de la actividad y el resto sería una clase práctica”...“realizarán la actividad de code.org relativa al tutorial de Angry Birds”*). Encontramos también algunas referencias a aprendizajes por pares o mentoría realizada por voluntarios de cursos superiores (*“I will also include (high school) students in my tech classes to assist their middle school peers”*), y a la metodología propuesta por la propia Code.org denominada ‘Ask 3, then me’³²², que consiste en que cuando un estudiante no sabe cómo completar un determinado reto de programación, pregunte al menos a 3 compañeros en busca de ayuda antes de consultar finalmente al profesor (*“dando instrucciones de que si alguien se queda*

³²⁰ Disponible en: <https://goo.gl/5Kxy2x>

³²¹ <https://hourofcode.com/es/promote/resources>

³²² <https://csedweek.org/educate/hoc>

atascado en algún punto pregunte a los compañeros que tiene al lado, y en caso de que ninguno de ellos sepa ayudarle, le pregunte a ella (la profesora)”

Llama la atención que no se contemplan mecanismos de evaluación formales a los estudiantes (“no hay evaluación”...“los alumnos no serán evaluados”...“no habrá una evaluación formal”), posiblemente por el carácter lúdico del evento (“será una actividad esencialmente lúdica”). Sólo se contempla el uso de algún cuestionario de satisfacción para profesores y/o estudiantes (“los alumnos/as deberán realizar un cuestionario al finalizar la actividad”... “cada grupo cumplimentará un formulario online de satisfacción”), el ‘tracking’³²³ de seguimiento que proporcionan las propias plataformas-tutoriales ‘on-line’ (“la evaluación que utilizamos es la que el mismo programa de registro nos permite ver aquellos alumnos que han completado la tarea de aquellos que aún no”) (Figura 5.13), y la entrega del certificado de completamiento de la actividad (“se dará un certificado de asistencia y aprovechamiento a aquellos alumnos que finalicen las 20 actividades de las que consta Angry Birds”... “al final cuando acaben les daremos el certificado que existe al finalizar la Hour of Code”) que ya ilustramos en el sub-epígrafe 2.2.3.1 de la parte teórica (Figura 2.11)

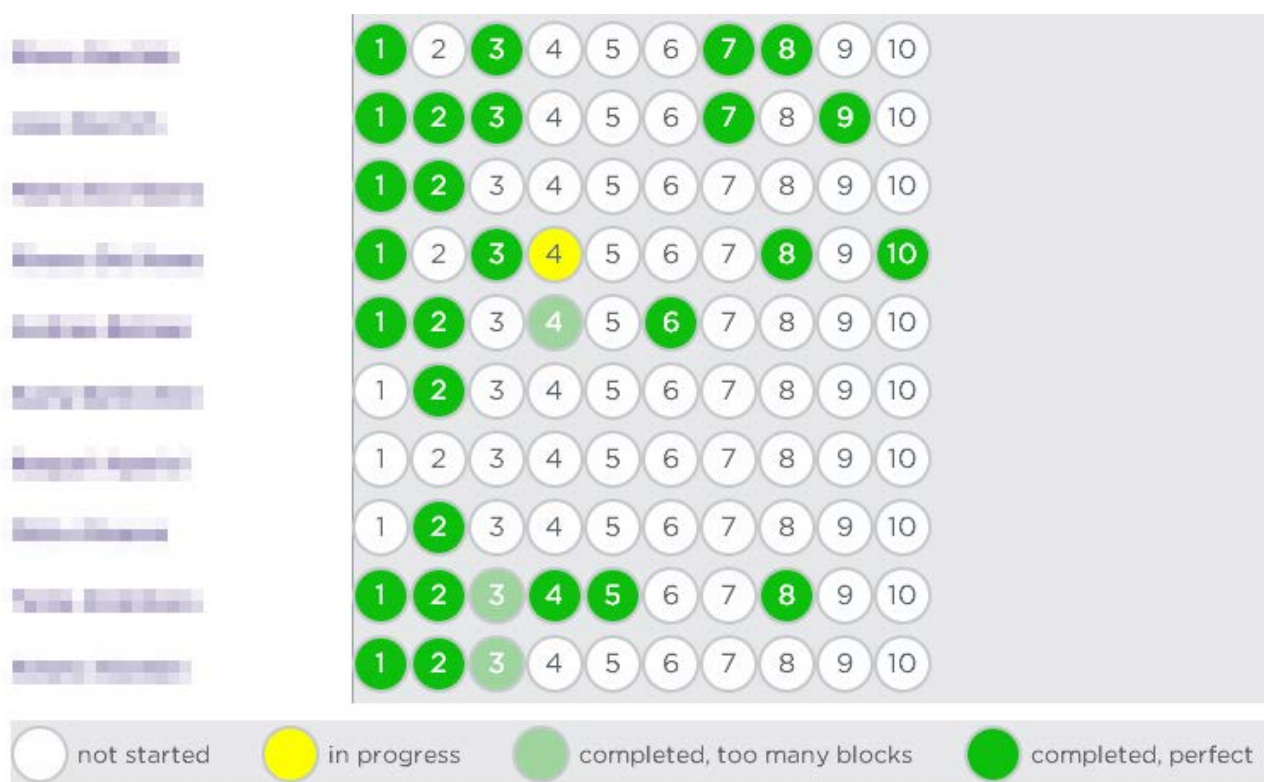


Figura 5.13. Ejemplo del ‘tracking’ que se puede realizar a los estudiantes, sobre los niveles que van intentando y completando a lo largo de ‘La Hora del Código’

Finalmente, con respecto a las edades involucradas, la oscilación es amplísima: entre Educación Infantil (“Educación Infantil (3, 4 y 5 años), aproximadamente 110 participantes”) y Formación Profesional; escogiéndose recursos y tutoriales adaptados en función de cada una de ellas. Por su especial interés, al ser la primera iniciativa en Educación Infantil de la que este doctorando tiene

³²³ ‘Tracking’: término en lengua inglesa para referirse coloquialmente al ‘rastreo’ que las plataformas ‘on-line’ de aprendizaje hacen de la actividad realizada en su seno por el estudiante.

noticia, se reproduce a continuación la descripción de la actividad que hace el sujeto ID_11 (Jorge Lobo, maestro y coordinador TIC del CEIP Lope de Vega de Madrid):

“ALPHABOT, LA ROBOT³²⁴: Educación Infantil (3, 4 y 5 años). Aproximadamente 110 participantes. Recursos: Disfraz de robot; Escenarios o recorridos; Tarjetas de programación para cada participante; Vídeo de introducción/motivación (si da tiempo a editarlo); PDI o proyector para el vídeo. Metodología: El grupo de alumnos se reunirá en una sala con los escenarios preparados. Se les explicará que ALPHABOT es una robot muy hábil y fuerte, pero que necesita ayuda para completar una serie de tareas. Ellos (los alumnos) pueden ayudarle a conseguir los objetivos con las tarjetas de programación que tienen. En cada paso, los alumnos tendrán que levantar la tarjeta que creen que es apropiada para guiar a la robot. El profesor tutor comprobará si todos están de acuerdo, y si hay diversidad de opiniones, pedirá a algunos alumnos que expliquen por qué han elegido una tarjeta y no otra, activará el robot para que ejecute las acciones o reseteará el último movimiento si no ha sido el más apropiado para que los alumnos puedan elegir de nuevo. Evaluación: Los profesores participantes evaluarán la actividad: - El grado de comprensión de las instrucciones por parte de los alumnos; - El grado de comprensión del funcionamiento del robot por parte de los alumnos; - El grado de implicación de los alumnos en la actividad; - Si la actividad se ha adaptado a la capacidad de los alumnos. Los alumnos no serán evaluados, pero tendrán un espacio de reflexión para expresar sus opiniones sobre la actividad, comentar cómo se han sentido, qué les ha gustado más, etc.” (Sujeto ID_11: Jorge Lobo, maestro y coordinador TIC del CEIP Lope de Vega de Madrid)

5.4.3. Relativos al cuestionario *post* para profesores

Pasamos ahora a dar respuesta a los objetivos $O_{2Pk...O_{2Pñ}}$, que se abordan a través del “CSEW: Cuestionario para profesores/as (*post*)” (**Anexo B**).

A través de la Figura 5.14 se puede acceder a un documento³²⁵ que incluye el contenido literal y completo de las respuestas de los 14 profesores a las 5 preguntas abiertas que componen el “CSEW: Cuestionario para profesores/as (*post*)”, y que cubren respectivamente los objetivos $O_{2Pk...O_{2Pñ}}$ de la investigación.



Figura 5.14. Código QR al documento con el contenido completo de las respuestas al cuestionario *post* para profesores

³²⁴ Puede consultarse una descripción más detallada, acompañada de imagen y vídeos, a través del blog de Jorge Lobo: <http://ceipmiskatonic.blogspot.com.es/2014/11/se-acerca-la-hora-del-codigo.html>
<http://ceipmiskatonic.blogspot.com.es/2014/12/alphabot-programando-en-educacion.html>

³²⁵ Disponible en: <https://goo.gl/Wi9sXk>

Con respecto a la **Pregunta 1** del cuestionario [*“¿Ha podido desarrollar las actividades de la CSEW tal y como tenía planificado? ¿Sí, No? ¿Por qué? Describa las actividades CSEW que finalmente han podido realizar: nº de estudiantes, niveles educativos, tutoriales utilizados, metodología seguida, sistema de evaluación, etc...”*], que cubre el objetivo de investigación **O_{2PK}**: en general, los profesores informan que sí pueden llevar a cabo las actividades según su planificación inicial (*“en buena medida”... “prácticamente sí”... “yes, the week went as planned”... “sí, ha ido todo según lo previsto”... “sí, porque han funcionado tal y como estaba previsto”*). Una descripción detallada, alguna bastante extensa, de las actividades finalmente llevadas a cabo puede leerse al comienzo del documento completo enlazado en la Figura 5.14. Baste aquí decir que las pocas alteraciones que hay en las actividades finalmente realizadas, respecto a la planificación inicial, fueron debidas a: desajustes en la temporalización de la actividad (*“el grupo de 2º Bachillerato no pudo asistir completo, tenían una obra de teatro en inglés e iban mal de tiempo para ensayar”... “es complicado parar actividades normales de una escuela para realizar una actividad no planeada a principio de año y fuera de currículo”... “por motivos de coordinación con los maestros y maestras tuve menos tiempo del que en un principio había planeado, y por ello no pude explicar a los alumnos/as tan detalladamente como quería la actividad”... “una de las sesiones de educación infantil se ha tenido de posponer para después de las vacaciones de Navidad por motivos de organización interna forzados por causas ajenas al centro”*); preferencias del alumnado por hacer la actividad en tableta en vez de en PC's fijos (*“los alumnos de 4º Primaria utilizaron la aplicación ‘The Foos’³²⁶ en vez de la de ‘Angry Birds’, por petición de los propios alumnos, que se sentían más motivados por el uso de las tablets”*); y cambios o reajustes en el tutorial inicialmente seleccionado por el profesor, a la luz de su dificultad inadecuada (*“en dos cursos de Primaria, algunos alumnos utilizaron la aplicación ‘The Foos’ en vez de la de ‘Angry Birds’ como medida de adaptación, ya que el nivel exigido por ‘Angry Birds’ les superaba”*)

Se confirma que no se lleva a cabo ninguna evaluación formal (*“there was no formal evaluation, but rather anecdotal observation of student performance through the session; given I usually had less than an hour with a group, no extended assessment or evaluation was possible”*). La evaluación se limita a procedimientos informales de observación, o seguimiento del *‘tracking’* ofrecido por las propias plataformas-tutoriales (*“el sistema de evaluación ha consistido en registrar aquellos que han conseguido llegar al final de la prueba, los 20 problemas, e imprimirles el certificado firmado y sellado por el centro”*)

Con respecto a la **Pregunta 2** del cuestionario [*“¿Cuáles son los aspectos MÁS POSITIVOS que desea resaltar del desarrollo de la CSEW en su centro?”*], que cubre el objetivo de investigación **O_{2PI}**: se señala fundamentalmente la alta motivación e interés mostrado por los estudiantes (*“el grado de motivación que he visto en los alumnos de todas las edades”... “además se lo han pasado bien aprendiendo a programar”... “les encanta y aprenden; qué mejor que realizar actividades motivadoras que hacen a gusto”... “los alumnos se han implicado mucho en la resolución de los problemas planteados; destaco sobre todo la motivación”... “sorprendentemente, durante las 4 sesiones no hubo ningún alumno descolgado o aburrido”*), debidos principalmente a la gamificación de los tutoriales y a la accesibilidad que ofrecen los lenguajes visuales de programación (*“los videos introductorios han ayudado mucho pero sobre todo creo que la presentación en forma de juego, con*

³²⁶ <http://thefoos.com/>

bloques tipo lego en vez de código puro y duro han conseguido un nivel de atención muy notable”... “acercar la programación a los alumnos desde pequeños y que vean que son bloques sencillos los que construyen los programas más grandes”), así como adaptabilidad de los tutoriales al nivel de cada estudiante (*“también me ha gustado mucho la adaptabilidad de los programas a los distintos niveles de los alumnos lo que ha permitido que en algunos grupos pudieran utilizar distintas aplicaciones a la vez”*)

Como resultado, los profesores indican que los estudiantes acaban con una buena percepción de autoeficacia (*“algunos alumnos han descubierto que ‘no se les da tan mal la informática’”*), incluidas las chicas (*“ha despertado el interés de algunos en la programación; y las chicas han descubierto que ellas también ‘pueden’”*). Y, como consecuencia, algunos centros se plantean ofrecer algún curso o currículo más avanzado y prolongado sobre ‘coding’ (*“I think it at least opened the door for people to think more about the place of programming at the school”... “ante todo destacaría el interés del alumnado ante esta actividad; los alumnos/as han querido continuar el aprendizaje y por ello nos estamos planteando desde el centro el proseguir con la misma durante el resto del curso escolar”*)

Otro aspecto positivo resaltado es la posibilidad de trabajar colaborativamente mediante la metodología ‘pair programming’³²⁷ o ‘programación por parejas’, especialmente en Primaria (*“el alumnado trabajó por parejas y la coordinación entre las parejas fue muy elevada”... “el aprendizaje en programación y el trabajo colaborativo”*), con mentores de cursos superiores (*“high school programming students helped lower school students with programming”*); y los beneficios de poder compartir los proyectos realizados (*“con lo fácil que es compartir los proyectos se pueden fomentar los valores del software libre”*). También se señala la buena acogida de profesores y padres (*“it was also great to share these sessions with classroom teachers, who do not have previous experience programming; parent response was also very positive”*). Finalmente, se pone en valor el hecho de que se puede participar en el evento ‘a coste cero’ (*“el bajo o nulo coste de esta actividad también es un punto positivo”*)

Con respecto a la **Pregunta 3** del cuestionario [*“¿Cuáles son los aspectos MÁS NEGATIVOS que desea resaltar del desarrollo de la CSEW en su centro? ¿Qué propuestas de mejora se plantea para los mismos?”*], que cubre el objetivo de investigación **O₂P_m**: se señalan pocos o ningún aspecto negativo (*“no he experimentado nada que se pudiera considerar negativo”... “no tengo aspectos negativos”... “no he encontrado aspectos negativos”*). De los pocos aspectos negativos reportados, éstos básicamente tienen que ver con incidencias de tipo técnico, no imputables al evento sino a las deficiencias de conexión de los centros (*“en algunas horas tuvimos problemas con el tiempo de respuesta, no sé si sería por los servidores o por el ancho de banda de la línea del centro; tenían que recargar la página o volver al ejercicio anterior y hacerlo de nuevo”... “a veces Internet no funciona bien y se desesperan entre actividad y actividad”*); o con dificultades en la temporalización del evento (*“nuestra adaptación a la hora de 60 minutos”... “no contar con un día para hacer este tipo de actividades y tener que ‘robar’ clase a otros profesores para juntar a un número suficiente de alumnos”... “the timing of the activities was difficult for middle and high school teachers and*

³²⁷ Code.org promueve la programación por parejas o ‘pair programming’ a través de vídeos como <https://youtu.be/vgkahOzFH2Q> (Code.org, 2014)

students, as many were finishing semester activities and preparing for exams”... “haber de desarrollarlo dentro de una misma semana ya que es un tiempo muy justo y a nivel de organización del centro”... “la falta de tiempo para hacer una introducción apropiada”).

Consistentemente, las propuestas de mejora tienen que ver con una mayor planificación del evento para el curso siguiente, con objeto de involucrar mejor al resto del claustro del centro (*“para el próximo curso se tratará de involucrar al claustro desde septiembre”... “para el próximo curso se propondrá tener dos sesiones, una teórica en la que aclarar todo lo que a los alumnos se les ocurra sobre programación o sobre la herramienta a utilizar, y otra para realizar las actividades”*); y/o con tener previstas actividades de emergencia por si falla la conexión a Internet durante el evento (*“el próximo año intentaría descargar e instalar en local las actividades”*)

Con respecto a la **Pregunta 4** del cuestionario [*“Relate alguna/s anécdota relevante o caso excepcional ocurrido a lo largo de la CSEW (p.e. algún estudiante que haya demostrado especial talento, algún estudiante que haya mostrado una conducta inusual, etc...)”*], que cubre el objetivo de investigación **O_{2Pn}**: los profesores relatan cierta casuística que nos resulta de gran interés. Por ejemplo, casos de ‘talentos especiales’ por su excelente ejecución durante las sesiones de ‘coding’ (*“efectivamente he podido observar alumnos con un talento especial; se dio el caso de dos alumnos de 2º de Primaria que aparentemente completaron los 20 niveles de Blockly en inglés en 45 minutos.”... “nos llamó la atención un alumno de sexto de primaria que en 20 minutos acabó los 20 problemas propuestos, sin haber programado anteriormente”*)³²⁸.

También se refieren varios casos de estudiantes (o grupos) habitualmente inquietos, disruptivos o con problemas de atención, que muestran un compromiso sostenido con las tareas de ‘coding’ (*“en general los estudiantes más inquietos y que menos aguantan las clases magistrales se adaptan muy bien a este tipo de instrucción basada en problemas y práctica”... “también quiero destacar que un alumno diagnosticado con TDAH permaneció atento y concentrado durante toda la sesión, trabajando por parejas y sin tener conductas disruptivas”... “la mayor sorpresa ha sido el buen comportamiento y alta motivación de dos de los grupos más disruptivos del centro; una sorpresa muy grata”... “un muy mal estudiante y además comportamental, de 2º de la ESO, repetidor, que ha conseguido llegar al final de la prueba; al día siguiente estaba en mi despacho solicitando saber cuándo se le haría entrega del certificado de aprovechamiento; ¡inusual!”*); o estudiantes con retrasos en otras áreas curriculares, que muestran interés y facilidad con el ‘coding’ (*“se han dado casos de dos estudiantes con un significativo desfase curricular avanzar mucho más rápido que otros compañeros sin dificultades”*)

Por otro lado, se producen casos de chicas que manifiestan prejuicios al respecto de su baja o nula capacidad para programar, prejuicios que se ven confrontados y desterrados tras su iniciación al ‘coding’ (*“una alumna entró preguntándome si era obligatorio hacer la actividad ya que a ella se le daba muy mal la informática; le indiqué que no podía cerrarse de antemano, y que con lo joven que era, si no cambiaba el chip tenía un gran problema, porque hoy en día nadie puede dar la espalda a la informática; le invité a intentarlo, y terminó emocionada y con otra visión totalmente distinta*

³²⁸ Como referencia para el lector, y siguiendo las propias orientaciones que ofrece la web del evento, hay que decir que esos 20 niveles están diseñados para ser completados en alrededor de una hora por estudiantes de finales de Primaria o inicios de Secundaria.

sobre su capacidad con el uso de los ordenadores”); y casos de parejas de estudiantes que programaron juntos (*‘pair programming’*) de manera excepcionalmente positiva (“*asimismo he podido observar una pareja de chico y chica en 4º de Primaria trabajando juntos en plena colaboración y armonía para conseguir el objetivo*”... “*la primera sesión la hicimos individual; la segunda la hicimos en pareja y se dieron cuenta que mejoraban significativamente sus resultados; tuvimos que hacer todas las demás en parejas; intentaba cambiar las parejas pero eran bastante reticentes ya que ‘con este compañero vamos más rápido’ eran sus palabras*”)

Finalmente, también se reporta alguna diferencia de género observada (“*las chicas prestaban más atención a las animaciones y los chicos al juego en sí*”); y casos de voluntarios casuales que ejercieron de mentores durante el evento con alumnos más pequeños, que manifiestan su deseo de continuar regularmente con sus tareas de mentorización (“*a couple of students volunteered to regularly help out after school and one volunteered to help out a lower school class during the school day*”)

Con respecto a la **Pregunta 5** del cuestionario [“*Desde la fundación Code.org (promotora de la CSEW) se propone continuar y profundizar con la introducción del ‘coding’ en las escuelas: <http://code.org/learn/beyond> ¿Tienen intención de continuar en los próximos meses con nuevas actividades curriculares relacionadas con el ‘coding’? ¿Cuáles?*”], que cubre el objetivo de investigación **O_{2Pñ}**: fundamentalmente se detectan iniciativas de proseguir con currículos más sistemáticos y avanzado de aprendizaje del ‘coding’ a final de la etapa de Primaria y Secundaria, con el curso trimestral ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) (“*tengo intención de proseguir con las dos clases de 5º de primaria, siguiendo el programa ‘Introducción a la Informática K-8’*... “*el curso pasado ya hicimos con los alumnos de 4º de ESO los cursos de 20 horas de coding de code.org*”... “*sí, los alumnos de mi grupo de 4º ESO están realizando el curso de 20 horas de programación por bloques que ofrece code.org*”), y también intenciones de profundización con Scratch³²⁹ en Secundaria (“*todos los años trabajamos con Scratch en 2º de la ESO, el segundo trimestre*”... “*con mis alumnos de 3º de ESO seguimos un poco con Scratch y después haremos programación de Android a través de App Inventor*”... “*usaremos Scratch con los alumnos de 4º de ESO; la idea es dedicarle al menos 20 horas*”), y con AppInventor³³⁰ en Secundaria y Bachillerato (“*con los de 4º de ESO aproveché para saltar a programación en Android con AppInventor*”... “*este curso hemos incorporado al currículo de Tecnología de la Información en 1º de bachillerato la programación de aplicaciones móviles Android con AppInventor y los alumnos están entusiasmados*”)

5.4.4. Relativos al cuestionario para estudiantes

Pasamos ahora a dar respuesta a los objetivos **O_{2E*}**, que se abordan a través del “*CSEW: Cuestionario para estudiantes*” (Anexo C). Recordemos que el “*CSEW: Cuestionario para estudiantes*”³³¹ consta de un total de 12 preguntas, siendo una combinación de preguntas de elección múltiple (preguntas 1, 2, 6 y 9), preguntas de selección múltiple o casillas de verificación (preguntas 3 y 10), de escalas tipo Likert (preguntas 4, 5, 7 y 8), y de preguntas abiertas (preguntas 11 y 12).

³²⁹ <https://scratch.mit.edu/>

³³⁰ <http://appinventor.mit.edu/explore/>

³³¹ <https://goo.gl/c3j5xq>

Convertidas las de selección múltiple en variables ‘dummy’ y dicotomizadas las respuestas abiertas, el cuestionario en su conjunto consta de un total de 35 elementos, arrojando un notable valor de fiabilidad $\alpha = 0,810$.

Con respecto a la **Pregunta 1** del cuestionario [“*P1. ¿Habías oído hablar de la ‘programación informática’ (‘coding’) antes de esta actividad de la ‘Computer Science Education Week (CSEW)’?*”], que aborda el objetivo de investigación O_{2Ea} : en la Tabla 5.21 (y Figura D.5 del anexo) se muestran los resultados para el total de la muestra. Tal y como puede observarse, algo más de la mitad de la muestra (53%) había oído hablar de la programación informática o ‘coding’ con anterioridad al evento HoC.

Tabla 5.21. Frecuencias y porcentajes de respuesta a la pregunta 1 del cuestionario para estudiantes

<i>P1. ¿Habías oído hablar de la 'programación informática' ('coding') antes de esta actividad de la CSEW?</i>					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No	702	42,2	47,0	47,0
	Sí	793	47,7	53,0	100,0
	Total	1495	90,0	100,0	
Perdidos	Sistema	167	10,0		
Total		1662	100,0		

Con respecto a posibles diferencias en la respuesta a esta pregunta según el sexo, se encuentran diferencias muy significativas ($\chi^2 = 22,108$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,122$; $p_{(C)} = 0,000 \ll 0,05$). Se detalla en la Tabla 5.22 (y la Figura D.6 del anexo). Tal y como puede verse, el 59,0% de los chicos declara haber oído hablar del ‘coding’ frente a sólo un 46,7% de las chicas.

Tabla 5.22. Tabla de contingencia Pregunta 1 * Sexo

		<i>P1. ¿Habías oído hablar de la 'programación informática' ('coding') antes de esta actividad de la CSEW?</i>		Total	
		No	Sí		
Sexo	Chico	Recuento	312	449	761
		% dentro de Sexo	41,0%	59,0%	100,0%
Sexo	Chica	Recuento	370	324	694
		% dentro de Sexo	53,3%	46,7%	100,0%
Total		Recuento	682	773	1455
		% dentro de Sexo	46,9%	53,1%	100,0%

Con respecto a posibles diferencias en la respuesta a esta pregunta según la etapa educativa, también se encuentran diferencias muy significativas ($\chi^2 = 17,927$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,109$; $p_{(C)} = 0,000 \ll 0,05$). Tal y como puede verse en la Tabla 5.23 (y la Figura D.7 del anexo), el 46,3% de los estudiantes de Primaria declara haber oído hablar del ‘coding’ frente a un 53,9% en Secundaria, 65,3% en Bachillerato y un 66,7% en Formación Profesional.

Tabla 5.23. Tabla de contingencia Pregunta 1 * Etapa educativa

		<i>P1. ¿Habías oído hablar de la 'programación informática' ('coding') antes de esta actividad de la CSEW?</i>		Total	
		No	Sí		
Etapa	Primaria	Recuento	226	195	421
		% dentro de Etapa	53,7%	46,3%	100,0%
	Secundaria	Recuento	415	486	901
		% dentro de Etapa	46,1%	53,9%	100,0%
	Bachillerato	Recuento	51	96	147
		% dentro de Etapa	34,7%	65,3%	100,0%
	FP	Recuento	5	10	15
		% dentro de Etapa	33,3%	66,7%	100,0%
Total	Recuento	697	787	1484	
	% dentro de Etapa	47,0%	53,0%	100,0%	

Una síntesis de cómo evoluciona el porcentaje de sujetos que declara haber oído hablar del 'coding' en función del sexo y a lo largo de las distintas etapas, puede verse en la Tabla 5.24 y Figura 5.15. La brecha de género a favor de los chicos en su conocimiento del término 'programación informática' o 'coding' alcanza su mayor magnitud en Educación Secundaria (diferencia +13,3% a favor de los chicos), en donde es estadísticamente muy significativa (Tabla 5.25).

Tabla 5.24. Evolución en la Pregunta 1 por sexo a lo largo de las distintas etapas educativas

	% que declara haber oído hablar del 'coding' antes del evento			
	Primaria (n=402)	Secundaria (n=888)	Bachillerato (n=143)	FP (n=15)
Chicos	50,5%	60,3%	70,1%	66,7%
Chicas	41,2%	47,0%	60,6%	66,7%
Brecha de género a favor de los chicos	+9,3%	+13,3%	+9,5%	0,0%

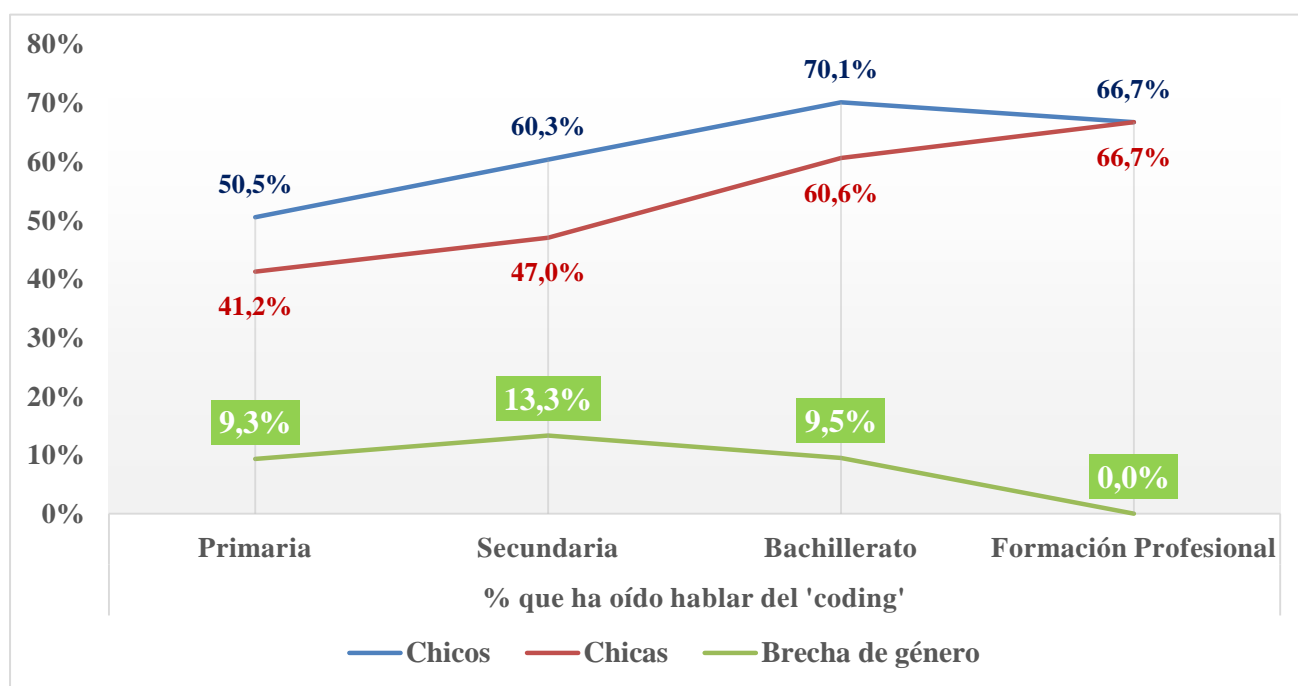


Figura 5.15. Evolución en la Pregunta 1 por sexo a lo largo de las distintas etapas educativas

Tabla 5.25. Resumen de significatividad de las diferencias por sexo en la Pregunta 1, diferenciado por etapas educativas

	N	Chi-Cuadrado (χ^2)	Signific. $P_{(\chi^2)}$	Coefficiente de Contingencia (C)	Signific. $P_{(C)}$
Primaria	402	3,523	,061	,093	,061
Secundaria	888	15,820**	,000	,132**	,000
Bachillerato	143	1,431	,232	,100	,232
Formación Profesional	15	,000	1,000	,000	1,000

**Estadísticamente muy significativo $p < 0,01$

Adicionalmente, encontramos diferencias significativas en la respuesta a la Pregunta 1 según la tipología de centro educativo al que pertenece el estudiante ($\chi^2 = 9,071$; $p_{(\chi^2)} = 0,028 < 0,05$; $C = 0,078$; $p_{(C)} = 0,028 < 0,05$). En este caso, un 56,8% de los estudiantes que provienen de centros privados o concertados ‘multietapa’ declaran haber oído hablar del ‘coding’, frente a un 52,0% de los estudiantes de institutos públicos (IES) o un 45,6% de estudiantes de colegios públicos (CEIP). Se detalla a continuación en la Tabla 5.26 (y Figura D.8 del anexo).

Tabla 5.26. Tabla de contingencia Pregunta 1 * Tipología de Centro

		<i>P1. ¿Habías oído hablar de la 'programación informática' ('coding') antes de esta actividad de la CSEW?</i>		Total	
		No	Sí		
Tipología	CEIP - Colegio Público Infantil y/o Primaria	Recuento	129	108	237
		% dentro de Tipología	54,4%	45,6%	100,0%
	IES - Instituto Público Secundaria, Bachill. y/o FP	Recuento	330	357	687
		% dentro de Tipología	48,0%	52,0%	100,0%
	Centro Privado o Concertado Multietapa	Recuento	227	298	525
		% dentro de Tipología	43,2%	56,8%	100,0%
	Centro Municipal de Educación No Formal	Recuento	9	14	23
		% dentro de Tipología	39,1%	60,9%	100,0%
	Total	Recuento	695	777	1472
		% dentro de Tipología	47,2%	52,8%	100,0%

Con respecto a la **Pregunta 2** del cuestionario [“P2. ¿Habías ‘programado’ alguna vez antes de esta actividad?”], que aborda el objetivo de investigación O_{2Eb} : en la Tabla 5.27 (y la Figura D.9 del anexo) se muestran los resultados para el total de la muestra. Tal y como puede observarse, algo menos de la mitad de la muestra (43,0%) había programado con anterioridad al evento.

Tabla 5.27. Frecuencias y porcentajes de respuesta a la pregunta 2 del cuestionario para estudiantes

		<i>P2. ¿Habías 'programado' alguna vez antes de esta actividad?</i>			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No	857	51,6	57,0	57,0
	Sí	647	38,9	43,0	100,0
	Total	1504	90,5	100,0	
Perdidos	Sistema	158	9,5		
Total		1662	100,0		

Con respecto a posibles diferencias en la respuesta a esta pregunta según el sexo, se encuentran diferencias muy significativas ($\chi^2 = 22,078$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,122$; $p_{(C)} = 0,000 \ll 0,05$). Se detalla en la Tabla 5.28 (y la Figura D.10 del anexo). Tal y como puede verse, el 49,2% de los chicos declara haber programado con anterioridad al evento frente a sólo un 37,0% de las chicas.

Tabla 5.28. Tabla de contingencia Pregunta 2 * Sexo

		<i>P2. ¿Habías 'programado' alguna vez antes de esta actividad?</i>		Total	
		No	Sí		
Sexo	Chico	Recuento	389	376	765
		% dentro de Sexo	50,8%	49,2%	100,0%
	Chica	Recuento	440	258	698
		% dentro de Sexo	63,0%	37,0%	100,0%
Total	Recuento	829	634	1463	
	% dentro de Sexo	56,7%	43,3%	100,0%	

Con respecto a posibles diferencias en la respuesta a esta pregunta según la etapa educativa, también se encuentran diferencias muy significativas ($\chi^2 = 47,653$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,176$; $p_{(C)} = 0,000 \ll 0,05$). Tal y como puede verse en la Tabla 5.29 (y la Figura D.11 del anexo), el 30,1% de los estudiantes de Primaria declara haber programado anteriormente frente a un 46,7% en Secundaria, 54,7% en Bachillerato y un 73,3% en Formación Profesional.

Tabla 5.29. Tabla de contingencia Pregunta 2 * Etapa educativa

		<i>P2. ¿Habías 'programado' alguna vez antes de esta actividad?</i>		Total	
		No	Sí		
Etapa	Primaria	Recuento	295	127	422
		% dentro de Etapa	69,9%	30,1%	100,0%
	Secundaria	Recuento	484	424	908
		% dentro de Etapa	53,3%	46,7%	100,0%
	Bachillerato	Recuento	67	81	148
		% dentro de Etapa	45,3%	54,7%	100,0%
	FP	Recuento	4	11	15
		% dentro de Etapa	26,7%	73,3%	100,0%
	Total	Recuento	850	643	1493
		% dentro de Etapa	56,9%	43,1%	100,0%

Una síntesis de cómo evoluciona el porcentaje de sujetos que declara haber programado con anterioridad a la actividad (experiencia previa con el 'coding') en función del sexo y a lo largo de las distintas etapas, puede verse en la Tabla 5.30 y Figura 5.16. En este caso, la brecha de género a favor de los chicos en su experiencia previa con el 'coding' alcanza su mayor magnitud en Bachillerato (diferencia +24,4% a favor de los chicos); tanto en Secundaria como en Bachillerato estas diferencias por sexo son estadísticamente muy significativa (Tabla 5.31).

Tabla 5.30. Evolución en la Pregunta 2 por sexo a lo largo de las distintas etapas educativas

	% que declara haber programado con anterioridad (experiencia previa con el 'coding')			
	Primaria (n=402)	Secundaria (n=895)	Bachillerato (n=144)	FP (n=15)
Chicos	33,3%	52,4%	66,2%	75,0%
Chicas	27,0%	40,8%	41,8%	66,7%
Brecha de género a favor de los chicos	+6,3%	+11,6%	+24,4%	+8,3%

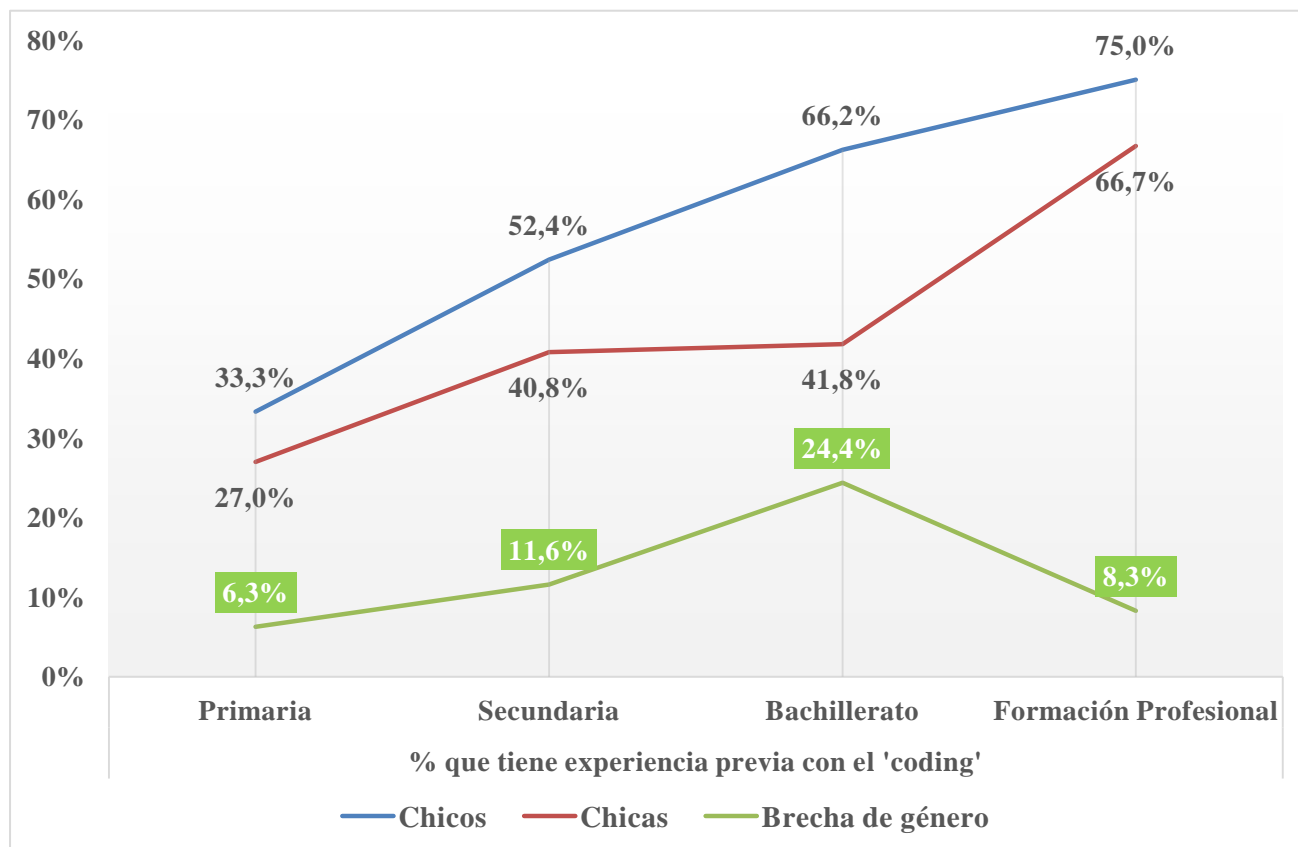


Figura 5.16. Evolución en la Pregunta 2 por sexo a lo largo de las distintas etapas educativas

Tabla 5.31. Resumen de significatividad de las diferencias por sexo en la Pregunta 2, diferenciado por etapas educativas

	N	Chi-Cuadrado (χ^2)	Signific. $P_{(\chi^2)}$	Coefficiente de Contingencia (C)	Signific. $P_{(C)}$
Primaria	402	1,939	,164	,069	,164
Secundaria	895	12,202**	,000	,116**	,000
Bachillerato	144	8,643**	,003	,238**	,003
Formación Profesional	15	,085	,770	,075	,770

**Estadísticamente muy significativo $p < 0,01$

Adicionalmente, encontramos diferencias muy significativas en la respuesta a la Pregunta 2 según la tipología de centro educativo al que pertenece el estudiante ($\chi^2 = 16,220$; $p_{(\chi^2)} = 0,001 \ll 0,05$; $C = 0,104$; $p_{(C)} = 0,001 \ll 0,05$). En este caso, un 48,2% de los estudiantes que provienen de centros privados o concertados ‘multietapa’ declaran haber programado anteriormente, frente a un 42,7% de los estudiantes de institutos públicos (IES) o un 32,8% de estudiantes de colegios públicos (CEIP). Se detalla a continuación en la Tabla 5.32 (y Figura D.12 del anexo).

Tabla 5.32. Tabla de contingencia Pregunta 2 * Tipología de centro

		<i>P2. ¿Habías 'programado' alguna vez antes de esta actividad?</i>		Total	
		No	Sí		
Tipología	CEIP - Colegio Público Infantil y/o Primaria	Recuento	160	78	238
		% dentro de Tipología	67,2%	32,8%	100,0%
	IES - Instituto Público Secundaria, Bachillerato y/o FP	Recuento	396	295	691
		% dentro de Tipología	57,3%	42,7%	100,0%
	Centro Privado o Concertado Multietapa	Recuento	274	255	529
		% dentro de Tipología	51,8%	48,2%	100,0%
	Centro Municipal de Educación No Formal	Recuento	12	11	23
		% dentro de Tipología	52,2%	47,8%	100,0%
	Total	Recuento	842	639	1481
		% dentro de Tipología	56,9%	43,1%	100,0%

Con respecto a la **Pregunta 3** del cuestionario [“P3. En caso afirmativo a la pregunta anterior, ¿con qué lenguaje/s informático/s de programación tienes experiencia?”], que aborda el objetivo de investigación O_{2Ec} : las respuestas se detallan a continuación en la Tabla 5.33 y Figura 5.17.

Tabla 5.33. Penetración de los lenguajes de programación entre los estudiantes

Lenguaje de programación	Frecuencia	Penetración (%) sobre el total de la muestra (N=1662)	Penetración (%) sobre la sub-muestra con experiencia previa en programación (N=647)
HTML*	233	14,02%	36,01%
Scratch** ³³²	200	12,03%	30,91%
JavaScript*	160	9,63%	24,73%
CSS*	74	4,45%	11,44%
Python*	60	3,61%	9,27%
Lightbot**	46	2,77%	7,11%
Blockly**	36	2,17%	5,56%
PHP*	32	1,93%	4,95%
Ruby*	23	1,38%	3,55%
Basic/Visual Basic/Small Basic/Just Basic**	14	0,84%	2,16%
jQuery*	12	0,72%	1,85%
AppInventor**	11	0,66%	1,70%
Game Maker Language**	8	0,48%	1,24%
Java**	7	0,42%	1,08%
Batch**	6	0,36%	0,93%
C++**	6	0,36%	0,93%
Lego/LegoWeDo**	6	0,36%	0,93%

³³² Algunos de los lenguajes visuales de programación, como Scratch, Blockly, Lightbot o AppInventor, son simultáneamente lenguajes de programación, y plataformas de aprendizaje del mismo. En su momento, nos inclinamos a no listarlos explícitamente entre las opciones de esta Pregunta 3 (sobre lenguajes), y sí listarlos en la Pregunta 10 (sobre plataformas)

Lenguaje de programación	Frecuencia	Penetración (%) sobre el total de la muestra (N=1662)	Penetración (%) sobre la sub-muestra con experiencia previa en programación (N=647)
Moway**	4	0,24%	0,62%
Pivot**	4	0,24%	0,62%
Arduino Programming Language**	3	0,18%	0,46%
Assembler**	3	0,18%	0,46%
C**	3	0,18%	0,46%
Crocodile**	3	0,18%	0,46%
HTML5**	3	0,18%	0,46%
Keynote**	3	0,18%	0,46%
C#**	2	0,12%	0,31%
LittleBits**	2	0,12%	0,31%
Blender**	1	0,06%	0,15%
Delphi**	1	0,06%	0,15%
Kodu**	1	0,06%	0,15%

* Lenguaje de programación explícitamente listado entre las posibles opciones a marcar en la Pregunta 3

** Lenguajes no explícitamente listados, que se mencionaron por los sujetos en la categoría 'Otros'

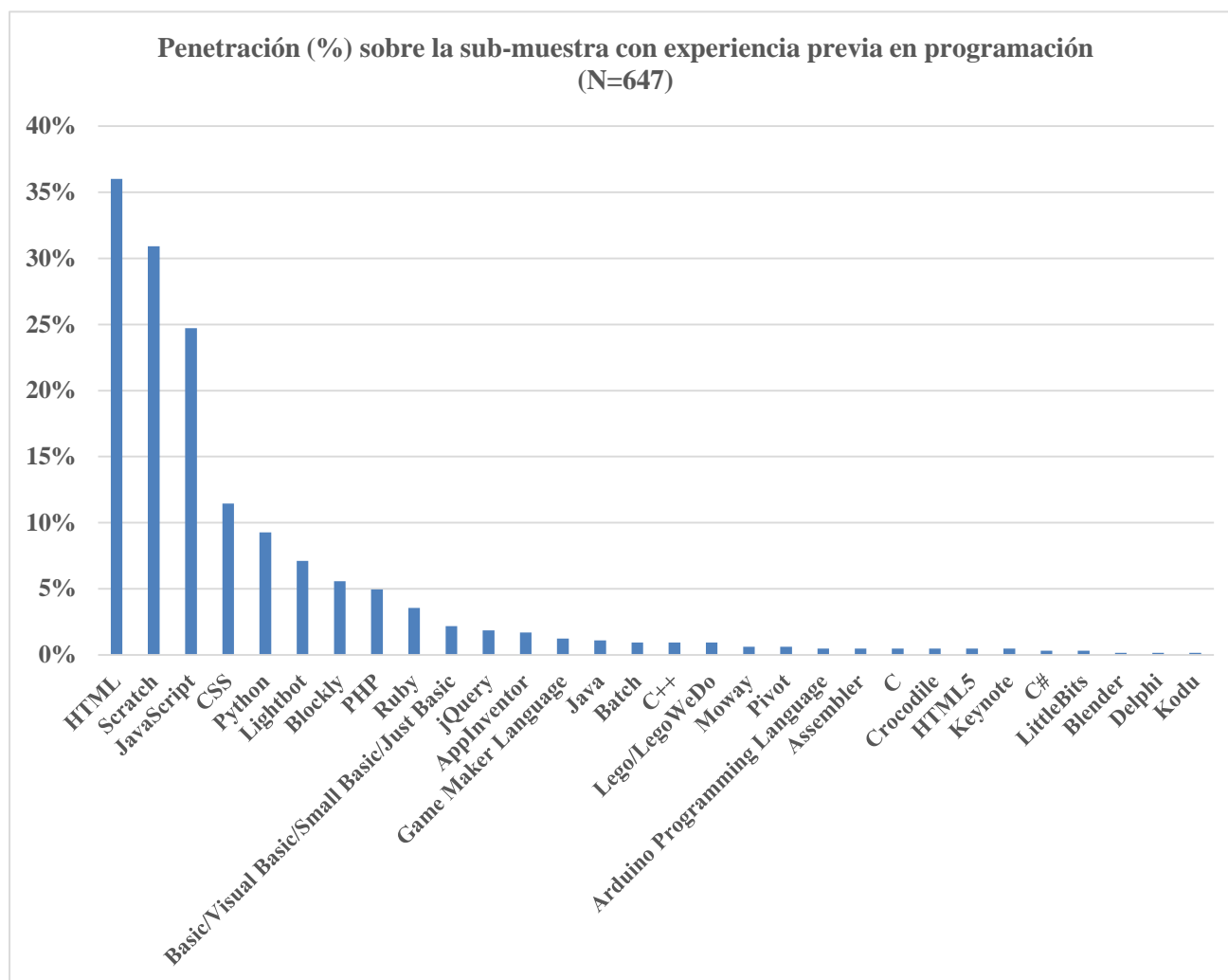


Figura 5.17. Penetración de los lenguajes de programación entre los estudiantes

Centrándonos en los 7 lenguajes de programación con más penetración entre los estudiantes (todos con un índice de penetración >2% sobre el total de la muestra, y >5% sobre la sub-muestra de estudiantes con experiencia previa en programación), éstos pueden clasificarse según la taxonomía que ya enunciamos en el epígrafe 2.4.1 de la parte teórica, como: 2 lenguajes textuales ‘puros’ (JavaScript y Python), 2 lenguajes textuales de ‘etiquetas, marcado o estilo’ o ‘pseudolenguajes’ (HTML y CSS), y 3 lenguajes visuales (2 ‘por bloques’: Blockly y Scratch; y 1 ‘por flechas’: Lightbot). En la siguiente Tabla 5.34, y Figuras 5.18 y 5.19, se muestran los índices de penetración de cada lenguaje sobre el total de la muestra, diferenciado por sexos. Tal y como puede observarse, en los lenguajes textuales ‘puros’ el índice de penetración es $\approx 2,5$ -3,0 veces superior entre los chicos; $\approx 1,5$ veces en el caso de los ‘pseudolenguajes’ textuales de etiquetas y estilo; ≈ 1 (igual penetración en chicos y en chicas) en el caso de los lenguajes visuales.

Tabla 5.34. Índices de penetración según lenguajes sobre el total de la muestra, y diferenciado por sexos

	Lenguajes Textuales Puros		Lenguajes de Etiquetas, Marcado y Estilo (Pseudolenguajes)		Lenguajes Visuales (por bloques o flechas)		
	Python	JavaScript	HTML	CSS	Lightbot	Scratch	Blockly
Chico	5,30%	13,50%	17,60%	5,60%	2,60%	11,70%	1,80%
Chica	1,70%	5,40%	10,60%	3,50%	2,80%	12,90%	2,60%
Factor de género (% chico) ÷ (% chica)	3,12	2,50	1,66	1,60	0,93	0,91	0,69

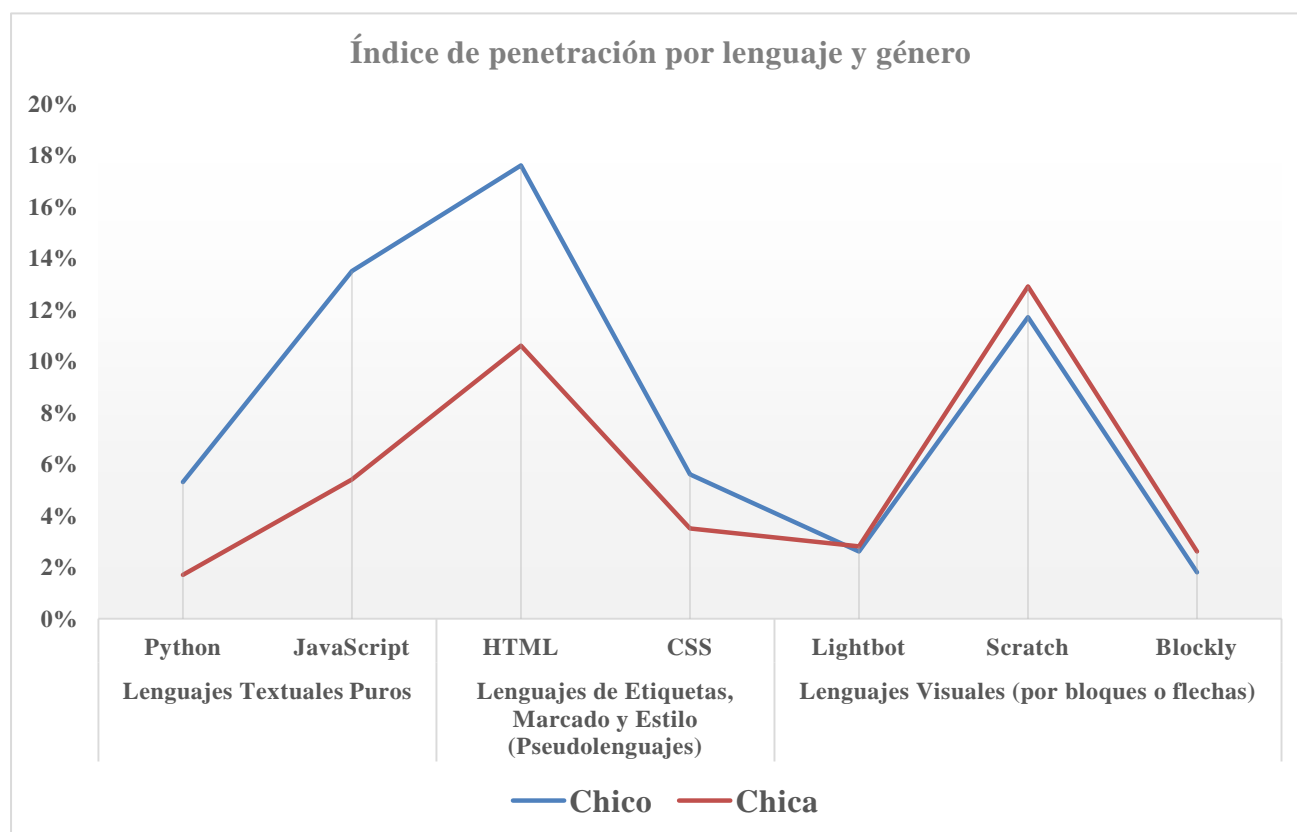


Figura 5.18. Índices de penetración de cada lenguaje sobre el total de la muestra, diferenciado por sexos

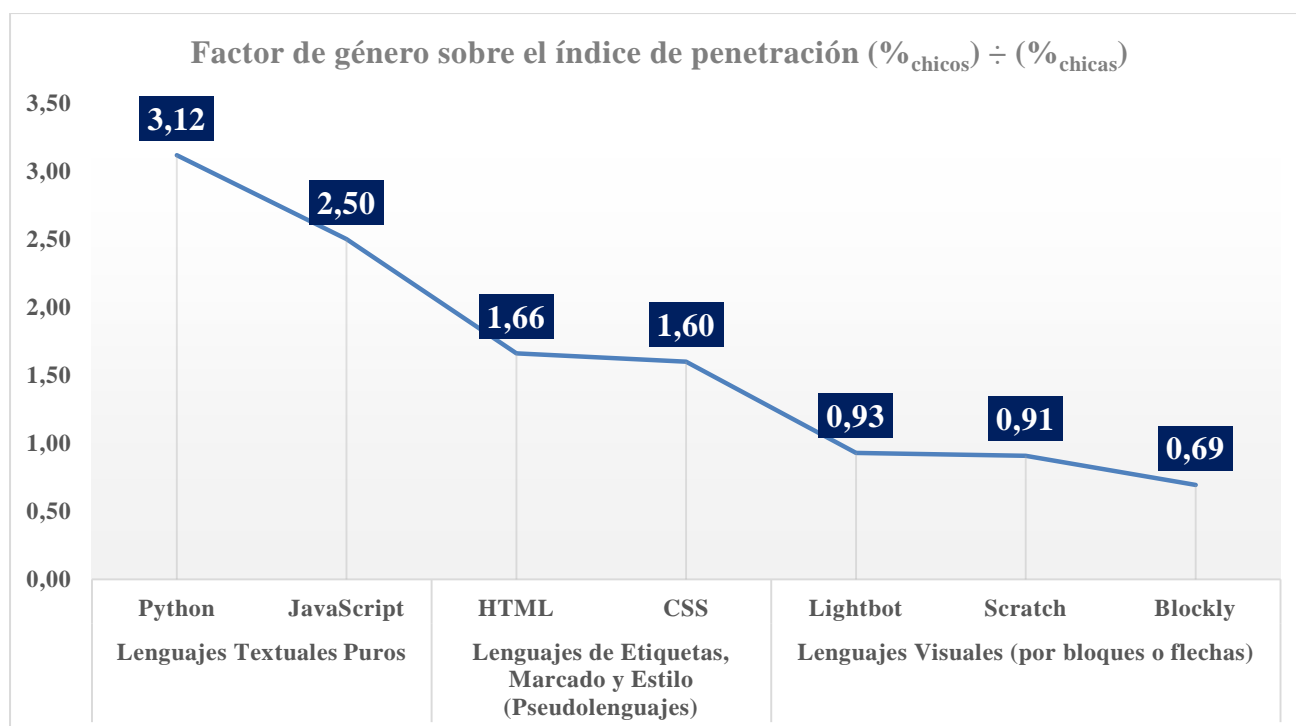


Figura 5.19. Factor de género sobre el índice de penetración $[(\%chicos) \div (\%chicas)]$, según lenguajes

Replicando el análisis, ahora por etapa educativa (Tabla 5.35, y Figura D.13 del anexo), comprobamos que, en síntesis: los lenguajes visuales por flechas tienen su mayor índice de penetración en Educación Primaria; los lenguajes visuales por bloques tienen su mayor índice de penetración en Educación Secundaria; los ‘pseudolenguajes’ textuales de etiquetas y estilo en Bachillerato; y los lenguajes textuales puros en Formación Profesional. Se observa pues una trayectoria evolutiva en lo referente a los lenguajes de programación utilizados por los estudiantes, en la línea que ya se señaló en el epígrafe 2.4.1 de la parte teórica de esta tesis doctoral.

Tabla 5.35. Índices de penetración según lenguajes sobre el total de la muestra, y diferenciado por etapa educativa

	Lenguajes Textuales Puros		Lenguajes de Etiquetas, Marcado y Estilo (Pseudolenguajes)		Lenguajes Visuales		
	Python	JavaScript	CSS	HTML	Por Bloques	Por Flechas	Lightbot
Primaria	1,4%	4,0%	0,6%	1,8%	6,7%	0,2%	8,7%
Secundaria	3,6%	11,6%	4,2%	16,4%	16,6%	3,2%	0,2%
Bachillerato	8,9%	12,1%	16,6%	38,2%	3,2%	2,5%	0,0%
FP	20,0%	33,3%	26,7%	33,3%	6,7%	0,0%	0,0%

Con respecto a la **Pregunta 4** del cuestionario [“P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?”], que aborda el objetivo de investigación **O_{2Ed}**; se trata de una escala tipo Likert de 5 puntos (desde 1=Nada a 5=Mucho), que interroga acerca de las siguientes seis aptitudes o habilidades: pensamiento lógico; creatividad; resolución de problemas; trabajo en equipo; persistencia-perseverancia; y capacidad de atención. En la siguiente Tabla 5.36 se detallan los estadísticos descriptivos principales para las seis aptitudes-habilidades, y en la Figura 5.20 se presenta una comparativa entre las medias de cada una de ellas.

Tabla 5.36. Estadísticos descriptivos relativos a la Pregunta 4 del cuestionario para estudiantes

		<i>P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?</i>					
		Pensamiento Lógico	Creatividad	Resolución de Problemas	Trabajo en Equipo	Persistencia-Perseverancia	Capacidad de Atención
N	Válido	1464	1435	1429	1439	1422	1441
	Perdidos	198	227	233	223	240	221
Media		3,66	3,61	3,49	3,01	3,44	3,78
Mediana		4,00	4,00	4,00	3,00	4,00	4,00
Moda		4	4	4	1	3	5
Desv. estándar		1,135	1,199	1,233	1,506	1,248	1,221
Varianza		1,288	1,438	1,520	2,267	1,557	1,492
Asimetría		-,736	-,556	-,432	-,046	-,381	-,800
Curtosis		-,065	-,580	-,758	-1,437	-,822	-,313
Mínimo		1	1	1	1	1	1
Máximo		5	5	5	5	5	5

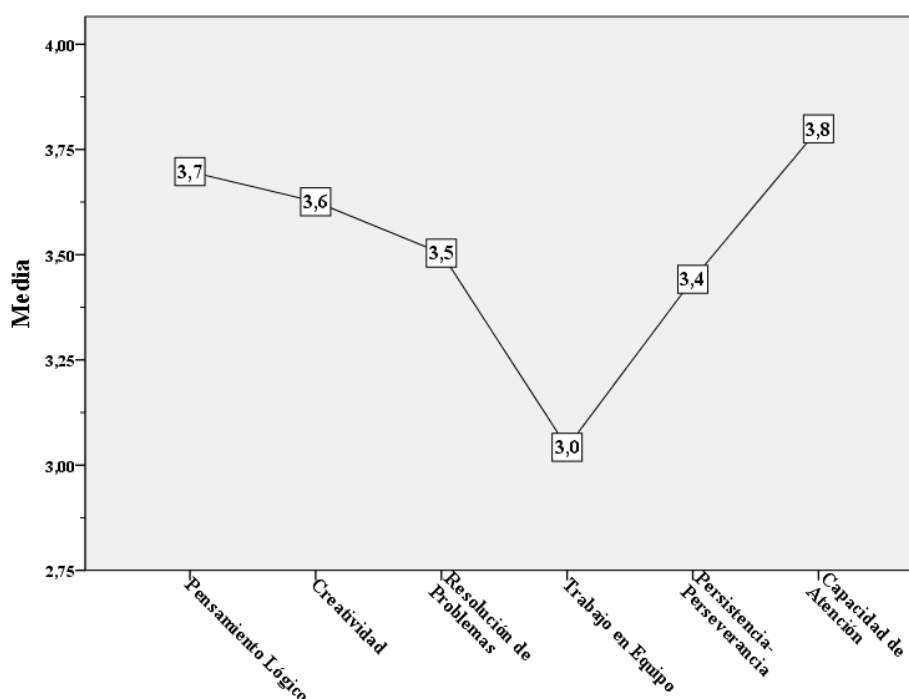


Figura 5.20. Comparativa de medias en las distintas aptitudes-habilidades

Tal y como puede observarse, la aptitud que arroja mayor media (y moda) es la ‘capacidad de atención’, seguida de cerca por ‘pensamiento lógico’, ‘creatividad’, ‘resolución de problemas’ y ‘persistencia-perseverancia’: todas ellas se sitúan en un nivel de valoración medio-alto, con mediana de 4 sobre 5; más lejos queda el ‘trabajo en equipo’ (se sitúa en un nivel de valoración medio, con mediana 3 sobre 5). En todas las aptitudes-habilidades aparecen distribuciones con asimetría negativa; a excepción del ‘trabajo en equipo’ que presenta una distribución casi simétrica (y casi bimodal en los valores extremos; siendo por tanto la variable con mayores índices de dispersión). Las distribuciones se presentan de manera detallada en las Tablas D.5 a D.10, y Figuras D.14 a D.19 del correspondiente anexo.

Con respecto a posibles diferencias por sexo en la media de las respuestas a la pregunta 4, aplicando la prueba *t* de Student no se encuentran diferencias estadísticamente significativas en ninguna de las seis aptitudes-habilidades interrogadas; para todas ellas $p_{(t)} > 0,05$ (Tabla D.11 del anexo).

Con respecto a posibles diferencias por etapa educativa en la media de las respuestas a la pregunta 4, aplicando un análisis de varianza (ANOVA), se encuentran diferencias muy significativas para todas las aptitudes-habilidades ($p_{(F)} = 0,000 \ll 0,05$), con valores de *F* especialmente altos para ‘trabajo en equipo’ ($F = 41,040$) y ‘capacidad de atención’ ($F = 33,058$). Se puede consultar con detalle en la Tabla D.12 del anexo. Tal y como puede apreciarse en la Figura 5.21, en todas las aptitudes-habilidades hay una valoración superior en Primaria respecto a Secundaria y Bachillerato. Una posible explicación está en un estilo de respuesta más aquiescente de los sujetos de menor edad en este tipo de escalas Likert; otra posible explicación puede tener que ver con la menor experiencia previa en ‘coding’ de los sujetos de Primaria, que provocaría una percepción más saliente (más exagerada) de las aptitudes que desarrolla. En cualquier caso, y recenrándonos en nuestro objetivo **O_{2Ed}** de investigación, se detecta un patrón evolutivo diferente para ‘trabajo en equipo’ y ‘capacidad de atención’.

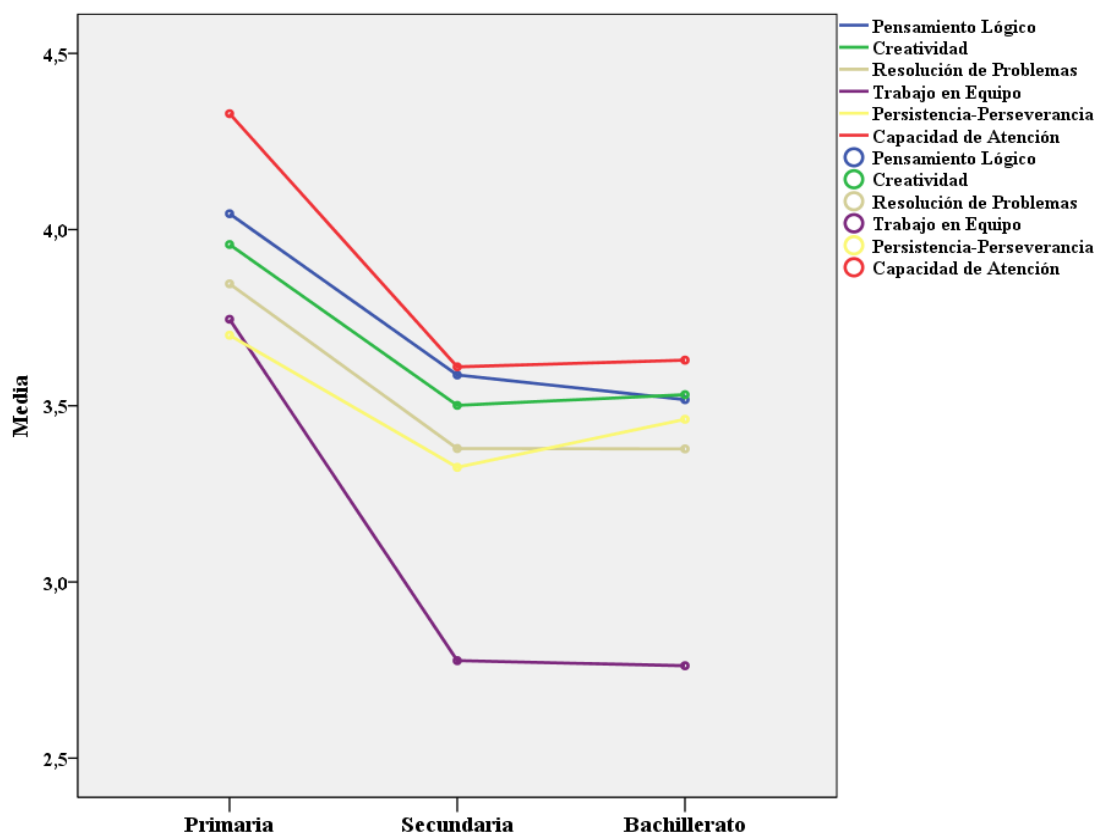


Figura 5.21. Diferencias por etapa educativa (se ha omitido FP por su escasa muestra) para la Pregunta 4

Con respecto a posibles diferencias en la media de las respuestas a la Pregunta 4, en función de tener experiencia previa, o no, en ‘coding’ (Pregunta 2); los resultados se muestran en la Tabla 5.37 y Figura 5.22. Vuelve a aparecer un patrón diferente para ‘trabajo en equipo’ y ‘capacidad de atención’, que son las dos únicas aptitudes-habilidades interrogadas en las que se encuentran diferencias estadísticamente significativas en función de tener o no experiencia previa en ‘coding’.

Más concretamente, la diferencia es significativa en el caso de ‘trabajo en equipo’ ($p_{(t)} = 0,011 < 0,05$), y muy significativa en el caso de ‘capacidad de atención’ ($p_{(t)} = 0,002 < 0,01$); si bien con tamaños del efecto ‘pequeños’ ($d < 0,20$).

Tabla 5.37. Diferencias según experiencia previa en ‘coding’ para la Pregunta 4

P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?	N	Media	Desviación estándar	t de Student	Significación de la diferencia $P_{(t)}$	Tamaño del efecto d de Cohen
Pensamiento Lógico	No	838	3,68	1,110	0,457	,648
	Sí	618	3,65	1,170		
Creatividad	No	815	3,62	1,197	0,502	,616
	Sí	612	3,59	1,203		
Resolución de problemas	No	815	3,52	1,228	0,856	,392
	Sí	606	3,46	1,237		
Trabajo en equipo	No	818	3,10	1,485	2,544*	,011
	Sí	613	2,90	1,526		
Persistencia-Perseverancia	No	809	3,45	1,247	0,325	,745
	Sí	605	3,43	1,252		
Capacidad de atención	No	819	3,87	1,186	3,107**	,002
	Sí	615	3,66	1,258		

* Significativo al nivel $p_{(t)} < 0,05$

** Significativo al nivel $p_{(t)} < 0,01$

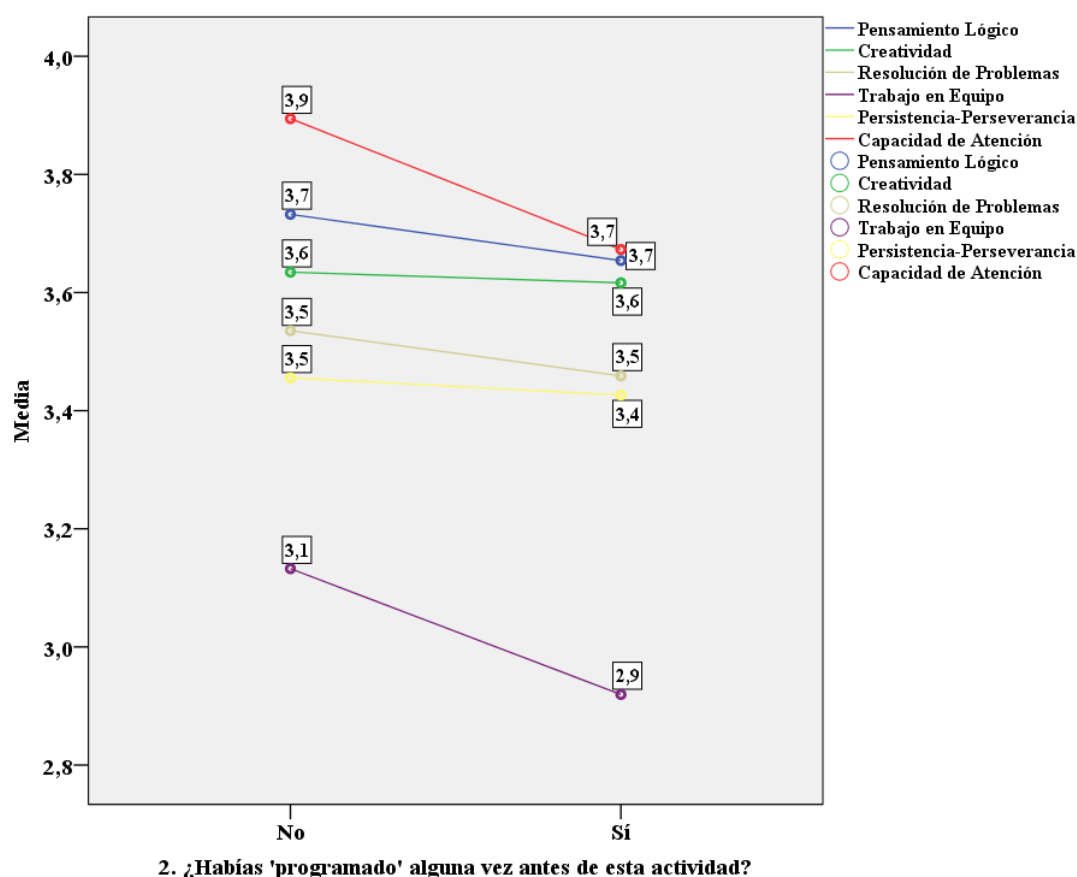


Figura 5.22. Diferencias según experiencia previa en ‘coding’ para la Pregunta 4

Pasando ahora a una perspectiva correlacional entre las 6 aptitudes-habilidades interrogadas en la Pregunta 4, en la Tabla 5.38 se muestra la correspondiente matriz de correlaciones. Todas las correlaciones resultantes son estadísticamente muy significativas, de signo positivo e intensidad moderada; oscilando entre valores $r = +0,577$ (correlación entre ‘pensamiento lógico’ y ‘resolución de problemas’) y $r = +0,301$ (correlación entre ‘pensamiento lógico’ y ‘trabajo en equipo’).

Tabla 5.38. Matriz de correlaciones entre las 6 aptitudes-habilidades interrogadas en la Pregunta 4

	(PL)	(C)	(RP)	(TE)	(PP)	(CA)
Pensamiento Lógico (PL)	1	,509**	,577**	,301**	,510**	,530**
Creatividad (C)		1	,452**	,393**	,414**	,500**
Resolución de Problemas (RP)			1	,331**	,520**	,509**
Trabajo en Equipo (TE)				1	,380**	,419**
Persistencia-Perseverancia (PP)					1	,518**
Capacidad de Atención (CA)						1

** Correlación muy significativa $p_{(r)} < 0,01$

Entendidas las 6 aptitudes-habilidades interrogadas en la Pregunta 4 como elementos de una escala, se calcula su fiabilidad como consistencia interna a través del estadístico Alfa de Cronbach, arrojando un valor notable de $\alpha = 0,830$. Dados estos niveles de correlación y fiabilidad, se entiende necesario y posible un análisis factorial exploratorio que se aproxime a la estructura latente en las 6 aptitudes-habilidades interrogadas.

En la Tabla D.13 del anexo se detallan las Pruebas KMO y Bartlett, que indican si la matriz de correlaciones es apta para ser factorizada (Gil Pascual, 2003): como ahí puede observarse, la medida de adecuación muestral KMO se aproxima a 1 ($KMO = 0,869 \approx 1$), y la prueba de esfericidad de Bartlett es altamente significativa ($p_{(\chi^2)} = 0,000 \ll 0,05$); y por tanto puede procederse al análisis factorial ulterior. Encontramos dos soluciones factoriales de interés, que resultan estadísticamente parsimoniosas a la par que interpretables y con suficiente fuerza explicativa.

a) Solución de 3 factores

En primer lugar, hallamos una solución de 3 factores que explica un 77,84% de la varianza total contenida en la matriz de correlaciones (Tablas 5.39 y 5.40)

Tabla 5.39. Comunalidades de las variables en la solución de 3 factores

	Inicial	Extracción
Pensamiento Lógico	1,000	,736
Creatividad	1,000	,899
Resolución de Problemas	1,000	,691
Trabajo en Equipo	1,000	,936
Persistencia-Perseverancia	1,000	,780
Capacidad de Atención	1,000	,628

Método de extracción: análisis de componentes principales³³³

³³³ Dada la naturaleza aproximativa y exploratoria de este primer estudio, se opta por un **análisis de componentes principales**, con objeto de maximizar la independencia de los componentes \approx factores. Aunque el análisis de componentes principales no supone un análisis factorial en sentido estricto, se ha señalado su bondad y uso habitual para este tipo de fines (Kline, 1994)

Tabla 5.40. Varianza total explicada por el modelo de 3 factores

Componente	Varianza total explicada								
	Autovalores iniciales			Sumas de extracción de cargas al cuadrado			Sumas de rotación de cargas al cuadrado		
	Total	% de varianza	% acumulado	Total	% de varianza	% acumulado	Total	% de varianza	% acumulado
1	3,321	55,347	55,347	3,321	55,347	55,347	2,189	36,483	36,483
2	,759	12,645	67,992	,759	12,645	67,992	1,320	21,997	58,480
3	,591	9,848	77,840	,591	9,848	77,840	1,162	19,360	77,840
4	,476	7,933	85,773						
5	,452	7,535	93,307						
6	,402	6,693	100,000						

Método de extracción: análisis de componentes principales.

En esta solución de 3 factores encontramos: a) un primer factor, compuesto por ‘persistencia-perseverancia’, ‘resolución de problemas’, ‘pensamiento lógico’ y ‘capacidad de atención’; que explica el 55,35% de la varianza total de la Pregunta 4, y que denominaremos ‘*el acto de programar en sí*’ (está compuesto por aptitudes-habilidades que son inherentes al acto de programar en sí, independientemente del entorno de programación utilizado y de la metodología de aprendizaje del ‘*coding*’ que se haya seguido); b) un segundo factor, compuesto por ‘creatividad’ (y con cierta carga de la variable ‘pensamiento lógico’); que explica el 12,65% de la varianza total, y que denominaremos ‘*el entorno-tarea de programación*’ (sería dependiente del entorno-tarea digital utilizado: por ejemplo, entornos poco creativos que plantean problemas cerrados o puzzles de programación como Code.org; frente a entornos altamente creativos como Scratch, en el que el estudiante debe codificar un proyecto abierto ‘desde cero’ u ‘*open-ended*’); y c) un tercer factor, compuesto por ‘trabajo en equipo’; que explica el 9,85% de la varianza total de la Pregunta 4, y que denominaremos ‘*la metodología de aprendizaje de la programación*’ (que sería dependiente de la metodología de aula marcada por el profesor, que puede incluir o no explícitamente la programación colaborativa = ‘*pair programming*’ o en solitario = ‘*single programming*’). Se detalla en las Tablas 5.41³³⁴ y 5.42, y en la Figura 5.23.

Tabla 5.41. Matriz factorial rotada para 3 factores

	Matriz de componente rotado ^a		
	1	2	3
Persistencia-Perseverancia	,818		
Resolución de Problemas	,768		
Pensamiento Lógico	,702	,493	
Capacidad de Atención	,593		
Creatividad		,886	
Trabajo en Equipo			,930

Método de extracción: análisis de componentes principales.

Método de rotación: Varimax con normalización Kaiser.

a. La rotación ha convergido en 6 iteraciones.

³³⁴ Para una interpretación más directa, en esta tabla se ordenan las variables de mayor a menor saturación en cada factor; y se dejan en blanco las cargas factoriales por debajo de 0,40 (Gil Pascual, 2003)

Tabla 5.42. Solución de 3 factores para la Pregunta 4

Factor	% Varianza explicada de la Pregunta 4
Factor 1: El Acto de Programar en sí: (Persistencia-Perseverancia + Resolución de Problemas + Pensamiento Lógico + Capacidad de Atención)	55,35%
Factor 2: El Entorno-Tarea de Programación (Creatividad... con lógica)	12,65%
Factor 3: La Metodología de Aprendizaje de la Programación (Trabajo en Equipo)	9,85%
Residual	22,16%
TOTAL	100%

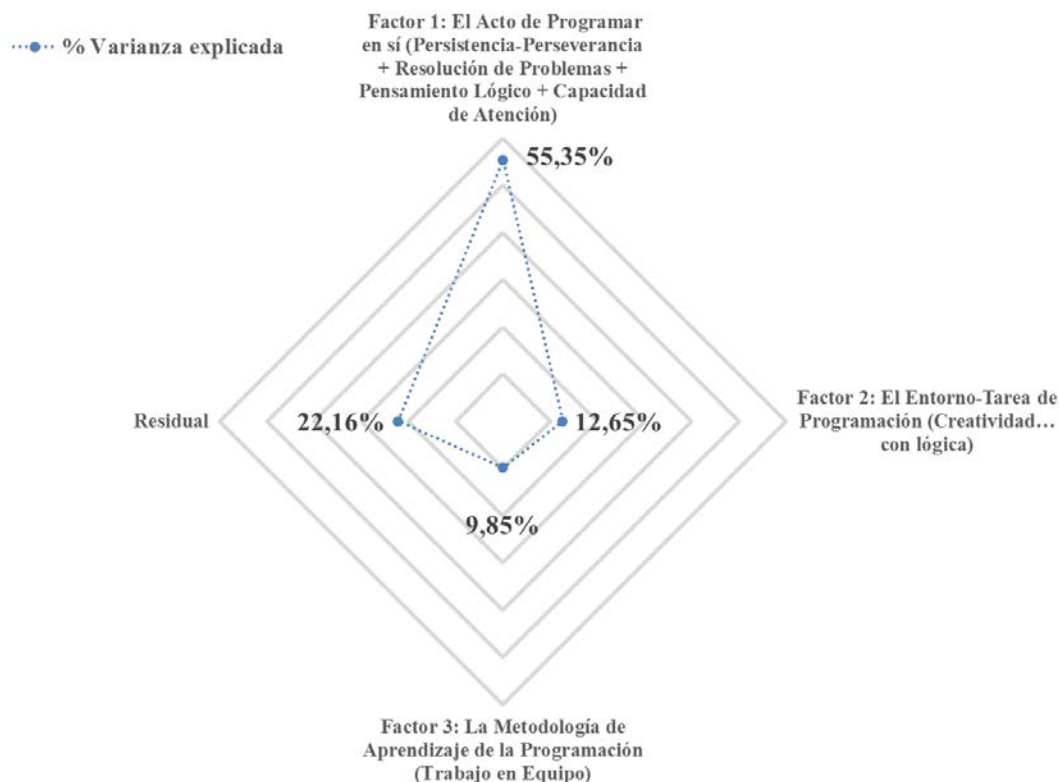


Figura 5.23. Solución de 3 factores para la Pregunta 4

b) Solución de 4 factores

Posteriormente, hallamos una solución de 4 factores que explica un 85,77% de la varianza total contenida en la matriz de correlaciones (Tablas 5.43 y 5.44)

Tabla 5.43. Comunalidades de las variables en la solución de 4 factores

	Comunalidades	
	Inicial	Extracción
Pensamiento Lógico	1,000	,746
Creatividad	1,000	,903
Resolución de Problemas	1,000	,881
Trabajo en Equipo	1,000	,982
Persistencia-Perseverancia	1,000	,826
Capacidad de Atención	1,000	,808

Método de extracción: análisis de componentes principales.

Tabla 5.44. Varianza total explicada por el modelo de 4 factores

Componente	Varianza total explicada								
	Autovalores iniciales			Sumas de extracción de cargas al cuadrado			Sumas de rotación de cargas al cuadrado		
	Total	% de varianza	% acumulado	Total	% de varianza	% acumulado	Total	% de varianza	% acumulado
1	3,321	55,347	55,347	3,321	55,347	55,347	1,478	24,628	24,628
2	,759	12,645	67,992	,759	12,645	67,992	1,424	23,735	48,363
3	,591	9,848	77,840	,591	9,848	77,840	1,206	20,093	68,456
4	,476	7,933	85,773	,476	7,933	85,773	1,039	17,317	85,773
5	,452	7,535	93,307						
6	,402	6,693	100,000						

Método de extracción: análisis de componentes principales

En esta solución de 4 factores encontramos: a) un primer factor, compuesto por ‘resolución de problemas’ y ‘pensamiento lógico’; que explica el 55,35% de la varianza total de la Pregunta 4, y que denominaremos ahora ‘*factor cognitivo*’; b) un segundo factor, compuesto por ‘persistencia-perseverancia’ y ‘capacidad de atención’; que explica el 12,65% de la varianza total, y que denominaremos ‘*factor motivacional-atencional*’; c) un tercer factor, compuesto por ‘creatividad’ (con cierta carga también en ‘pensamiento lógico’ y ‘capacidad de atención’); que explica el 9,85% de la varianza total, y que denominaremos ‘*factor creativo*’; y d) un cuarto factor, compuesto por ‘trabajo en equipo’; que explica el 7,93% de la varianza, y que denominaremos ‘*factor metodológico*’. Se muestra en detalle en las Tablas 5.45 y 5.46, y Figura 5.24.

Tabla 5.45. Matriz factorial rotada para 4 factores

	Matriz de componente rotado ^a			
	Componente			
	1	2	3	4
Resolución de Problemas	,876			
Pensamiento Lógico	,668		,413	
Persistencia-Perseverancia		,795		
Capacidad de Atención		,736	,458	
Creatividad			,874	
Trabajo en Equipo				,942

Método de extracción: análisis de componentes principales.

Método de rotación: Varimax con normalización Kaiser.

a. La rotación ha convergido en 6 iteraciones.

Tabla 5.46. Solución de 4 factores para la Pregunta 4

Factor	% Varianza explicada de la Pregunta 4
Factor 1 - Cognitivo (Resolución de Problemas + Pensamiento Lógico)	55,35%
Factor 2 – Motivacional/Atencional (Persistencia-Perseverancia + Capacidad de Atención)	12,65%
Factor 3 - Creativo (Creatividad... con atención y lógica)	9,85%
Factor 4 - Metodológico (Trabajo en Equipo)	7,93%
Residual	14,23%
TOTAL	100,00%

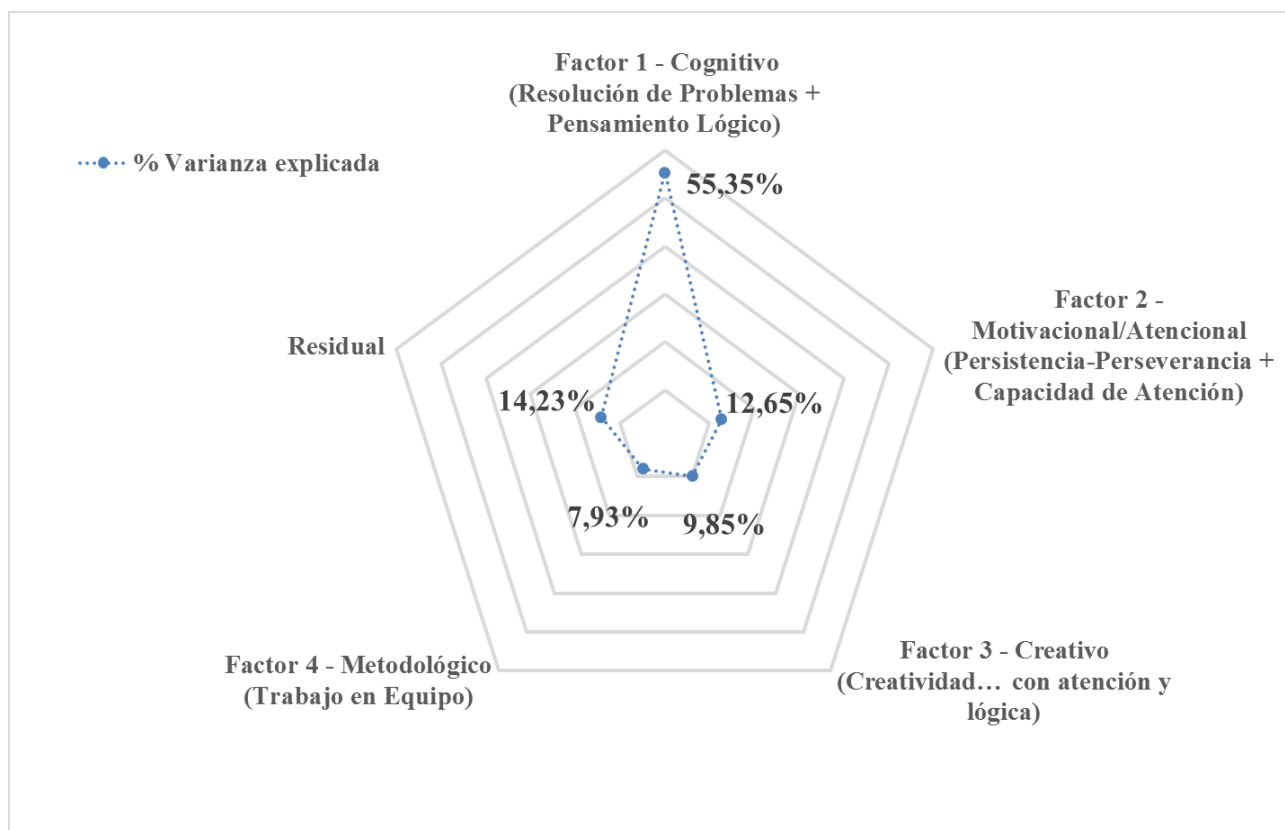


Figura 5.24. Solución de 4 factores para la Pregunta 4

Con respecto a la **Pregunta 5** del cuestionario [“P5. Tras tu experiencia con la actividad de la CSEW, ¿cómo dirías que se te da la 'programación informática' ('coding')?”], que aborda el objetivo de investigación **O_{2Ee}**, sobre percepción de ‘auto-eficacia’ de los estudiantes en tareas de ‘coding’ tras la celebración del evento: es una escala tipo Likert de 5 puntos (desde 1=Se me da muy mal a 5=Se me da muy bien). En la Tabla 5.47 se aportan los estadísticos descriptivos principales. Como puede observarse, globalmente la percepción de ‘auto-eficacia’ se sitúa en la parte medio-alta de la escala (mediana de 4 sobre 5).

Tabla 5.47. Estadísticos descriptivos relativos a la Pregunta 5 del cuestionario para estudiantes

<i>P5. Tras tu experiencia con la actividad de la CSEW, ¿cómo dirías que se te da la 'programación informática' ('coding')?</i>		
N	Válido	1440
	Perdidos	222
Media		3,83
Mediana		4,00
Moda		4
Desviación estándar		,951
Varianza		,904
Asimetría		-,950
Curtosis		1,142
Mínimo		1
Máximo		5

En la Tabla 5.48 (y Figura D.20 del anexo) se detalla la distribución de frecuencias de las respuestas a la Pregunta 5. Tan sólo el 6,9% de los estudiantes afirma que el ‘coding’ se les da ‘mal’ o ‘muy mal’ tras la actividad realizada durante ‘La Hora del Código’; el 22,9% afirma que ‘regular’; y un 70,2% afirma que se les da ‘bien’ o ‘muy bien’.

Tabla 5.48. Distribución de frecuencias en la Pregunta 5 del cuestionario para estudiantes

<i>P5. Tras tu experiencia con la actividad de la CSEW, ¿cómo dirías que se te da la 'programación informática' ('coding')?</i>					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Se me da muy mal	56	3,4	3,9	3,9
	Se me da mal	43	2,6	3,0	6,9
	Se me da regular	330	19,9	22,9	29,8
	Se me da bien	674	40,6	46,8	76,6
	Se me da muy bien	337	20,3	23,4	100,0
	Total	1440	86,6	100,0	
Perdidos	Sistema	222	13,4		
Total		1662	100,0		

Con respecto a la **Pregunta 6** del cuestionario [“P6. ¿Has conseguido completar el 100% de la actividad propuesta?”], que aborda el objetivo de investigación **O_{2Ef}**: las frecuencias de respuesta se muestran en la Tabla 5.49 (y Figura D.21 en anexo). Tal y como puede observarse, aproximadamente 2 de cada 3 estudiantes (64,8%) afirma que pudo completar plenamente la actividad propuesta en su centro durante el evento; mientras que 1 de cada 3 afirma que no pudo hacerlo (35,2%).

Tabla 5.49. Distribución de frecuencias en la Pregunta 6 del cuestionario para estudiantes

<i>P6. ¿Has conseguido completar el 100% de la actividad propuesta?</i>					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No	504	30,3	35,2	35,2
	Sí	928	55,8	64,8	100,0
	Total	1432	86,2	100,0	
Perdidos	Sistema	230	13,8		
Total		1662	100,0		

Pasamos ahora a hacer un análisis conjunto de las preguntas 5 y 6, dado que son consecutivas en el cuestionario y están enunciadas siguiendo una secuencia lógica; con la particularidad e interés de que la pregunta 5 [“P5. Tras tu experiencia con la actividad de la CSEW, ¿cómo dirías que se te da la 'programación informática' ('coding')?”]³³⁵, interroga acerca de una percepción subjetiva del estudiante; mientras que la pregunta 6 [“P6. ¿Has conseguido completar el 100% de la actividad propuesta?”]³³⁶, lo hace acerca de un hecho objetivo.

Tomada la pregunta 6 como una variable ‘dummy’ (0=no; 1=sí), encontramos una correlación estadísticamente muy significativa, de signo positivo y de intensidad moderadamente baja ($r = +0,226$; $p_{(r)} = 0,000 \ll 0,05$), con la pregunta 5. Existe pues correlación entre el hecho de haber completado o no la actividad propuesta con la percepción de autoeficacia subsiguiente.

³³⁵ Para las próximas tablas y figuras se abreviará como “P5. Autoeficacia ‘coding’”

³³⁶ Para las próximas tablas y figuras se abreviará como “P6. Completar actividad”

De manera gráfica, en la Figura 5.25 se muestra cómo varía la distribución en las respuestas a la pregunta 5 de autoeficacia, en función de haber o no completado la actividad propuesta.

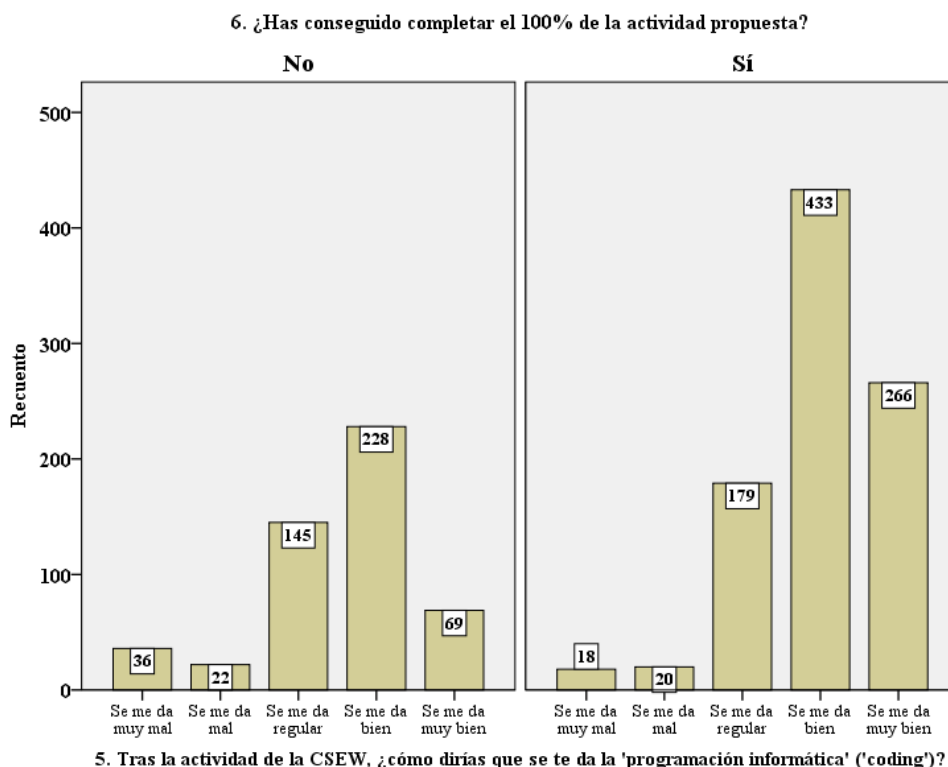


Figura 5.25. Distribución de frecuencias para la Pregunta 5, en función de la respuesta a la Pregunta 6

En una aproximación complementaria a este mismo asunto, aplicando una prueba *t* de Student (Tabla 5.50), encontramos una diferencia muy significativa entre las medias de autoeficacia en función de haber o no completado la actividad ($t = -8,309$; $p_{(t)} = 0,000 \ll 0,05$); con un tamaño del efecto asociado $d = 0,47$, que puede considerarse un efecto ‘moderado’.

Tabla 5.50. Prueba de diferencia de medias en la Pregunta 5, en función de la respuesta a la Pregunta 6

	P6. Completar actividad	N	Media	Desviación estándar	<i>t</i>	$P_{(t)}$	Tamaño del efecto
P5. Autoeficacia 'coding'	No	500	3,54	1,023	-8,309**	,000	0,47
	Sí	916	3,99	,866			

** Significativo al nivel $p_{(t)} < 0,01$

Al analizar el comportamiento de las respuestas a las preguntas 5 y 6 en función del sexo, encontramos diferencias estadísticamente muy significativas a favor de los chicos, aunque de tamaño ‘pequeño’ ($d \approx 0,20$). Se detalla en la siguiente Tabla 5.51 (y Figuras D.22 y D.23 del anexo)

Tabla 5.51. Diferencias por sexo en las preguntas 5 y 6

	Sexo	N	Media	Desviación estándar	<i>t</i>	$P_{(t)}$	Tamaño del Efecto
P5. Autoeficacia 'coding'	Chico	733	3,94	,966	4,445**	,000	0,23
	Chica	671	3,72	,909			
P6. Completar actividad	Chico	732	0,69	,461	3,796**	,000	0,19
	Chica	665	0,60	,491			

** Significativo al nivel $p_{(t)} < 0,01$

Por tanto, las chicas afirman que consiguen acabar la actividad HoC en menor medida que los chicos; y además manifiestan una menor percepción de autoeficacia posterior en ‘coding’. Es plausible que esta diferencia sea, al menos parcialmente, debida a la menor experiencia previa en ‘coding’ (Pregunta 2) de las chicas. De hecho, las respuestas a la Pregunta 2 [“P2. ¿Habías ‘programado’ alguna vez antes de esta actividad?”³³⁷], correlacionan muy significativamente con las respuestas a las preguntas 5 y 6 (Tabla 5.52); y en las medias de éstas encontramos diferencias estadísticamente muy significativas a favor de los sujetos con experiencia previa en ‘coding’ (Tabla 5.53 y Figura D.24 del anexo)

Tabla 5.52. Correlaciones de la Pregunta 2 con las preguntas 5 y 6

	<i>P5. Autoeficacia ‘coding’</i>	<i>P6. Completar actividad</i>
	Correlación de Pearson	,129**
	Sig. (bilateral)	,000
P2. Experiencia previa ‘coding’	N	1434

** Correlación muy significativa $p(r) < 0,01$

Tabla 5.53. Diferencias en las preguntas 5 y 6, en función de la experiencia previa en ‘coding’ (Pregunta 2)

	<i>P2. Experiencia previa ‘coding’</i>	N	Media	Desviación estándar	t	P(t)	Tamaño del Efecto
P5. Autoeficacia ‘coding’	No	821	3,72	1,012	-5,045**	,000	0,27
	Sí	613	3,97	,846			
P6. Completar actividad	No	818	,60	,491	-4,802**	,000	0,25
	Sí	611	,72	,451			

** Significativo al nivel $p(t) < 0,01$

En esta línea, si volvemos a analizar las diferencias en la Pregunta 6 en función del sexo, pero controlando ahora la experiencia previa en ‘coding’, entonces dichas diferencias se tornan en estadísticamente no significativas ($F_{\text{sexo}*\text{experiencia previa}} = 0,607$; $p(F_{\text{sexo}*\text{experiencia previa}}) = 0,436 > 0,05$). El MANOVA se detalla en la siguiente Tabla 5.54.

Tabla 5.54. MANOVA sobre la Pregunta 6, en función del sexo y la experiencia previa en ‘coding’

Pruebas de los efectos inter-sujetos					
Variable dependiente: P6. ¿Has conseguido completar el 100% de la actividad propuesta?					
Origen	Suma de cuadrados tipo III	gl	Media cuadrática	F	Sig.
Modelo corregido	7,911 ^a	3	2,637	11,811**	,000
Intersección	573,708	1	573,708	2569,590**	,000
Sexo	2,073	1	2,073	9,286**	,002
Experiencia previa ‘coding’	4,657	1	4,657	20,860**	,000
Sexo * Experiencia previa ‘coding’	,136	1	,136	,607	,436
Error	310,566	1391	,223		
Total	903,000	1395			
Total corregida	318,477	1394			

a. R cuadrado = ,025 (R cuadrado corregida = ,023)

** Significativo al nivel $p(F) < 0,01$

³³⁷ Para las próximas tablas y figuras se abreviará como “P2. Experiencia previa ‘coding’”

También encontramos diferencias estadísticamente muy significativas en función de la etapa educativa, tanto para la Pregunta 5 ($F = 14,838$; $p_{(F)} = 0,000 \ll 0,05$) como para la Pregunta 6 ($F = 12,656$; $p_{(F)} = 0,000 \ll 0,05$). Se detalla a continuación en las Figuras 5.26 y 5.27 (y Tabla D.14 en el anexo)

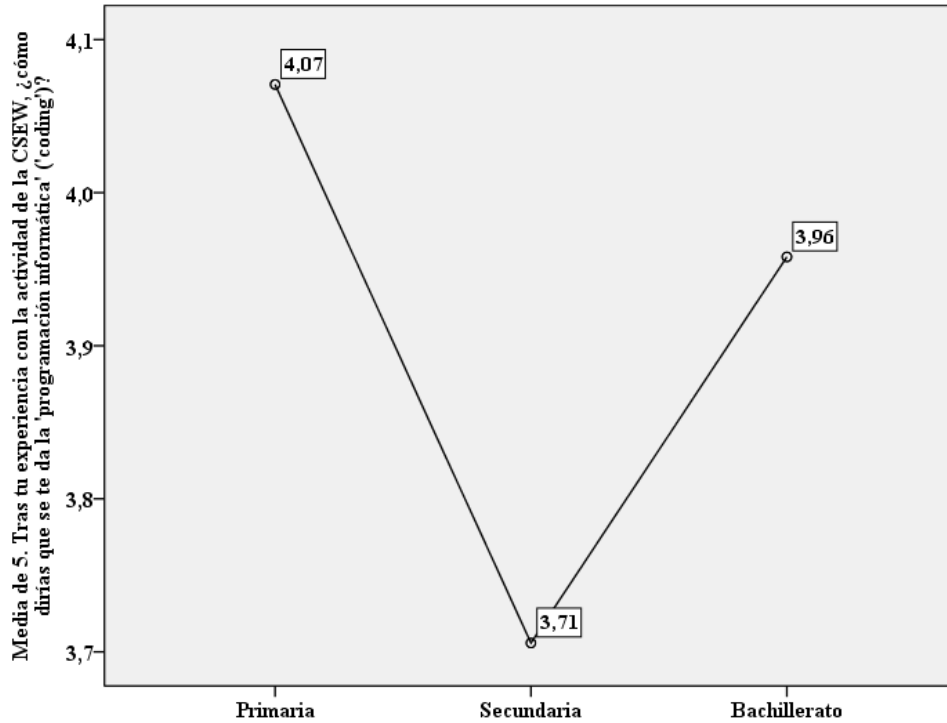


Figura 5.26. Media en la Pregunta 5 según etapa educativa (se omite FP por su escasa muestra)

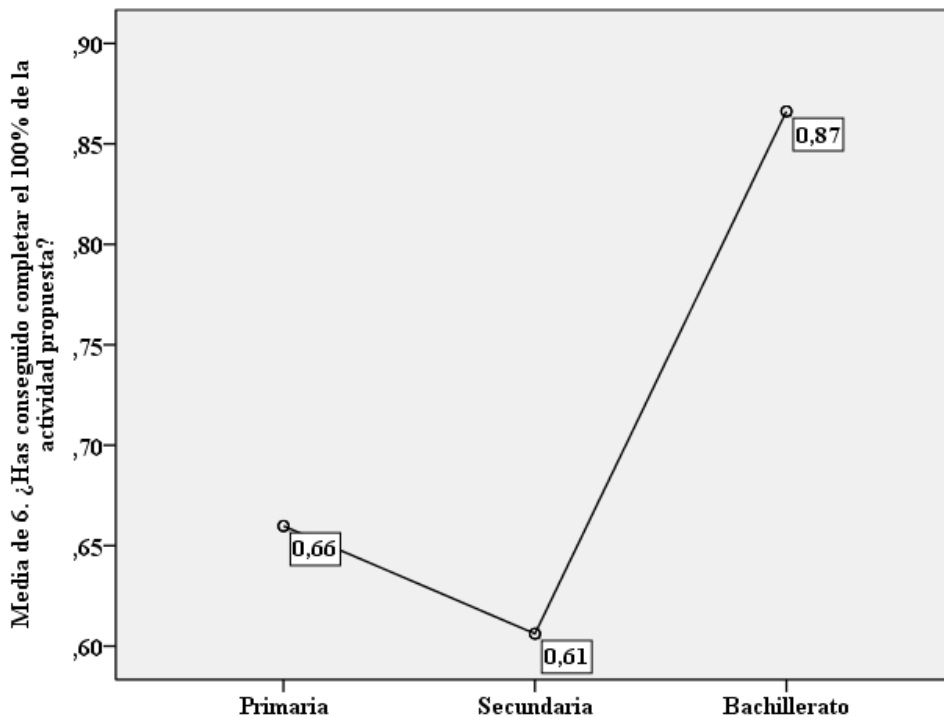


Figura 5.27. Media en la Pregunta 6 según etapa educativa (se omite FP por su escasa muestra)

Finalmente, para cerrar el análisis conjunto de las preguntas 5 y 6, cruzamos ahora las respuestas a dichas preguntas con las que se dieron a la Pregunta 4. En la Tabla 5.55 se muestran los coeficientes de correlación correspondientes. Se observan correlaciones estadísticamente muy significativas, de signo positivo e intensidad moderadamente baja, entre todas las variables; destacando la correlación que se produce entre la percepción de que el ‘coding’ contribuye a desarrollar el pensamiento lógico y la percepción de autoeficacia en ‘coding’ tras la actividad ($r = +0,279$)

Tabla 5.55. Correlaciones de las preguntas 5 y 6, con la pregunta 4

<i>P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?</i>						
	Pensamiento Lógico	Creatividad	Resolución de Problemas	Trabajo en Equipo	Persistencia-Perseverancia	Capacidad de Atención
<i>P5. Autoeficacia ‘coding’</i>	,279**	,224**	,233**	,158**	,243**	,240**
<i>P6. Completar actividad</i>	,087**	,120**	,112**	,073**	,071**	,085**

** Correlación muy significativa $p(r) < 0,01$

Como era de esperar, atendiendo a la anterior estructura correlacional entre las variables, encontramos diferencias estadísticamente muy significativas en todas las aptitudes-habilidades interrogadas en la Pregunta 4, tanto en función de la Pregunta 5 (Figura 5.28 y Tabla D.15 del anexo) como en función de la Pregunta 6 (Figura 5.29 y Tabla 5.56)

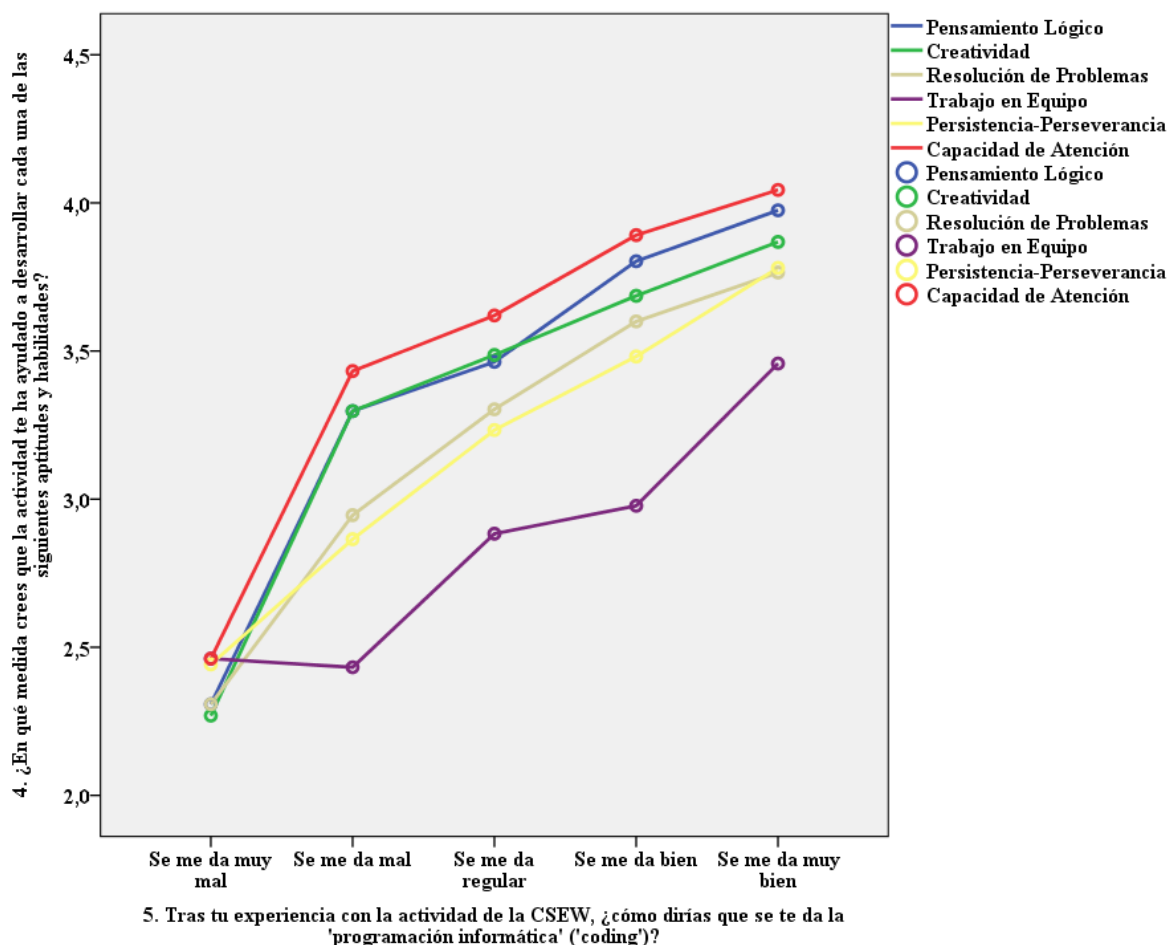


Figura 5.28. Medias de las respuestas a la Pregunta 4, en función de la Pregunta 5

Tal y como puede observarse, en la medida que el sujeto tiene una mayor percepción de autoeficacia tras la actividad, tiende a atribuir al ‘coding’ una mayor capacidad para desarrollar sus distintas aptitudes-habilidades; especialmente aquellas que tienen que ver con dimensiones internas del sujeto: pensamiento lógico ($F = 33,678$; $p_{(F)} = 0,000 \ll 0,05$), capacidad de atención ($F = 25,040$; $p_{(F)} = 0,000 \ll 0,05$), creatividad ($F = 22,591$; $p_{(F)} = 0,000 \ll 0,05$), resolución de problemas ($F = 22,375$; $p_{(F)} = 0,000 \ll 0,05$), y persistencia-perseverancia ($F = 21,775$; $p_{(F)} = 0,000 \ll 0,05$). En menor medida con el trabajo en equipo ($F = 10,735$; $p_{(F)} = 0,000 \ll 0,05$), de corte más contextual.

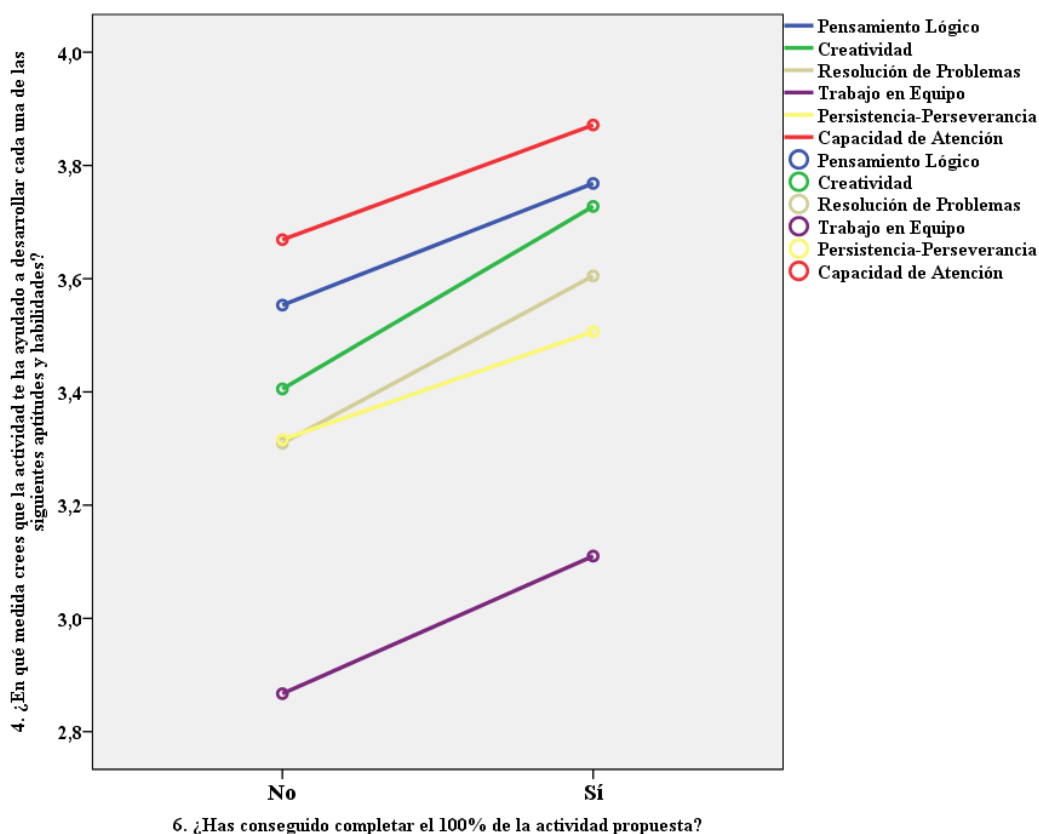


Figura 5.29. Medias de las respuestas a la Pregunta 4, en función de la Pregunta 6

Tabla 5.56. Diferencias en la Pregunta 4, en función de la Pregunta 6

P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar...?	P6. Completar actividad	N	Media	Desviación estándar	t	p(t)	Tamaño del Efecto d de Cohen																																																								
Pensamiento Lógico	No	498	3,53	1,151	-3,282**	,001	0,19																																																								
	Sí	915	3,74	1,119				Creatividad	No	485	3,40	1,220	-4,514**	,000	0,25	Sí	900	3,70	1,172	Resolución de Problemas	No	492	3,30	1,253	-4,204**	,000	0,24	Sí	889	3,60	1,215	Trabajo en Equipo	No	490	2,85	1,481	-2,724**	,007	0,15	Sí	900	3,08	1,517	Persistencia-Perseverancia	No	485	3,32	1,278	-2,630**	,009	0,14	Sí	889	3,50	1,231	Capacidad de Atención	No	487	3,64	1,257	-3,133**	,002	0,18
Creatividad	No	485	3,40	1,220	-4,514**	,000	0,25																																																								
	Sí	900	3,70	1,172				Resolución de Problemas	No	492	3,30	1,253	-4,204**	,000	0,24	Sí	889	3,60	1,215	Trabajo en Equipo	No	490	2,85	1,481	-2,724**	,007	0,15	Sí	900	3,08	1,517	Persistencia-Perseverancia	No	485	3,32	1,278	-2,630**	,009	0,14	Sí	889	3,50	1,231	Capacidad de Atención	No	487	3,64	1,257	-3,133**	,002	0,18	Sí	906	3,86	1,188								
Resolución de Problemas	No	492	3,30	1,253	-4,204**	,000	0,24																																																								
	Sí	889	3,60	1,215				Trabajo en Equipo	No	490	2,85	1,481	-2,724**	,007	0,15	Sí	900	3,08	1,517	Persistencia-Perseverancia	No	485	3,32	1,278	-2,630**	,009	0,14	Sí	889	3,50	1,231	Capacidad de Atención	No	487	3,64	1,257	-3,133**	,002	0,18	Sí	906	3,86	1,188																				
Trabajo en Equipo	No	490	2,85	1,481	-2,724**	,007	0,15																																																								
	Sí	900	3,08	1,517				Persistencia-Perseverancia	No	485	3,32	1,278	-2,630**	,009	0,14	Sí	889	3,50	1,231	Capacidad de Atención	No	487	3,64	1,257	-3,133**	,002	0,18	Sí	906	3,86	1,188																																
Persistencia-Perseverancia	No	485	3,32	1,278	-2,630**	,009	0,14																																																								
	Sí	889	3,50	1,231				Capacidad de Atención	No	487	3,64	1,257	-3,133**	,002	0,18	Sí	906	3,86	1,188																																												
Capacidad de Atención	No	487	3,64	1,257	-3,133**	,002	0,18																																																								
	Sí	906	3,86	1,188																																																											

** Significativo al nivel $p_{(t)} < 0,01$

Con respecto a la **Pregunta 7** del cuestionario [“P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?”], que aborda el objetivo de investigación **O_{2Eg}**: es una escala tipo Likert de 5 puntos (1=*Suspense*; 2=*Aprobado*; 3=*Notable bajo*; 4=*Notable alto*; 5=*Sobresaliente*). En la Tabla 5.57 se aportan los estadísticos descriptivos principales.

Tabla 5.57. Estadísticos descriptivos de la Pregunta 7 del cuestionario para estudiantes

		Matemáticas	Lengua	Música	Tecnología / Informática	Plástica
N	Válido	1458	1449	1365	1370	1392
	Perdidos	204	213	297	292	270
Media		3,34	3,22	3,73	3,64	3,82
Mediana		4,00	3,00	4,00	4,00	4,00
Moda		5	4	5	5	5
Desviación estándar		1,376	1,273	1,257	1,280	1,226
Varianza		1,894	1,621	1,580	1,639	1,503
Asimetría		-,282	-,146	-,634	-,587	-,772
Mínimo		1	1	1	1	1
Máximo		5	5	5	5	5

Tal y como puede observarse en la anterior tabla (y Figura D.25 del anexo), el par de asignaturas ‘Matemáticas-Lengua’ es el que presenta un rendimiento académico informado por los estudiantes más bajo; en el otro extremo del gráfico, el par de asignatura ‘Música-Plástica’ presenta el rendimiento académico más alto; y entre ambos la asignatura de ‘Tecnología / Informática’.

Se encuentran diferencias estadísticamente muy significativas en función del sexo, a favor de las chicas, en el rendimiento informado en Lengua, Música y Plástica ($p_{(t)} = 0,000 \ll 0,05$). No encontramos diferencias en función del sexo, ni en el rendimiento informado en Matemáticas ($p_{(t)} = 0,844 > 0,05$), ni en el de Tecnología / Informática ($p_{(t)} = 0,221 > 0,05$). Se detalla en la siguiente Tabla 5.58 (y Figura D.26 del anexo).

Tabla 5.58. Diferencias en la Pregunta 7, en función del sexo

<i>P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?</i>	Sexo	N	Media	Desviación estándar	t	P _(t)	d de Cohen
Matemáticas	Chico	740	3,35	1,377	0,196	,844	0,01
	Chica	680	3,34	1,375			
Lengua	Chico	735	3,08	1,278	-4,313**	,000	0,23
	Chica	677	3,37	1,250			
Música	Chico	693	3,61	1,306	-3,682**	,000	0,20
	Chica	637	3,86	1,187			
Tecnología / Informática	Chico	708	3,69	1,274	1,224	,221	0,07
	Chica	629	3,60	1,275			
Plástica	Chico	717	3,63	1,283	-6,421**	,000	0,35
	Chica	639	4,05	1,098			

** Significativo al nivel $p_{(t)} < 0,01$

Con respecto a la etapa educativa, encontramos diferencias estadísticamente muy significativas para todas las asignaturas interrogadas (Tabla 5.59). Tal y como puede observarse en la Figura 5.30,

existe un apreciable descenso en el rendimiento académico informado de la etapa de Educación Primaria a la de Educación Secundaria; que se recupera en Bachillerato, debido especialmente a las féminas.

Tabla 5.59. Diferencias en la Pregunta 7, en función de la etapa educativa

<i>P7. ¿Cuál es tu nota habitual en las siguientes asignaturas?</i>	Etapa Educativa	N	Media	Desviación estándar	F	p(F)
Matemáticas	Primaria	403	3,77	1,239	21,128**	,000
	Secundaria	888	3,13	1,385		
	Bachillerato	140	3,42	1,389		
	Formación Profesional	15	3,60	1,502		
Lengua	Primaria	398	3,60	1,157	16,788**	,000
	Secundaria	880	3,07	1,297		
	Bachillerato	144	3,13	1,184		
	Formación Profesional	15	3,47	1,506		
Música	Primaria	389	3,81	1,183	4,748**	,003
	Secundaria	821	3,66	1,291		
	Bachillerato	128	4,07	1,145		
	Formación Profesional	15	3,53	1,356		
Tecnología / Informática	Primaria	344	3,98	1,160	14,663**	,000
	Secundaria	868	3,48	1,317		
	Bachillerato	132	3,81	1,078		
	Formación Profesional	15	4,07	1,387		
Plástica	Primaria	389	4,03	1,108	5,505**	,001
	Secundaria	844	3,73	1,269		
	Bachillerato	132	3,78	1,155		
	Formación Profesional	15	3,67	1,397		

** Significativo al nivel $p_{(F)} < 0,01$

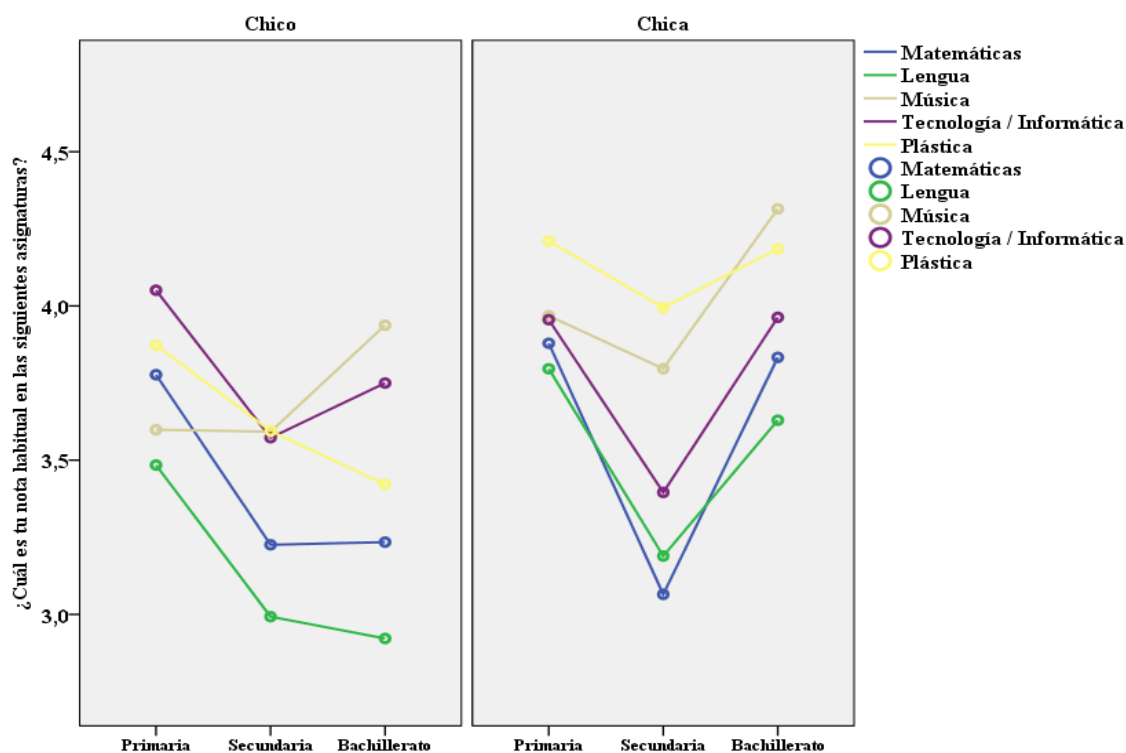


Figura 5.30. Medias en la Pregunta 7, en función de sexo y etapa educativa (se omite FP por su escasa muestra)

Tomada la Pregunta 7 como una escala de 5 elementos, presenta una adecuada fiabilidad como consistencia interna ($\alpha = 0,834$). Aparecen correlaciones estadísticamente muy significativas, de signo positivo e intensidad moderada-alta, entre los rendimientos informados de todas las asignaturas (Tabla 5.60); destacando las correlaciones entre Matemáticas y Lengua ($r = +0,642$), y Matemáticas con Tecnología/Informática ($r = +0,596$).

Tabla 5.60. Correlaciones entre los elementos de la Pregunta 7

	Matemáticas	Lengua	Música	Tecnología / Informática	Plástica
Matemáticas	1	,642**	,453**	,596**	,416**
Lengua		1	,491**	,536**	,463**
Música			1	,488**	,514**
Tecnología / Informática				1	,456**
Plástica					1

** Correlación muy significativa $p(r) < 0,01$

Centrándonos, ahora sí, en el objetivo específico de investigación O_{2Eg} , en la siguiente Tabla 5.61 se muestran las correlaciones entre los distintos elementos de la Pregunta 7, con respecto a las preguntas 2 (*Experiencia previa 'coding'*), 5 (*Autoeficacia 'coding'*), y 6 (*Completar actividad HoC*)

Tabla 5.61. Correlaciones entre los elementos de la Pregunta 7, con las preguntas 2, 5 y 6

	Matemáticas	Lengua	Música	Tecnología / Informática	Plástica
P2. Experiencia previa 'coding'	,131**	,090**	,109**	,189**	,085**
P5. Autoeficacia 'coding'	,182**	,116**	,112**	,240**	,112**
P6. Completar actividad HoC	,180**	,109**	,088**	,207**	,092**

** Correlación muy significativa $p(r) < 0,01$

Tal y como puede observarse en la tabla anterior, aparecen correlaciones estadísticamente muy significativas, de signo positivo e intensidad moderadamente baja. Destaca que, tanto para la Pregunta 2 como para las preguntas 5 y 6, las correlaciones de mayor intensidad aparecen con el rendimiento informado en Matemáticas y Tecnología/Informática (en negrita y fondo verde).

Como análisis complementario, y más en detalle, en la siguiente Tabla 5.62 (y Figura D.27 del anexo) se muestran las diferencias en los distintos elementos de la Pregunta 7, en función de tener o no experiencia previa en 'coding' (Pregunta 2). Todas las diferencias son estadísticamente muy significativas ($p(t) \ll 0,05$), apareciendo los mayores tamaños del efecto en el caso de Tecnología/Informática ($d = 0,39$) y Matemáticas ($d = 0,27$).

Tabla 5.62. Diferencias en la Pregunta 7, en función de la experiencia previa en 'coding' (Pregunta 2)

P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?	P2. Experiencia previa 'coding'	N	Media	Desviación estándar	t	P(t)	d de Cohen
Matemáticas	No	835	3,20	1,389	-5,059**	,000	0,27
	Sí	616	3,56	1,324			
Lengua	No	830	3,13	1,259	-3,417**	,001	0,18
	Sí	612	3,36	1,273			
Música	No	780	3,62	1,255	-4,052**	,000	0,23
	Sí	578	3,90	1,233			

<i>P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?</i>	<i>P2. Experiencia previa 'coding'</i>	N	Media	Desviación estándar	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i> de Cohen
Tecnología / Informática	No	784	3,43	1,302	-7,212**	,000	0,39
	Sí	582	3,92	1,193			
Plástica	No	801	3,74	1,242	-3,198**	,001	0,17
	Sí	584	3,95	1,184			

** Significativo al nivel $p(t) < 0,01$

Ahora bien, si replicamos el análisis anterior diferenciado por sexo, comprobamos que el efecto de tener experiencia previa en 'coding' sobre el rendimiento es especialmente destacable entre el colectivo masculino (Tabla 5.63 y Figura 5.31).

Tabla 5.63. Correlaciones entre preguntas 2 y 7, diferenciado por sexos

		<i>P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?</i>				
		Matemáticas	Lengua	Música	Tecnología/ Informática	Plástica
<i>P2. Experiencia previa 'coding'</i>	Chico	,216**	,138**	,133**	,259**	,143**
	Chica	,034	,065	,110**	,108**	,049

** Correlación muy significativa $p(r) < 0,01$

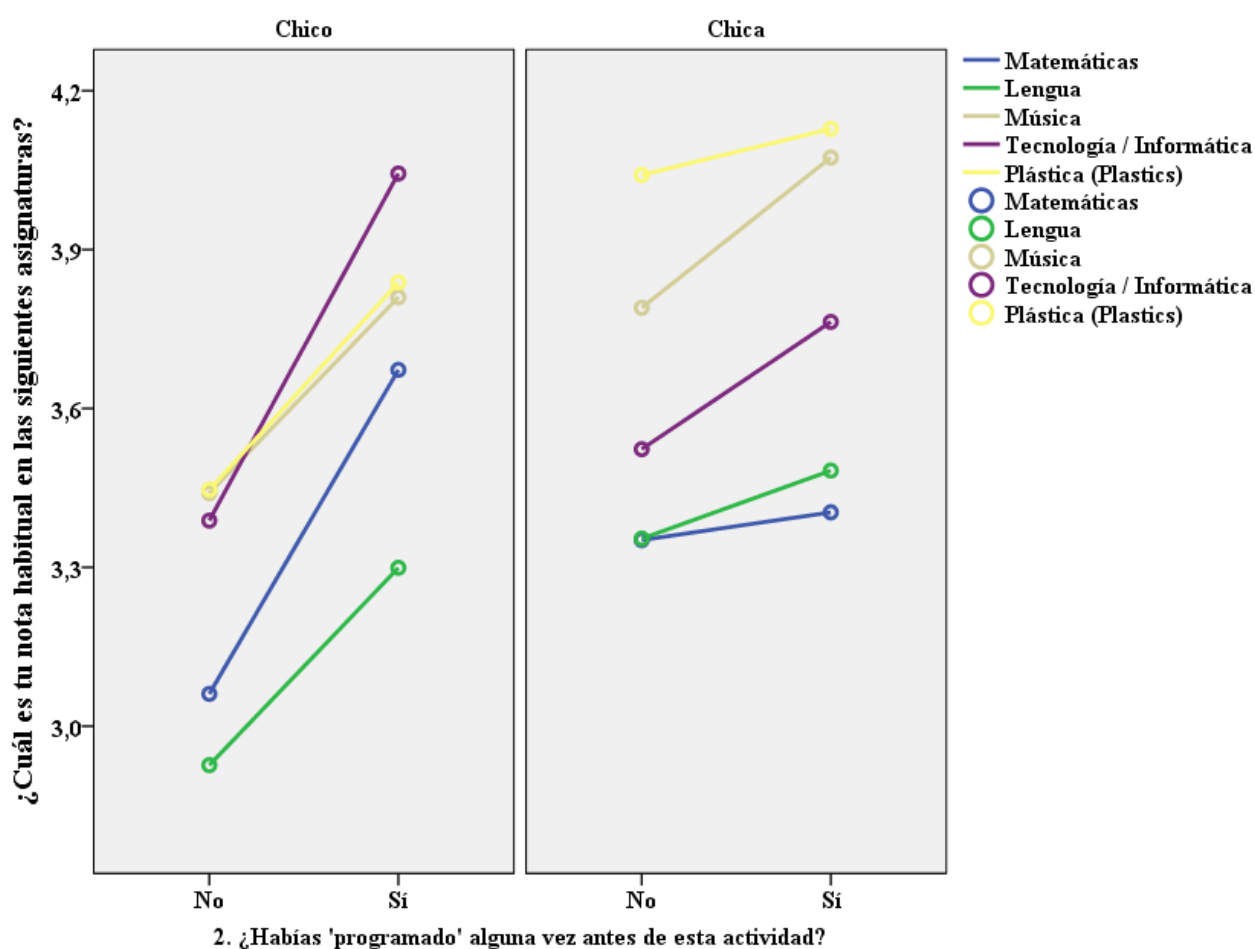


Figura 5.31. Medias en la Pregunta 7, en función de la experiencia previa en 'coding' (Pregunta 2) y sexo

Análogamente, en la siguiente Figura 5.32 (y Tabla D.16 del anexo) se muestran las diferencias en los distintos elementos de la Pregunta 7, en función de la percepción de autoeficacia en ‘coding’ tras la actividad (Pregunta 5). Todas las diferencias son estadísticamente muy significativas ($p_{(F)} \ll 0,05$), apareciendo las mayores diferencias en el caso de Tecnología/Informática ($F = 21,916$) y Matemáticas ($F = 12,933$).

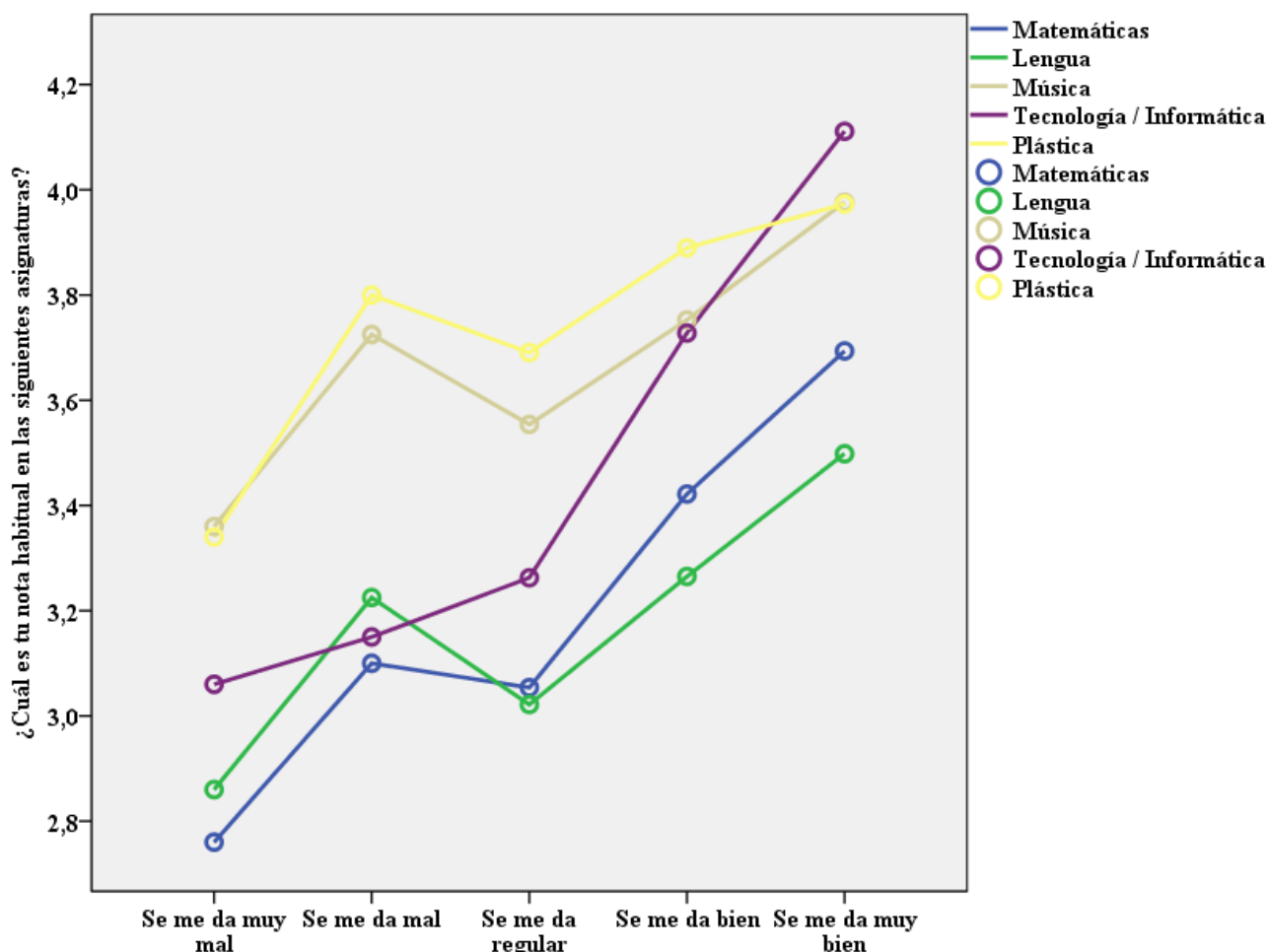


Figura 5.32. Medias en la Pregunta 7, en función de la autoeficacia en ‘coding’ (Pregunta 5)

Otra vez, si replicamos el análisis anterior diferenciado por sexo, comprobamos que el efecto de la autoeficacia en ‘coding’ sobre el rendimiento informado es especialmente destacable en el colectivo masculino (Tabla 5.64 y Figura 5.33)

Tabla 5.64. Correlación entre las preguntas 5 y 7, diferenciado por sexo

		P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?				
		Matemáticas	Lengua	Música	Tecnología / Informática	Plástica
P5. Autoeficacia ‘coding’	Chico	,226**	,190**	,143**	,291**	,158**
	Chica	,118**	,054	,068	,152**	,071

** Correlación muy significativa $p_{(r)} < 0,01$

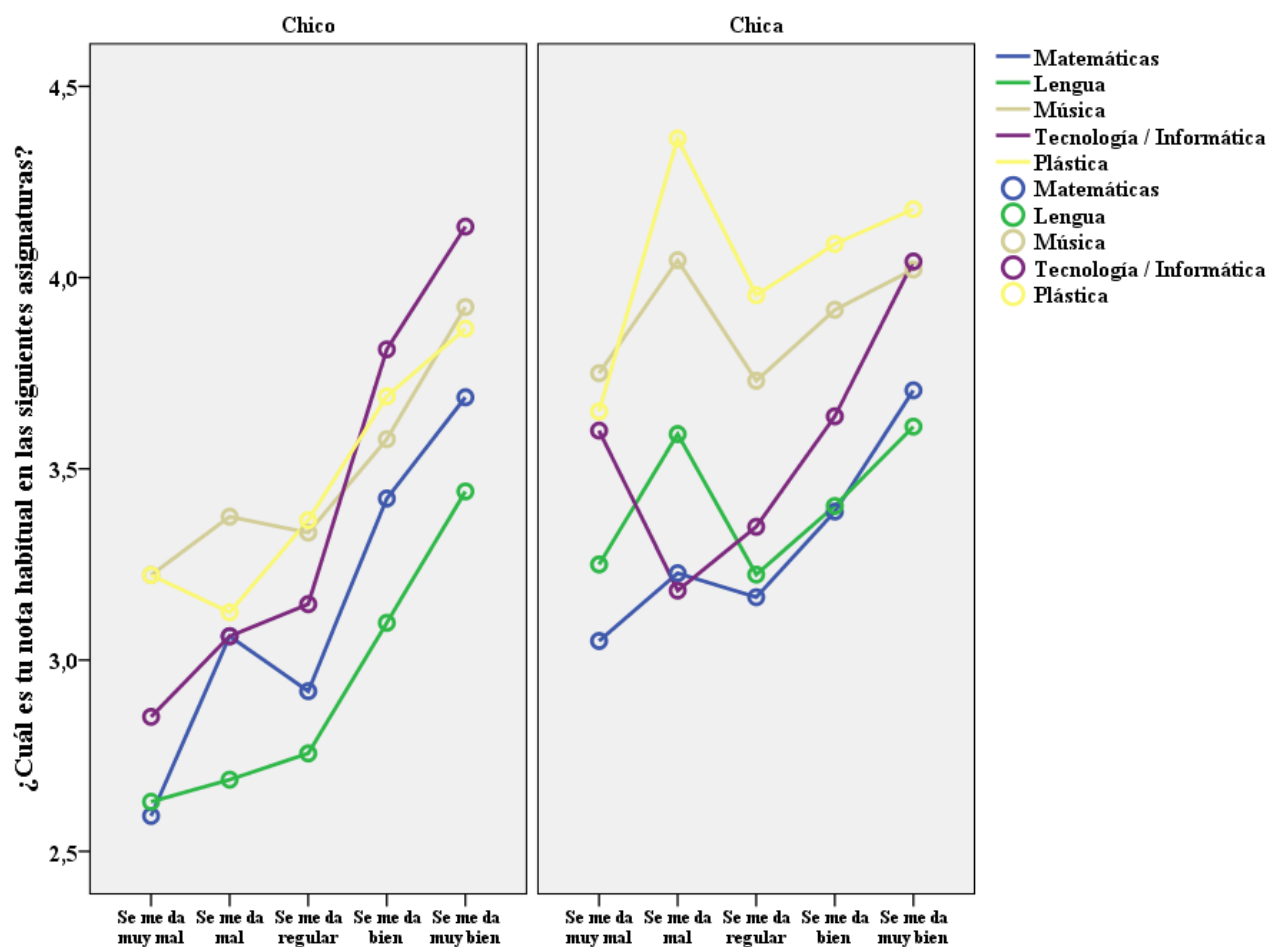


Figura 5.33. Medias en la Pregunta 7, en función de la autoeficacia en ‘coding’ (Pregunta 5) y sexo

Finalmente, en la siguiente Tabla 5.65 (y Figura D.28 del anexo) se muestran las diferencias en los distintos elementos de la Pregunta 7, en función de haber conseguido acabar, o no, la actividad HoC (Pregunta 6). Todas las diferencias son estadísticamente muy significativas ($p_{(t)} \ll 0,05$), a favor del grupo que sí completó la actividad HoC, apareciendo los mayores tamaños del efecto en Tecnología/Informática ($d = 0,44$) y Matemáticas ($d = 0,38$).

Tabla 5.65. Diferencias en la Pregunta 7, en función de haber completado la actividad HoC (Pregunta 6)

P7. ¿Cuál es tu nota habitual en las siguientes asignaturas?	P6. Completar actividad HoC	N	Media	Desviación estándar	t	P _(t)	d de Cohen
Matemáticas	No	501	3,01	1,374	-6,889**	,000	0,38
	Sí	917	3,53	1,333			
Lengua	No	498	3,04	1,301	-4,098**	,000	0,23
	Sí	913	3,33	1,237			
Música	No	463	3,59	1,311	-3,136**	,002	0,18
	Sí	868	3,82	1,211			
Tecnología / Informática	No	454	3,28	1,360	-7,382**	,000	0,44
	Sí	881	3,84	1,185			
Plástica	No	474	3,67	1,303	-3,288**	,001	0,19
	Sí	883	3,91	1,164			

** Significativo al nivel $p_{(t)} < 0,01$

En este caso, si replicamos el análisis diferenciado por sexo, comprobamos que el efecto de haber conseguido acabar la actividad HoC sobre el rendimiento informado es similar entre chicos y chicas (Tabla 5.66 y Figura 5.34).

Tabla 5.66. Correlación entre las preguntas 6 y 7, diferenciado por sexo

		<i>P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?</i>				
		Matemáticas	Lengua	Música	Tecnología / Informática	Plástica
<i>P6. Completar actividad HoC</i>	Chico	,170**	,117**	,102**	,218**	,105**
	Chica	,187**	,123**	,105**	,192**	,125**

** Correlación muy significativa $p_{(r)} < 0,01$

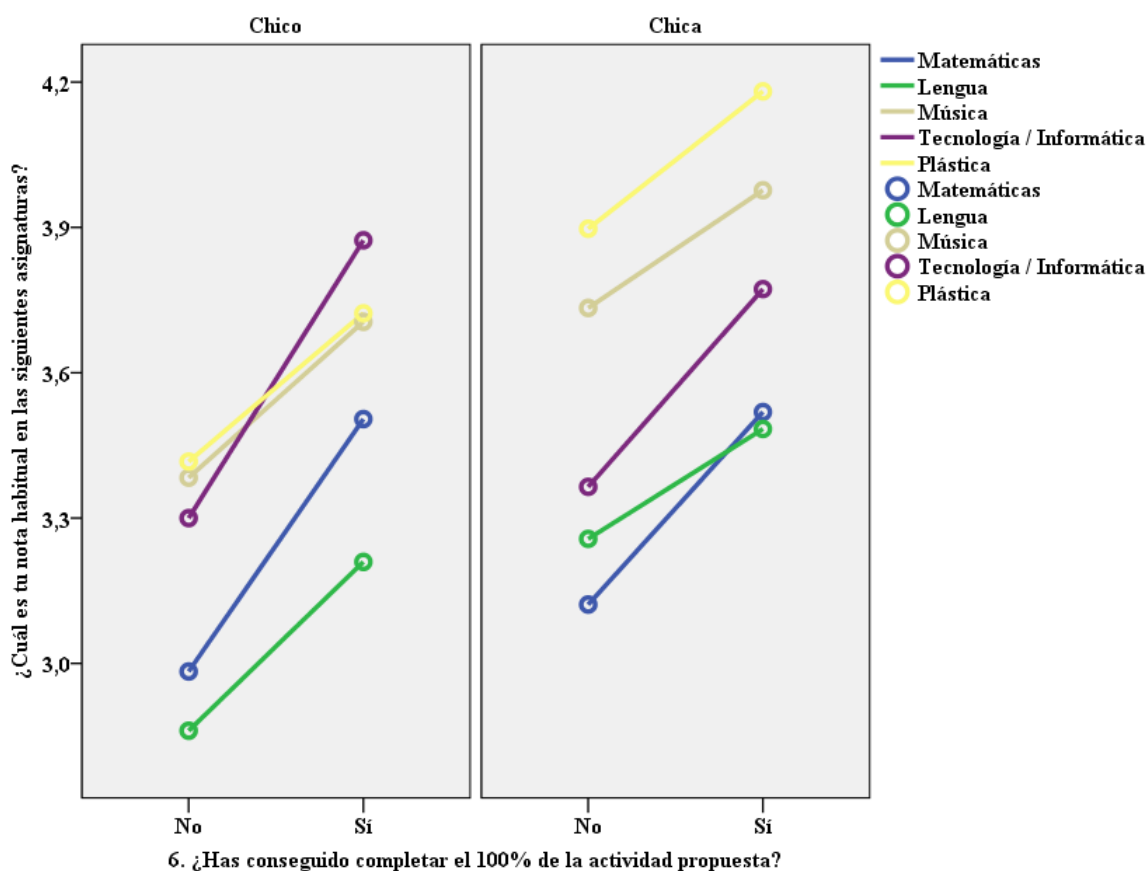


Figura 5.34. Medias en la Pregunta 7, en función de haber conseguido completar la actividad HoC (Pregunta 6) y sexo

Con respecto a la **Pregunta 8** del cuestionario [“P8. Tras la actividad, ¿cómo de motivado te sientes para continuar aprendiendo 'programación informática' ('coding')?”], que aborda el objetivo de investigación **O_{2Eh}**: es una escala tipo Likert de 5 puntos (desde 1=Nada motivado a 5=Muy motivado). En la Tabla 5.67 se aportan los estadísticos descriptivos principales. Tal y como puede verse, los sujetos se sitúan principalmente en la parte medio-alta de la escala (Media = 3,70; Mediana = 4,00; Moda = 4,00); constituyendo una distribución de puntuaciones con clara asimetría negativa. En la Tabla 5.68 y Figura 5.35 se detalla la distribución de frecuencias de respuesta a la Pregunta 8: tan sólo un 12,6% se declara ‘nada motivado’ o ‘poco motivado’ para continuar aprendiendo ‘coding’; el 27,2% se declara ‘algo motivado’; y un 60,2% dice estar ‘bastante motivado’ o ‘muy motivado’ para seguir con el aprendizaje del ‘coding’ tras su experiencia en el evento de ‘La Hora del Código’.

Tabla 5.67. Descriptivos relativos a la Pregunta 8 del cuestionario para estudiantes

<i>P8. Tras la actividad, ¿cómo de motivado te sientes para continuar aprendiendo 'programación informática' ('coding')?</i>		
N	Válido	1430
	Perdidos	232
Media		3,70
Mediana		4,00
Moda		4
Desviación estándar		1,097
Varianza		1,204
Asimetría		-,631
Curtosis		-,157
Mínimo		1
Máximo		5

Tabla 5.68. Distribución de frecuencias de las respuestas a la Pregunta 8 del cuestionario para estudiantes

<i>P8. Tras la actividad, ¿cómo de motivado te sientes para continuar aprendiendo 'programación informática' ('coding')?</i>					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada motivado	71	4,3	5,0	5,0
	Poco motivado	109	6,6	7,6	12,6
	Algo motivado	389	23,4	27,2	39,8
	Bastante motivado	474	28,5	33,1	72,9
	Muy motivado	387	23,3	27,1	100,0
	Total	1430	86,0	100,0	
Perdidos	Sistema	232	14,0		
Total		1662	100,0		

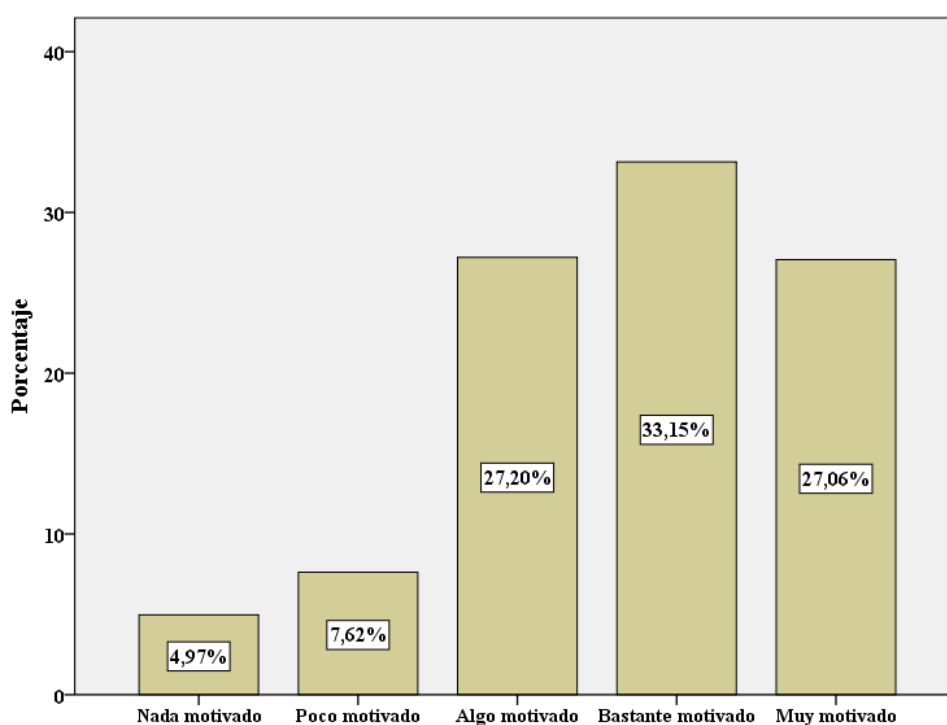


Figura 5.35. Distribución de frecuencias de las respuestas a la Pregunta 8 del cuestionario para estudiantes

Con respecto a la **Pregunta 9** del cuestionario [“P9. ¿Te gustaría continuar aprendiendo ‘programación informática’ (‘coding’) a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)?”], que aborda el objetivo de investigación **O_{2Ei}**: en la siguiente Tabla 5.69 (y Figura D.29 del anexo) se muestra la distribución de frecuencias de las respuestas obtenidas. Tal y como puede observarse, 8 de cada 10 estudiantes declara que le gustaría seguir aprendiendo ‘coding’ a través de entornos ‘on-line’ autoguiados.

Tabla 5.69. Distribución de frecuencias de las respuestas a la Pregunta 9 del cuestionario para estudiantes

P9. ¿Te gustaría continuar aprendiendo 'programación informática' ('coding') a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)?					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	No	282	17,0	19,6	19,6
	Sí	1154	69,4	80,4	100,0
	Total	1436	86,4	100,0	
Perdidos	Sistema	226	13,6		
Total		1662	100,0		

Pasamos ahora a hacer un análisis conjunto de la **Pregunta 8**³³⁸ y la **Pregunta 9**³³⁹, que se tomará como una variable ‘dummy’ ($0=no$; $1=sí$), dado que ambas están secuenciadas en el cuestionario de manera lógica. Tal y como era de esperar, aparece una correlación estadísticamente muy significativa, de signo positivo e intensidad moderada-alta, entre las respuestas a ambas preguntas ($r = +0,516$; $p_{(r)} < 0,01$). Se ilustra con un diagrama de dispersión en la Figura D.30 del anexo.

Encontramos diferencias estadísticamente muy significativas en función del sexo para ambas preguntas ($p_{(t)} \ll 0,05$), a favor del colectivo masculino (Tabla 5.70); si bien con unos tamaños del efecto pequeños ($d = 0,19$ para la Pregunta 8; y $d = 0,15$ para la Pregunta 9).

Tabla 5.70. Diferencias para las preguntas 8 y 9, según el sexo

	Sexo	N	Media	Desviación estándar	t	P_(t)	d de Cohen
P8. Motivación proseguir aprendizaje ‘coding’	Chico	728	3,80	1,106	3,602**	,000	0,19
	Chica	670	3,59	1,073			
P9. Determinación aprender ‘coding’ on-line	Chico	728	,83	,373	2,823**	,005	0,15
	Chica	672	,77	,419			

Ahora bien, tal y como podemos comprobar en la siguiente Tabla 5.71, esas diferencias por sexo son sólo estadísticamente muy significativas en la etapa de Educación Secundaria.

Tabla 5.71. Diferencias para las preguntas 8 y 9, según el sexo; diferenciado por etapas educativas

Etapas Educativas	Sexo	N	Media	Desviación estándar	t	p_(t)	d de Cohen
Primaria	P8. Motivación proseguir aprendizaje ‘coding’	Chico	181	4,25	,960	-0,070	,944
		Chica	192	4,26	,852		
	P9. Determinación aprender ‘coding’ on-line	Chico	179	,88	,329	-1,460	,145
		Chica	194	,92	,268		

³³⁸ En las próximas tablas y figuras se abreviará como “P8. Motivación proseguir aprendizaje ‘coding’”

³³⁹ En las próximas tablas y figuras se abreviará como “P9. Determinación aprender ‘coding’ on-line”

Etapa Educativa		Sexo	N	Media	Desviación estándar	t	p _(t)	d de Cohen
Secundaria	P8. Motivación proseguir aprendizaje 'coding'	Chico	455	3,64	1,089	4,667**	,000	0,31
		Chica	408	3,31	1,022			
	P9. Determinación aprender 'coding' on-line	Chico	458	,83	,376	4,425**	,000	0,31
		Chica	407	,70	,458			
Bachillerato	P8. Motivación proseguir aprendizaje 'coding'	Chico	75	3,71	1,148	2,041*	,043	0,35
		Chica	65	3,32	1,062			
	P9. Determinación aprender 'coding' on-line	Chico	75	,77	,421	0,009	,993	0,00
		Chica	66	,77	,422			
Formación Profesional	P8. Motivación proseguir aprendizaje 'coding'	Chico	12	3,42	1,730	-2,082	,062	0,97
		Chica	3	4,67	,577			
	P9. Determinación aprender 'coding' on-line	Chico	11	,73	,467	-1,936	,082	0,82
		Chica	3	1,00	,000			

* Significativo al nivel p_(t) < 0,05

** Significativo al nivel p_(t) < 0,01

En esta misma línea, encontramos diferencias estadísticamente muy significativas ($p_{(F)} = 0,000 \ll 0,05$) en función de la etapa educativa, tanto para la Pregunta 8 como para la Pregunta 9 (Tabla D.17 del anexo). Tal y como puede verse en las Figuras 5.36 y 5.37, hay un brusco descenso en las medias de ambas preguntas al pasar de Educación Primaria a Educación Secundaria (atribuible especialmente al colectivo femenino, según acabamos de ver), que luego se recupera ligeramente en la etapa de Bachillerato.

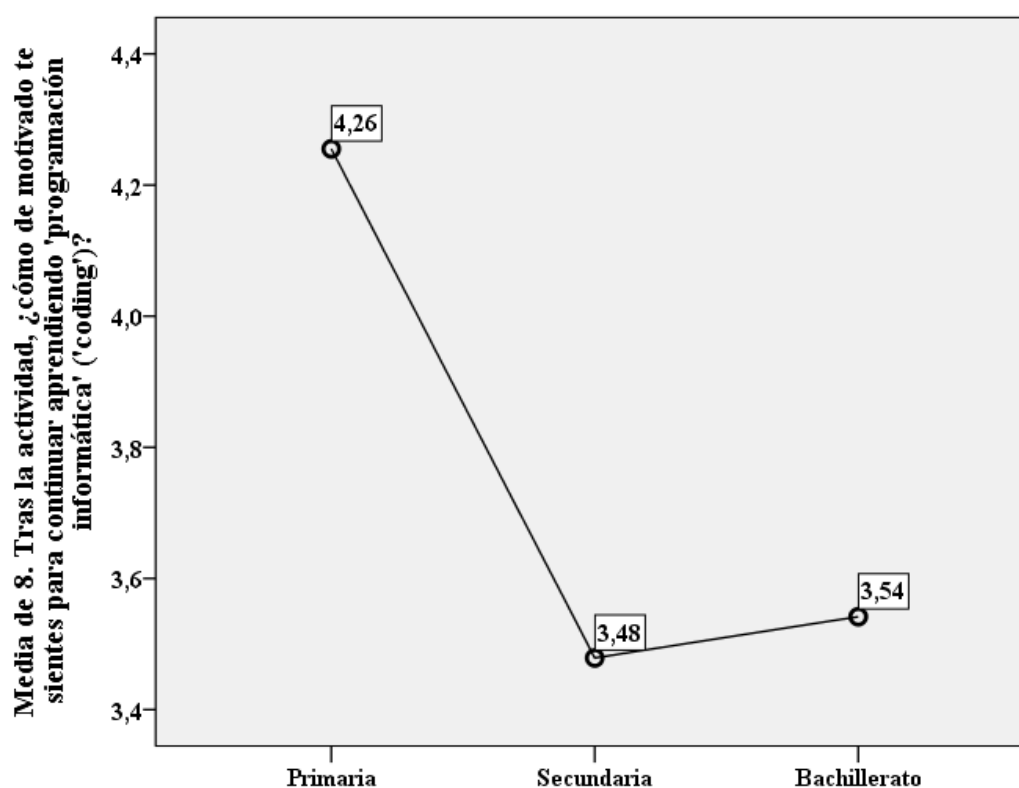


Figura 5.36. Medias en la Pregunta 8 según la etapa educativa (se omite FP por su escasa muestra)

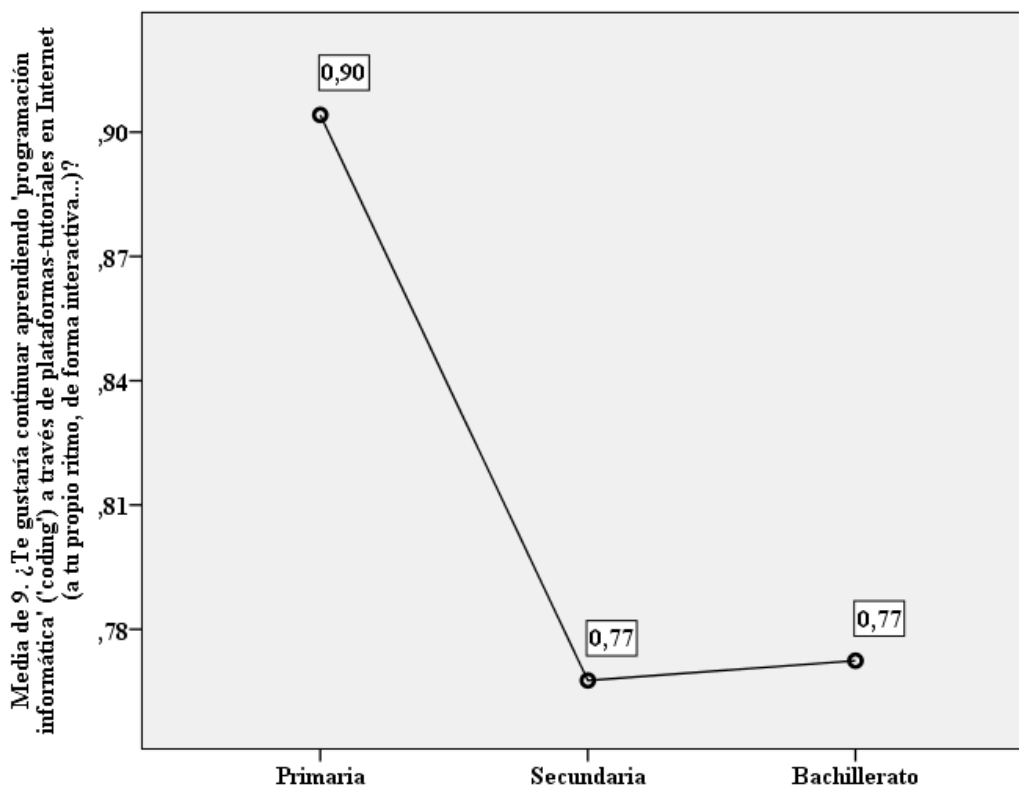


Figura 5.37. Medias en la Pregunta 9 según la etapa educativa (se omite FP por su escasa muestra)

Pasamos ahora a explorar la relación entre las preguntas 8 y 9, con la Pregunta 5 de autoeficacia. Tal y como puede observarse en la siguiente Tabla 5.72, aparecen correlaciones estadísticamente muy significativas, de signo positivo e intensidad moderada.

Tabla 5.72. Correlaciones de la Pregunta 5 con las preguntas 8 y 9

	P8. Motivación proseguir aprendizaje 'coding'	P9. Determinación aprender 'coding' on-line
P5. Autoeficacia 'coding'	,497**	,331**

** Correlación muy significativa $p(r) < 0,01$

Tal y como era de esperar, aparecen diferencias estadísticamente muy significativas ($p(F) = 0,000 \ll 0,05$), tanto en la Pregunta 8 como en la Pregunta 9, en función de la autoeficacia (Tabla D.18 del anexo). Según se observa en la próxima Figura 5.38, hay una relación casi lineal entre los incrementos de percepción de autoeficacia con los de motivación a proseguir el aprendizaje del 'coding': en tanto en cuanto el sujeto acaba la actividad HoC con una buena sensación de competencia en tareas de 'coding', en dicha medida se siente motivado a proseguir con el aprendizaje del mismo.

La relación es ligeramente distinta al representar como varía la media en la pregunta 9, de determinación de aprender 'coding on-line', en función de la percepción de autoeficacia (Figura 5.39). En este caso parece encontrarse un techo, según la tendencia asintótica horizontal de la gráfica, en el cual, por más que aumente la percepción de autoeficacia, ya prácticamente no se incrementa el porcentaje de sujetos dispuestos aprender 'coding' a través de plataformas-tutoriales 'on-line' autoguiados. Así pues, hay un grupo reducido (pero consistente) de estudiantes que, a pesar de su buena autoeficacia en 'coding', renunciarían a seguir aprendiéndolo según un sistema 'on-line'

(presumiblemente sí estarían dispuestos a proseguir el aprendizaje del 'coding', pero a través de otros sistemas).

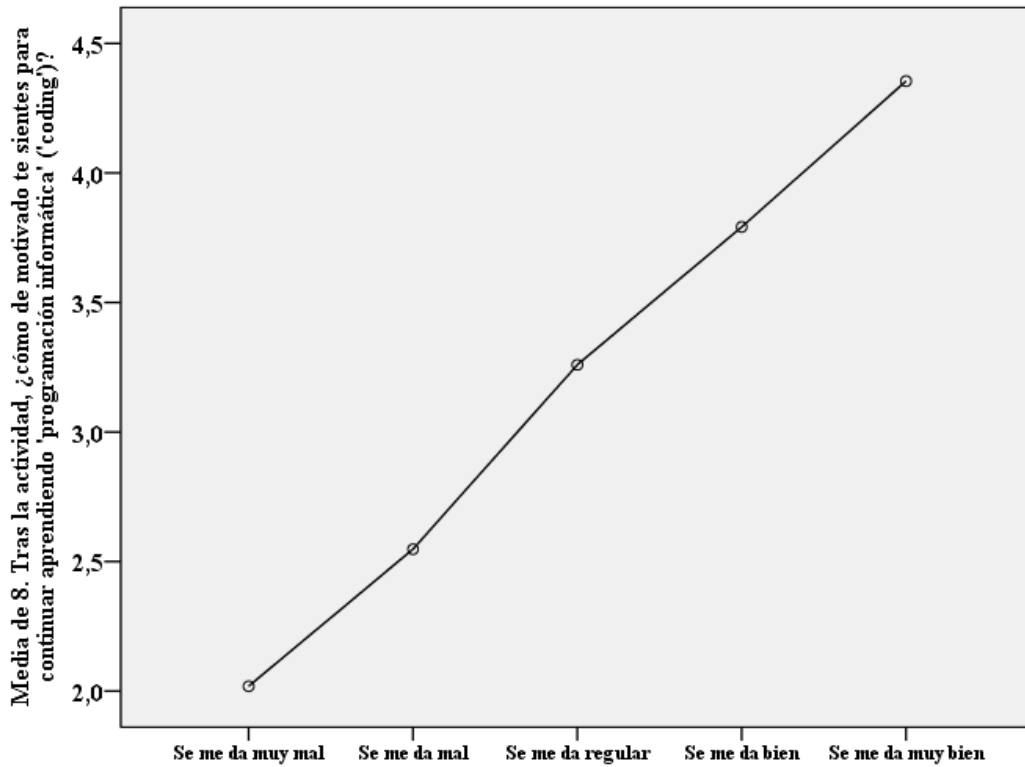


Figura 5.38. Media en la Pregunta 8, en función de la percepción de autoeficacia (Pregunta 5)

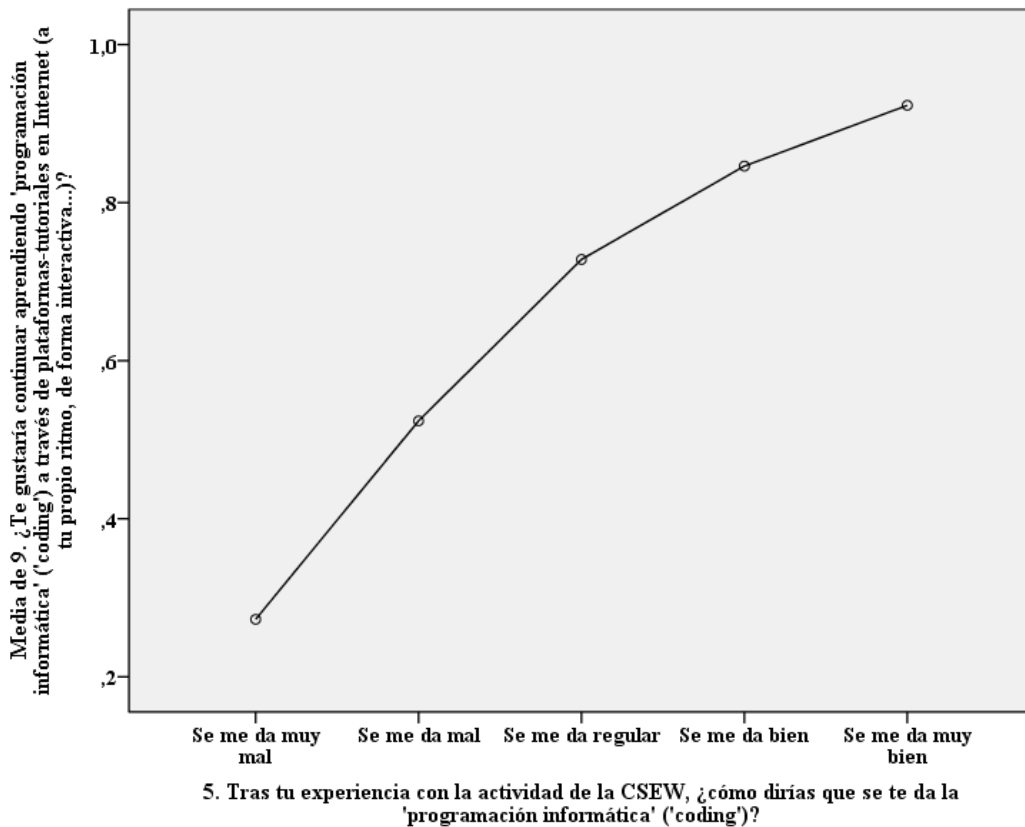


Figura 5.39. Media en la Pregunta 9, en función de la percepción de autoeficacia (Pregunta 5)

Diferencias en este mismo sentido se producen en función de la Pregunta 6. Aparecen diferencias estadísticamente muy significativas ($p_{(t)} = 0,000 \ll 0,05$), tanto en la Pregunta 8 como en la Pregunta 9, a favor del grupo que afirma haber conseguido acabar la actividad HoC propuesta en su centro (Tabla 5.73)

Tabla 5.73. Diferencias en las preguntas 8 y 9, en función de la Pregunta 6

	<i>P6. Completar actividad HoC</i>	N	Media	Desviación estándar	<i>t</i>	$p_{(t)}$	<i>d de Cohen</i>
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	No	491	3,44	1,128	-6,313**	,000	0,36
	Sí	897	3,83	1,054			
<i>P9. Determinación aprender 'coding' on-line</i>	No	488	,74	,439	-4,229**	,000	0,25
	Sí	904	,84	,368			

** Significativo al nivel $p_{(t)} < 0,01$

Venimos viendo, por tanto, una sucesión de relaciones concatenadas a lo largo de las preguntas del cuestionario, que queda bien representada en la siguiente matriz de correlaciones (Tabla 5.74). Tal y como puede observarse, se detectan dos cuadrantes de intensidad moderada en la matriz (en negrita): uno formado por las preguntas 1 y 2 (fondo rosa), relativo a la exposición al 'coding'; y otro formado por las preguntas 5, 6, 8 y 9 (fondo azul), relativo a la autoeficacia-motivación en 'coding'.

Tabla 5.74. Matriz de correlaciones de las preguntas 1, 2, 5, 6, 8 y 9 del cuestionario para estudiantes

	<i>P1</i>	<i>P2</i>	<i>P5</i>	<i>P6</i>	<i>P8</i>	<i>P9</i>
<i>P1. Haber oído 'coding'</i>	1	,379**	,146**	,053*	,107**	,105**
<i>P2. Experiencia previa 'coding'</i>		1	,129**	,125**	,042	,060*
<i>P5. Autoeficacia 'coding'</i>			1	,226**	,497**	,331**
<i>P6. Completar actividad HoC</i>				1	,170**	,119**
<i>P8. Motivación proseguir aprendizaje 'coding'</i>					1	,516**
<i>P9. Determinación aprender 'coding' on-line</i>						1

* Correlación significativa $p_{(r)} < 0,05$

** Correlación muy significativa $p_{(r)} < 0,01$

Así pues, en este punto, detectamos la necesidad de aplicar algunas técnicas de análisis multivariante para enriquecer nuestra descripción; que den cuenta del comportamiento conjunto y complejo de todas o parte de las variables medidas a través del cuestionario. En concreto, van a realizarse los siguientes análisis: a) un análisis de clúster o conglomerados, a partir de las respuestas a dichas preguntas 1, 2, 5, 6, 8 y 9; b) un análisis de regresión múltiple sobre la Pregunta 8 (tomada como variable continua de intervalo); y c) un análisis discriminante sobre la Pregunta 9 (tomada como variable categórica dicotómica)

a) Análisis de clúster

En el análisis de clúster incluimos todas las variables mostradas en la anterior Tabla 5.74, es decir, las respuestas a las preguntas 1, 2, 5, 6, 8 y 9. Según se intuye en la anterior matriz de correlaciones, se hipotetizan cuatro clústeres distintos que surgirían de la combinación de los dos siguientes ejes-bloques:

- Eje I: formado por las preguntas 1 y 2; que clasificaría a los sujetos en función de tener, o no tener, exposición previa al ‘coding’
- Eje II: formado por las preguntas 5, 6, 8 y 9; que clasificaría a los sujetos en función de tener alta o baja autoeficacia-motivación en el aprendizaje del ‘coding’

Llevamos a cabo, pues, el análisis de clúster según los siguientes parámetros de cálculo estadístico (Gil Pascual, 2003): se estandarizan todas las variables dado que no todas son del mismo rango; al ser un número masivo de casos, se aplica un análisis de clúster de *k-medias* sobre dichas variables; se indica formar 4 clústeres; con el método “*iterar y clasificar*” (10 iteraciones máximo, criterio de convergencia 0); guardamos clúster de pertenencia y distancia al centro del clúster.

Para ser consultado en el Anexo D, en la Tabla D.19 se muestran los estadísticos descriptivos principales de las preguntas incluidas en el análisis de clúster. En la Tabla D.20 se muestran los centros de clústeres (expresados como puntuaciones típicas) encontrados inicialmente, y en la Tabla D.21 se detalla el historial de iteraciones seguidas, que convergió satisfactoriamente en las 10 iteraciones fijadas como máximo para el análisis.

Llegamos entonces a la matriz con los centros finales de los cuatro clústeres (Tabla 5.75), expresados en puntuaciones típicas. Para su mejor interpretación, las celdas de la tabla se representan según el siguiente código de colores: las puntuaciones entre -0,15 y +0,15 (en el entorno de la media de la variable) se representan en fondo blanco; las puntuaciones < -0,15 (inferiores a la media de la variable) se representan en fondo rosa; y las puntuaciones > +0,15 (superiores a la media de la variable) se representan en fondo verde.

Tabla 5.75. Centros de clústeres finales (expresados como puntuaciones típicas o *Zscore*)

	Clúster			
	1	2	3	4
<i>Zscore: P1. Haber oído ‘coding’</i>	-,66188	,21724	,45735	-,28239
<i>Zscore: P2. Experiencia previa ‘coding’</i>	-,58442	,25314	1,15052	-,86859
<i>Zscore: P5. Autoeficacia ‘coding’</i>	-1,62720	,04164	,29970	,14107
<i>Zscore: P6. Completar actividad HoC</i>	-,70526	,14073	,17565	-,02154
<i>Zscore: P8. Motivación proseguir aprendizaje ‘coding’</i>	-1,43225	-,72413	,28540	,27446
<i>Zscore: P9. Determinación aprender ‘coding’ on-line</i>	-1,66806	-2,02221	,49416	,49416

La matriz de centros finales es consistente con la estructura hipotetizada, y permite hacer la siguiente descripción de los clústeres:

- **Clúster 1:** son estudiantes sin exposición previa al ‘coding’ con anterioridad al evento, que tienen dificultades para completar la actividad HoC y una muy baja autoeficacia subsiguiente en tareas de ‘coding’ ($Zscore_{P5} = -1,62$); y no están motivados para continuar aprendiendo ‘coding’, mucho menos on-line. Se pueden etiquetar como ‘*novatos de baja autoeficacia-motivación*’ o, simplemente, ‘*no interesados*’
- **Clúster 2:** son estudiantes que han tenido exposición previa al ‘coding’ con anterioridad al evento, se sitúan en valores medios en lo relativo a completar la actividad HoC y en la autoeficacia subsiguiente; pero no están motivados para proseguir el aprendizaje del ‘coding’,

y bajo ningún concepto lo harían a través de plataformas-tutoriales on-line ($Z_{score_{P9}} = -2,02$). Se pueden etiquetar como ‘*veteranos de autoeficacia media y baja motivación*’ o, simplemente, ‘*desencantados*’

- **Clúster 3:** son estudiantes con mucha exposición y experiencia previa en ‘*coding*’ ($Z_{score_{P2}} = +1,15$), que no tienen dificultades para completar la actividad HoC y que manifiestan una buena autoeficacia subsiguiente; están motivados para seguir aprendiendo ‘*coding*’, y más a través de entornos ‘*on-line*’. Se pueden etiquetar como ‘*veteranos de alta autoeficacia-motivación*’ o, simplemente, ‘*enganchados*’
- **Clúster 4:** son estudiantes sin exposición previa al ‘*coding*’ con anterioridad al evento ($Z_{score_{P2}} = -0,86$), que se sitúan en valores medios en lo relativo a completar la actividad HoC y en la autoeficacia subsiguiente; y que se muestran motivados en continuar aprendiendo ‘*coding*’, incluido a través de plataformas-tutoriales ‘*on-line*’. Se pueden etiquetar como ‘*novatos de autoeficacia media y alta motivación*’ o, simplemente, ‘*entusiasmados*’

Así pues, el análisis de clúster realizado es comprensivo y comprensible: ha generado 4 conglomerados bien diferenciados en todas las variables incluidas en el análisis, y que sirven como categorías descriptivas fácilmente interpretables y susceptibles de ser reducidas a etiquetas sencillas. Como prueba estadística que lo refleja, en la Tabla D.22 del anexo se detalla el ANOVA que arroja diferencias muy significativas ($p_{(F)} = 0,000 \ll 0,05$) entre los clústeres para todas las variables incluidas en el análisis.

Tal y como señalábamos más arriba, en el transcurso del análisis guardamos el clúster de pertenencia de cada sujeto. En la Tabla 5.76 se muestra el número de sujetos que han sido asignados a cada clúster: tal y como puede observarse casi 8 de cada 10 sujetos (un 79,4%) se sitúa en los clústeres 3 y 4, ‘*enganchados*’ y ‘*entusiasmados*’ respectivamente, que tienen en común su alta motivación para proseguir el aprendizaje del ‘*coding*’. Sólo un 20,6% de sujetos se sitúan en los clústeres de baja motivación, el clúster 1 (10,0%) de ‘*no interesados*’ y el clúster 2 (10,7%) de ‘*desencantados*’.

Tabla 5.76. Número de sujetos asignados a cada clúster

		Número de caso de clúster			
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Clúster 1	135	8,1	10,0	10,0
	Clúster 2	144	8,7	10,7	20,6
	Clúster 3	485	29,2	35,9	56,5
	Clúster 4	588	35,4	43,5	100,0
	Total	1352	81,3	100,0	
Perdidos	Sistema	310	18,7		
Total		1662	100,0		

Según vemos en la Tabla D.23 del anexo, existen diferencias en la composición por sexo de cada clúster. Dichas diferencias son estadísticamente muy significativas ($\chi^2 = 29,955$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,149$; $p_{(C)} = 0,000 \ll 0,05$). Los clústeres 1, 2 y 4 son de ligero predominio femenino, mientras que el clúster 3 es de claro predominio masculino. Se ilustra en la siguiente Figura 5.40.

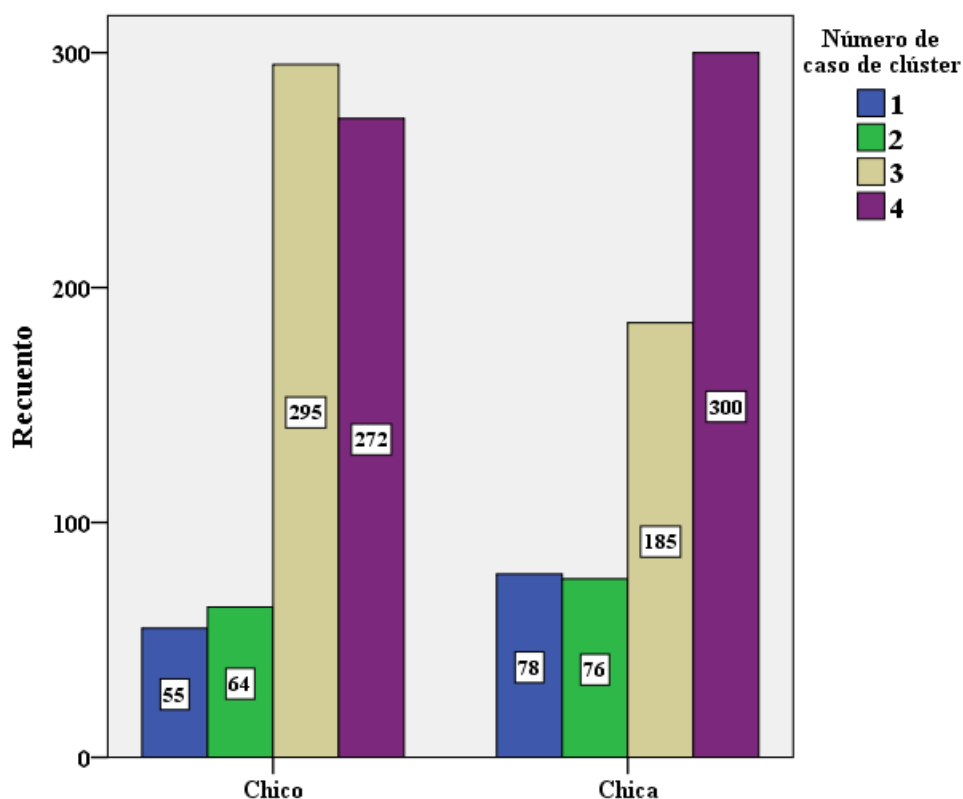


Figura 5.40. Distribución de frecuencias según sexo en los clústeres

Un análisis más refinado muestra que dicha distribución diferencial por sexos entre los clústeres emerge a partir de Educación Secundaria, en donde es estadísticamente muy significativa ($\chi^2 = 33,023$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,197$; $p_{(C)} = 0,000 \ll 0,05$); no así en Educación Primaria donde la composición por sexo de los clústeres es estadísticamente equivalente ($\chi^2 = 4,434$; $p_{(\chi^2)} = 0,218 > 0,05$; $C = 0,112$; $p_{(C)} = 0,218 > 0,05$). Se ilustra en la Tabla 5.77 y Figura 5.41.

Tabla 5.77. Tabla de contingencia sexo * clúster * etapa educativa

Etapa Educativa	Sexo	Número de caso de clúster	Número de caso de clúster				Total	χ^2	$P_{(\chi^2)}$
			1	2	3	4			
Primaria	Chico	Recuento	8	14	52	95	169	4,434	,218
		% dentro de Sexo	4,7%	8,3%	30,8%	56,2%	100,0%		
	Chica	Recuento	8	7	48	117	180		
		% dentro de Sexo	4,4%	3,9%	26,7%	65,0%	100,0%		
Secundaria	Chico	Recuento	44	33	201	151	429	33,023**	,000
		% dentro de Sexo	10,3%	7,7%	46,9%	35,2%	100,0%		
	Chica	Recuento	62	61	115	152	390		
		% dentro de Sexo	15,9%	15,6%	29,5%	39,0%	100,0%		
Bachillerato	Chico	Recuento	2	14	34	23	73	9,032*	,029
		% dentro de Sexo	2,7%	19,2%	46,6%	31,5%	100,0%		
	Chica	Recuento	7	8	19	30	64		
		% dentro de Sexo	10,9%	12,5%	29,7%	46,9%	100,0%		
FP	Chico	Recuento	1	2	7	1	11	1,791	,617
		% dentro de Sexo	9,1%	18,2%	63,6%	9,1%	100,0%		
	Chica	Recuento	0	0	2	1	3		
		% dentro de Sexo	0,0%	0,0%	66,7%	33,3%	100,0%		

* Significativo al nivel $p_{(\chi^2)} < 0,05$

** Significativo al nivel $p_{(\chi^2)} < 0,01$

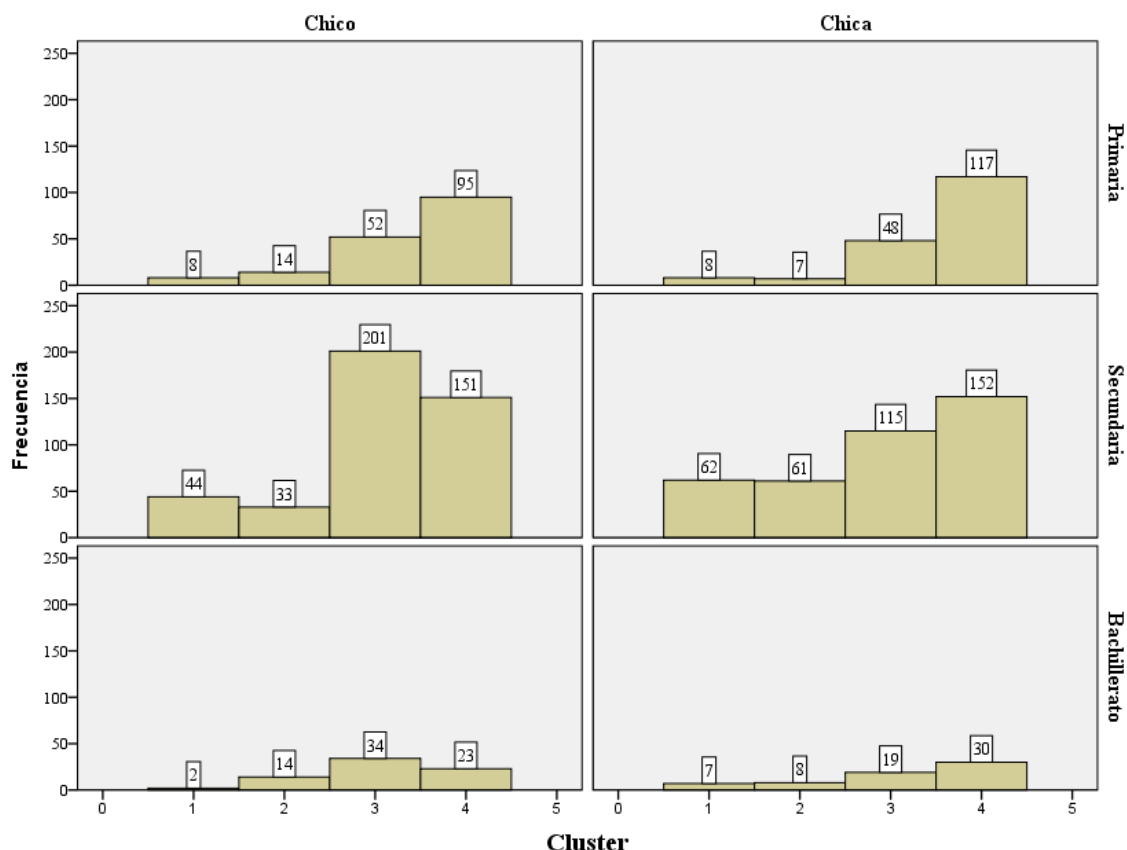


Figura 5.41. Distribución de frecuencias según sexo y etapa educativa en los clústeres

Finalmente, encontramos diferencias muy significativas en la distribución de sujetos en los clústeres, en función de la titularidad de su centro educativo ($\chi^2 = 21,395$; $p_{(\chi^2)} = 0,002 \ll 0,05$; $C = 0,126$; $p_{(C)} = 0,002 \ll 0,05$). Se detalla en la Figura 5.42 (y Tabla D.24 del anexo). Tal y como puede observarse, esencialmente en los centros privados hay predominio de sujetos del clúster 3 ('enganchados') mientras que en los centros públicos lo hay del clúster 4 ('entusiasmados').

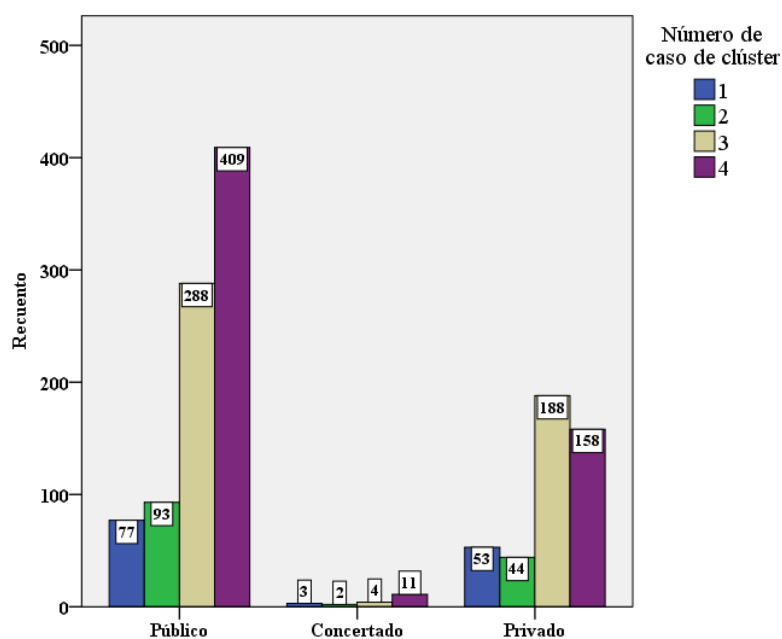


Figura 5.42. Distribución de frecuencias según titularidad de centro en los clústeres

b) Análisis de regresión múltiple

Aplicamos un análisis de regresión múltiple sobre la **Pregunta 8** (“P8. Tras la actividad, ¿cómo de motivado te sientes para continuar aprendiendo ‘programación informática’ (‘coding’)?”), o abreviadamente, “P8. Motivación proseguir aprendizaje ‘coding’”), que tomaremos entonces como nuestra variable dependiente (continua y de intervalo). Nuestra intención, pues, es construir un modelo explicativo-predictivo de la motivación final para seguir aprendiendo ‘coding’ tras la actividad HoC introductoria. En dicho modelo se pueden incluir, en principio, tantas variables independientes como se desee; ya sean de intervalo, ordinales, e incluso nominales (en este caso deben ser transformadas previamente en variables ‘dummy’). Ahora bien, se trata de llegar a una solución parsimoniosa: maximizar la cantidad de varianza (R^2) de la variable dependiente explicada por el modelo con el mínimo de variables independientes. Para nuestro análisis, utilizamos el método ‘por pasos’ e incluimos como variables independientes todas las preguntas previas a la Pregunta 8 (preguntas 1-7), además de ‘sexo’³⁴⁰, ‘etapa educativa’³⁴¹ y ‘titularidad’³⁴². En la Tabla D.25 del anexo se muestran los estadísticos descriptivos de las variables incluidas inicialmente.

Aplicando el método ‘por pasos’ (criterio: probabilidad de F para entrar $\leq 0,050$; probabilidad de F para eliminar $\geq 0,100$), en la siguiente Tabla 5.78 se muestra el progresivo orden de entrada de las variables independientes, en los sucesivos modelos explicativos.

Tabla 5.78. Variables introducidas y eliminadas en los sucesivos modelos de regresión según el método ‘por pasos’

Variables entradas/eliminadas ^a			
Modelo	Variables introducidas	Variables eliminadas	Método
1	P5. Autoeficacia ‘coding’	.	Por pasos (Criterios: Probabilidad-de-F-para-entrar $\leq 0,050$, Probabilidad-de-F- para-eliminar $\geq 0,100$).
2	P4.2 Desarrollar Creatividad	.	
3	Etapa Educativa Dummy	.	
4	P4.3 Desarrollar Resolución de Problemas	.	
5	P7.4 Rendimiento Tecnología / Informática	.	
6	P4.6 Desarrollar Capacidad de Atención	.	
7	P4.4 Desarrollar Trabajo en Equipo	.	
8	P4.1 Desarrollar Pensamiento Lógico	.	
9	P1. Haber oído ‘coding’	.	

a. Variable dependiente: P8. Motivación proseguir aprendizaje ‘coding’

En la siguiente Tabla 5.79 se presenta el resumen del modelo de regresión múltiple. Tal y como puede observarse, el modelo más parsimonioso es el **Modelo 4** (en negrita y fondo verde en la siguiente Tabla 5.79), dado que ya modelos posteriores no llegan a incrementar el R cuadrado ni siquiera en un 1% (Cambio de $R^2 < 0,01$). El Modelo 4 consigue, a través de sólo 4 variables independientes (P5. Autoeficacia ‘coding’, P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, y P4.3 Desarrollar Resolución de Problemas) correlacionar con la variable dependiente (P8. Motivación proseguir aprendizaje ‘coding’) de manera positiva y moderadamente alta ($r = +0,65$), explicando un 42,1% de su varianza ($R^2_{ajustado} = 0,421$).

³⁴⁰ Convertida en 0=chico, 1=chica

³⁴¹ Convertida en 0=Primaria, 1=Post-primaria

³⁴² Convertida en 0=Público, 1=Privado/Concertado

Tabla 5.79. Resumen del modelo de regresión múltiple

Modelo	R	R cuadrado	R cuadrado ajustado	Error estándar de la estimación	Estadísticas de cambios				
					Cambio de cuadrado de R	Cambio en F	df1	df2	Sig. Cambio en F
1	,502 ^a	,252	,252	,950	,252	357,543	1	1060	,000
2	,601 ^b	,361	,360	,878	,109	180,617	1	1059	,000
3	,631 ^c	,398	,396	,853	,037	64,308	1	1058	,000
4	,650^d	,423	,421	,835	,025	46,086	1	1057	,000
5	,656 ^e	,431	,428	,830	,008	14,645	1	1056	,000
6	,662 ^f	,438	,435	,825	,007	12,990	1	1055	,000
7	,665 ^g	,442	,439	,822	,005	8,530	1	1054	,004
8	,668 ^h	,446	,442	,820	,003	6,606	1	1053	,010
9	,669 ⁱ	,448	,443	,819	,002	4,688	1	1052	,031

a. Predictores: (Constante), P5. Autoeficacia 'coding'

b. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad

c. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy

d. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas

e. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas, P7.4 Rendimiento Tecnología / Informática

f. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas, P7.4 Rendimiento Tecnología / Informática, P4.6 Desarrollar Capacidad de Atención

g. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas, P7.4 Rendimiento Tecnología / Informática, P4.6 Desarrollar Capacidad de Atención, P4.4 Desarrollar Trabajo en Equipo

h. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas, P7.4 Rendimiento Tecnología / Informática, P4.6 Desarrollar Capacidad de Atención, P4.4 Desarrollar Trabajo en Equipo, P4.1 Desarrollar Pensamiento Lógico

i. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas, P7.4 Rendimiento Tecnología / Informática, P4.6 Desarrollar Capacidad de Atención, P4.4 Desarrollar Trabajo en Equipo, P4.1 Desarrollar Pensamiento Lógico, P1. Haber oído 'coding'

j. Variable dependiente: P8. Motivación proseguir aprendizaje 'coding'

En la Tabla 5.80 se muestra el ANOVA del Modelo 4, que indica su capacidad para detectar diferencias estadísticamente muy significativas ($p_{(F)} = 0,000 \ll 0,05$) en la variable dependiente.

Tabla 5.80. ANOVA del Modelo 4 de regresión múltiple

Modelo	ANOVA ^a					
	Suma de cuadrados	gl	Media cuadrática	F	Sig.	
4	Regresión	540,683	4	135,171	193,679**	,000 ^b
	Residuo	737,694	1057	,698		
	Total	1278,378	1061			

a. Variable dependiente: P8. Motivación proseguir aprendizaje 'coding'

b. Predictores: (Constante), P5. Autoeficacia 'coding', P4.2 Desarrollar Creatividad, Etapa Educativa Dummy, P4.3 Desarrollar Resolución de Problemas

** Significativo al nivel $p_{(F)} < 0,01$

A continuación, se muestra la tabla de coeficientes para el Modelo 4 (Tabla 5.81), que da una idea del peso relativo de cada una de las variables independientes para explicar-predecir la variable dependiente. Así, encontramos coeficientes estandarizados de signo positivo (a mayor valor de la variable independiente se predice un valor mayor en la dependiente), ordenados de mayor a menor peso: P5. Autoeficacia ‘coding’ ($\beta_t = 0,363$); P4.2 Desarrollar Creatividad ($\beta_t = 0,239$); y P4.3 Desarrollar Resolución de Problemas ($\beta_t = 0,182$). Y un coeficiente estandarizado de signo negativo (a mayor valor de la variable independiente se predice un valor menor en la dependiente): Etapa Educativa Dummy ($\beta_t = -0,182$). Tal y como puede observarse adicionalmente en la Tabla 5.81, se encuentran diferencias estadísticamente muy significativas en la variable dependiente ($p_{(t)} < 0,01$), para todas y cada una de las variables predictoras del modelo.

Tabla 5.81. Coeficientes para el Modelo 4^a de regresión múltiple

Modelo	Coeficientes no estandarizados		Coeficientes estandarizados	t	Sig.
	B	Error estándar	Beta		
(Constante)	1,094	,142		7,689**	,000
P5. Autoeficacia ‘coding’	,416	,028	,363	14,745**	,000
4 P4.2 Desarrollar Creatividad	,220	,025	,239	8,872**	,000
Etapa Educativa Dummy	-,461	,061	-,182	-7,577**	,000
P4.3 Desarrollar Resolución de Problemas	,160	,024	,182	6,789**	,000

a. Variable dependiente: P8. Motivación proseguir aprendizaje ‘coding’

** Significativo al nivel $p_{(t)} < 0,01$

Con respecto al comportamiento de los residuos del modelo de regresión múltiple, se detallan en la Tabla D.26 del anexo. Tal y como puede verse en las Figuras D.31 y D.32 del anexo, hay normalidad en la distribución de los residuos tipificados, lo cual apuntala la bondad de nuestro modelo de regresión.

De las Figuras D.33 a D.36 del anexo se ilustran los gráficos de regresión parcial de cada una de las variables independientes de nuestro Modelo 4, de mayor a menor peso en el modelo, sobre la variable dependiente. Tal y como puede observarse, las relaciones son directas en todos los casos, a excepción de la regresión de la ‘Etapa Educativa’ sobre la variable dependiente ‘P8. Motivación proseguir aprendizaje ‘coding’ en donde hay relación inversa.

Finalmente, como muestra de la potencia del modelo, en la Tabla 5.82 se muestra la correlación entre las puntuaciones observadas en la variable dependiente (P8. Motivación proseguir aprendizaje ‘coding’) y las puntuaciones predichas en dicha variable por el modelo de regresión múltiple. Tal y como era de esperar, es una correlación estadísticamente muy significativa, de signo positivo e intensidad moderadamente alta ($r = +0,656$)

Tabla 5.82. Correlación entre las puntuaciones observadas y predichas por el modelo en la variable dependiente

	Puntuaciones predichas por el modelo (sin estandarizar)
P8. Motivación proseguir aprendizaje ‘coding’ (puntuaciones observadas)	Correlación de Pearson ,656**
	Sig. (bilateral) ,000
	N 1221

** Correlación significativa al nivel $p_{(r)} < 0,01$

Y en la Figura 5.43 se ilustra el diagrama de dispersión entre dicho par de puntuaciones: observadas y predichas por el modelo en la variable dependiente. Como ya hemos visto anteriormente, el coeficiente de determinación (R^2), que indica el porcentaje de la varianza de la variable dependiente explicado por el modelo, se sitúa en el entorno del 43,0%.

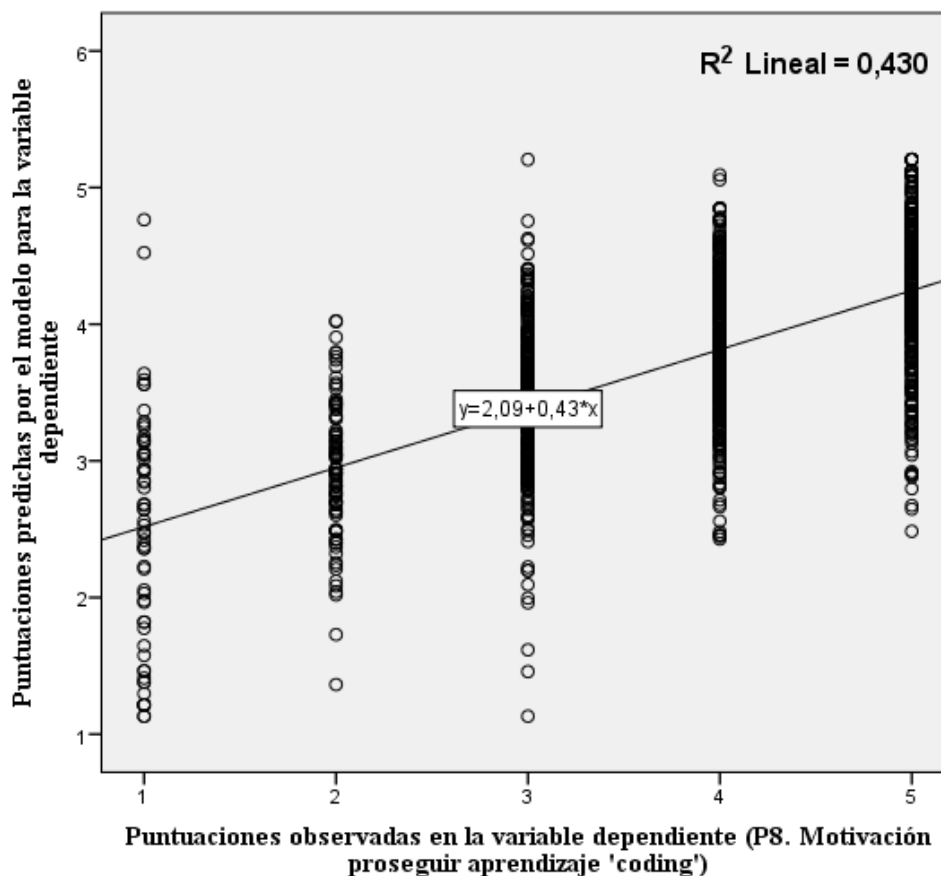


Figura 5.43. Diagrama de dispersión entre puntuaciones observadas-predichas por el modelo para la variable dependiente

c) Análisis discriminante

Realizamos un análisis discriminante sobre la **Pregunta 9** (“P9. ¿Te gustaría continuar aprendiendo 'programación informática' ('coding') a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)?”), o abreviadamente, “P9. Determinación aprender ‘coding’ on-line”), que será pues nuestra variable dependiente (variable categórica dicotómica). Buscamos una función discriminante que sea capaz de diferenciar y predecir entre los sujetos que contesten ‘sí’ o ‘no’ a dicha Pregunta 9, a partir de sus respuestas al resto de preguntas del cuestionario (variables independientes).

Se utilizará el método ‘por pasos’, que es el que ofrece una solución más parsimoniosa y elegante, al incluir en la función discriminante sólo las variables independientes con suficiente peso relativo para realizar la discriminación.

En la Tabla D.27 del anexo, se muestran los estadísticos descriptivos de todas las variables independientes inicialmente incluidas en el análisis, expresadas en función de la respuesta a la variable dependiente (P9. Determinación aprender ‘coding’ on-line).

En la Tabla D.28 del anexo, se presentan las pruebas de igualdad de medias entre los dos grupos posibles en la variable dependiente (“sí” o “no”), para todas las variables independientes incluidas inicialmente en el análisis. Tal y como puede observarse, para algunas variables independientes no se encuentran diferencias significativas ($p_{(F)} > 0,05$) entre los grupos de la dependiente; lo cual reafirma la necesidad de seguir el método ‘por pasos’ para nuestro análisis discriminante (que excluirá de la función discriminante todas aquellas variables independientes irrelevantes)

Acometemos entonces el análisis discriminante propiamente dicho, según el método ‘por pasos’ (criterio: la F mínima parcial para entrar es 3,84; la F máxima parcial para eliminar es 2,71). En la Tabla 5.83 se detallan las variables que sucesivamente van entrando en la función discriminante en cada escalón. Tras 4 pasos o escalones (Tabla 5.84), queda definida una función discriminante que incluye las siguientes variables independientes: P8. Motivación proseguir aprendizaje ‘coding’; P5. Autoeficacia ‘coding’; P4.1 Desarrollar Pensamiento Lógico; y P2. Experiencia previa ‘coding’.

Tabla 5.83. Variables que entran en la función discriminante en cada escalón

		Variables entradas/eliminadas ^{a,b,c,d}							
Escalón	Especificado	Lambda de Wilks				F exacta			
		Estadístico	df1	df2	df3	Estadístico	df1	df2	Sig.
1	<i>P8. Motivación aprendizaje ‘coding’</i>	,728	1	1	1044,000	390,095	1	1044,000	,000
2	<i>P5. Autoeficacia ‘coding’</i>	,720	2	1	1044,000	202,853	2	1043,000	,000
3	<i>P4.1 Desarrollar Pensamiento Lógico</i>	,717	3	1	1044,000	137,127	3	1042,000	,000
4	<i>P2. Experiencia previa ‘coding’</i>	,714	4	1	1044,000	104,360	4	1041,000	,000

En cada paso, se entra la variable que minimiza la lambda de Wilks global.

a. El número máximo de pasos es 40.

b. La F mínima parcial para entrar es 3.84.

c. La F máxima parcial para eliminar es 2.71.

d. El nivel F , la tolerancia o VIN no suficiente para un cálculo adicional.

Tabla 5.84. Variables que componen la función discriminante en cada escalón

		Variables en el análisis		
Escalón		Tolerancia	F para eliminar	Lambda de Wilks
1	<i>P8. Motivación aprendizaje ‘coding’</i>	1,000	390,095	
2	<i>P8. Motivación aprendizaje ‘coding’</i>	,835	238,463	,885
	<i>P5. Autoeficacia ‘coding’</i>	,835	11,637	,728
3	<i>P8. Motivación aprendizaje ‘coding’</i>	,762	189,300	,847
	<i>P5. Autoeficacia ‘coding’</i>	,829	10,234	,724
	<i>P4.1 Desarrollar Pensamiento Lógico</i>	,868	4,365	,720
4	<i>P8. Motivación aprendizaje ‘coding’</i>	,762	189,962	,844
	<i>P5. Autoeficacia ‘coding’</i>	,814	8,292	,719
	<i>P4.1 Desarrollar Pensamiento Lógico</i>	,862	5,110	,717
	<i>P2. Experiencia previa ‘coding’</i>	,978	4,626	,717

A continuación se resumen las características de la función discriminante obtenida. La función discriminante tiene una correlación canónica con la variable dependiente $C = +0,535$ (Tabla 5.85); el cuadrado de dicho coeficiente nos indica el porcentaje de la varianza total de la variable dependiente

explicado por la función discriminante (28,62%). En la Tabla 5.86 se comprueba que las medias de los dos grupos generados por la función discriminante (centroides) difieren de manera estadísticamente muy significativa ($\chi^2 = 351,346$; $p_{(\chi^2)} = 0,000 \ll 0,05$). Y en la Tabla 5.87 se aportan los coeficientes estandarizados de la función discriminante, que indican la aportación relativa de cada variable independiente a la función; de mayor a menor: P8. Motivación aprendizaje ‘coding’ (0,841); P5. Autoeficacia ‘coding’ (0,184); P4.1 Desarrollar Pensamiento Lógico (0,141); y P2. Experiencia previa ‘coding’ (0,126).

Tabla 5.85. Autovalores de la función discriminante

Función	Autovalores			Correlación canónica
	Autovalor	% de varianza	% acumulado	
1	,401 ^a	100,0	100,0	,535

a. Se utilizaron las primeras 1 funciones discriminantes canónicas en el análisis.

Tabla 5.86. Discriminación residual

Prueba de funciones	Lambda de Wilks	Chi-cuadrado	gl	Sig.
1	,714	351,346**	4	,000

** Significativo al nivel $p_{(\chi^2)} < 0,01$

Tabla 5.87. Coeficientes estandarizados de la función discriminante

Coeficientes de función discriminante canónica estandarizadas		Función
		1
<i>P8. Motivación aprendizaje ‘coding’</i>		,841
<i>P5. Autoeficacia ‘coding’</i>		,184
<i>P4.1 Desarrollar Pensamiento Lógico</i>		,141
<i>P2. Experiencia previa ‘coding’</i>		,126

Para acabar de caracterizar la función discriminante, en la siguiente Tabla 5.88 se muestran los coeficientes no estandarizados de la función, de mayor a menor; y en la Tabla 5.89 se aportan los valores de la función discriminante en los centroides de ambos grupos.

Tabla 5.88. Coeficientes no estandarizados de la función discriminante

Coeficientes de la función discriminante canónica		Función
		1
<i>P8. Motivación aprendizaje ‘coding’</i>		,900
<i>P2. Experiencia previa ‘coding’</i>		,254
<i>P5. Autoeficacia ‘coding’</i>		,204
<i>P4.1 Desarrollar Pensamiento Lógico</i>		,130
<i>(Constante)</i>		-4,713

Coeficientes no estandarizados

Tabla 5.89. Valores de la función discriminante en los centroides de grupo

Funciones en centroides de grupo		Función
		1
<i>P9. Determinación aprender ‘coding’ on-line</i>		
No		-1,317
Sí		,304

Las funciones discriminantes canónicas sin estandarizar se han evaluado en medias de grupos

Por tanto, las variables críticas para discriminar-predecir si los sujetos se mostrarán determinados a proseguir su aprendizaje del ‘coding’ a través de plataformas-tutoriales ‘on-line’ autoguiados son: su motivación por aprender ‘coding’, su experiencia previa en ‘coding’, su percepción de autoeficacia en ‘coding’ y su percepción de que el ‘coding’ contribuye a desarrollar el pensamiento lógico. Valores altos en las anteriores variables conllevan una alta probabilidad de responder sí a la variable dependiente; y viceversa.

Pasamos, finalmente, a aportar las estadísticas de clasificación de la función discriminante. En la Tabla D.29 del anexo se muestran las probabilidades previas de clasificación para cada uno de los grupos. Después, en la Tabla D.30 se detallan los coeficientes de función de clasificación para cada uno de dichos grupos.

Tal y como puede verse en la Tabla 5.90, nuestra función discriminante clasifica correctamente un 86,9% de los casos (celdas en fondo verde); lo que nos da una medida de su bondad para diferenciar y predecir la respuesta que un sujeto dará a la variable dependiente. Sólo un 13,1% de los casos (celdas en fondo rosa) son clasificados incorrectamente. Ahora bien, los porcentajes de clasificación son bien distintos entre el grupo que originalmente contestó que ‘sí’ a la variable dependiente (tan sólo 3,9% de casos clasificados incorrectamente); y el grupo que contestó que ‘no’ (ahí hay un 50,8% de casos clasificados incorrectamente: 135 sujetos ‘anómalos’ que veremos con más detalle)

Tabla 5.90. Estadísticas de clasificación de la función discriminante

Resultados de clasificación ^{ab}					
		<i>P9. Determinación aprender ‘coding’ on-line</i>	Pertenencia a grupos pronosticada		Total
			No	Sí	
Original	Recuento	No	131	135	266
		Sí	43	1054	1097
		Casos sin agrupar	3	18	21
%		No	49,2	50,8	100,0
		Sí	3,9	96,1	100,0
		Casos sin agrupar	14,3	85,7	100,0

a. 86,9% de casos agrupados originales clasificados correctamente (en verde)

b. 13,1% de casos agrupados originales clasificados incorrectamente (en rosa)

Dicho de otra manera, nuestra función es muy eficaz para discriminar entre aquellos sujetos que originalmente contestaron que ‘sí’ a la variable dependiente (en un 96,1% predice que así es). Queda ilustrado en la siguiente Figura 5.44.

Por el contrario, la función se muestra bastante poco eficaz para discriminar entre aquellos sujetos que originalmente contestaron que ‘no’ a la variable dependiente (sólo predice correctamente el 49,2% de los casos); según se ilustra en la Figura 5.45.

9. ¿Te gustaría continuar aprendiendo 'programación informática' ('coding') a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)? = Sí

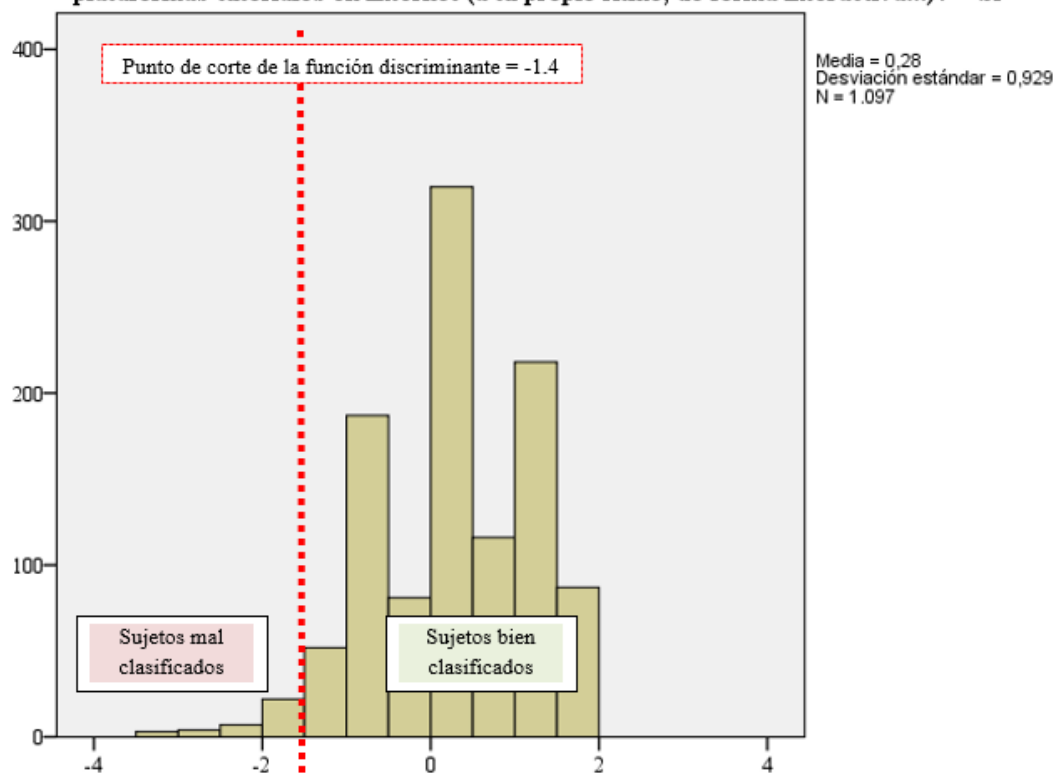


Figura 5.44. Puntuaciones en la función discriminante para los sujetos que contestaron originalmente 'sí' a la P9

9. ¿Te gustaría continuar aprendiendo 'programación informática' ('coding') a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)? = No

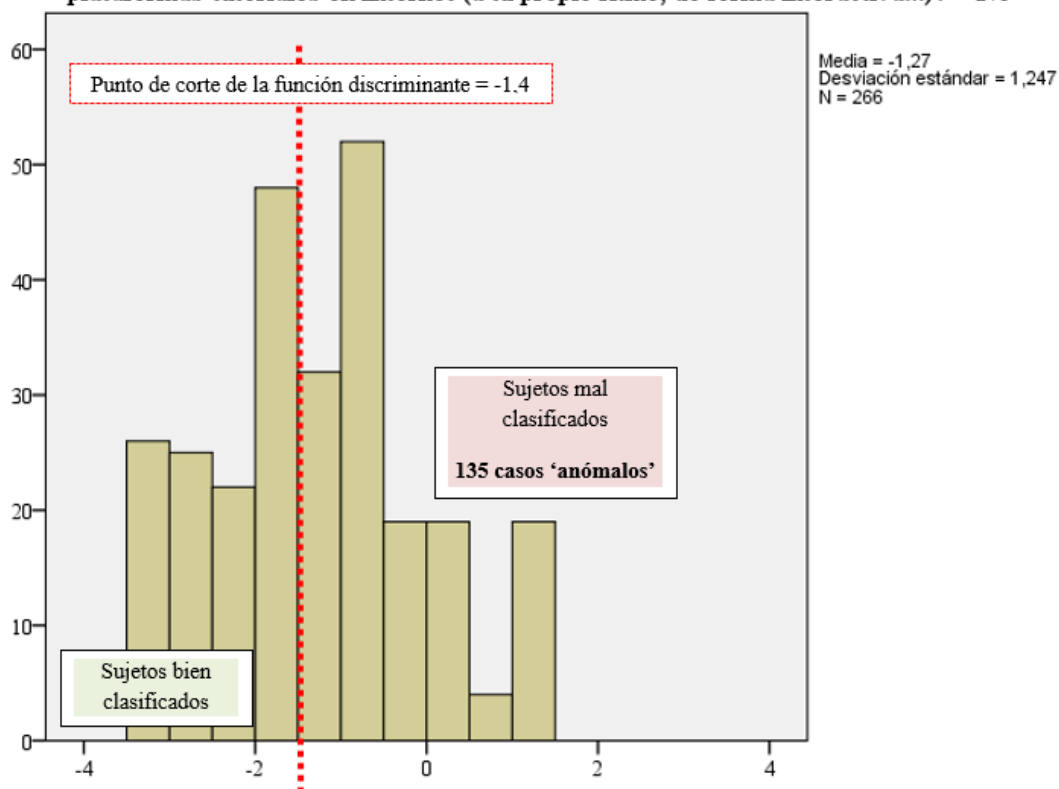


Figura 5.45. Puntuaciones en la función discriminante para los sujetos que contestaron originalmente 'no' a la P9

Concretamente, hay 135 sujetos ‘anómalos’ que, prediciendo la función discriminante que iban a contestar que ‘sí’ a la Pregunta 9 (es decir, sujetos con motivación, autoeficacia, y experiencia previa en ‘coding’ en niveles suficientes); finalmente contestaron que ‘no’. ¿Cómo son estos 135 sujetos ‘anómalos’? Según puede verse en las Figuras 5.46 a 5.48 (y Tablas D.31 a D.33 del anexo) es un grupo significativamente femenino ($\chi^2 = 4,372$; $p_{(\chi^2)} = 0,037 < 0,05$; $C = 0,052$; $p_{(C)} = 0,037 < 0,05$), significativamente de Educación Secundaria ($\chi^2 = 11,123$; $p_{(\chi^2)} = 0,011 < 0,05$; $C = 0,082$; $p_{(C)} = 0,011 < 0,05$), y muy significativamente compuesto por sujetos del clúster 1 (‘no interesados’) y, especialmente, del clúster 2 (‘desencantados’) ($\chi^2 = 648,172$; $p_{(\chi^2)} = 0,000 << 0,05$; $C = 0,569$; $p_{(C)} = 0,000 << 0,05$).

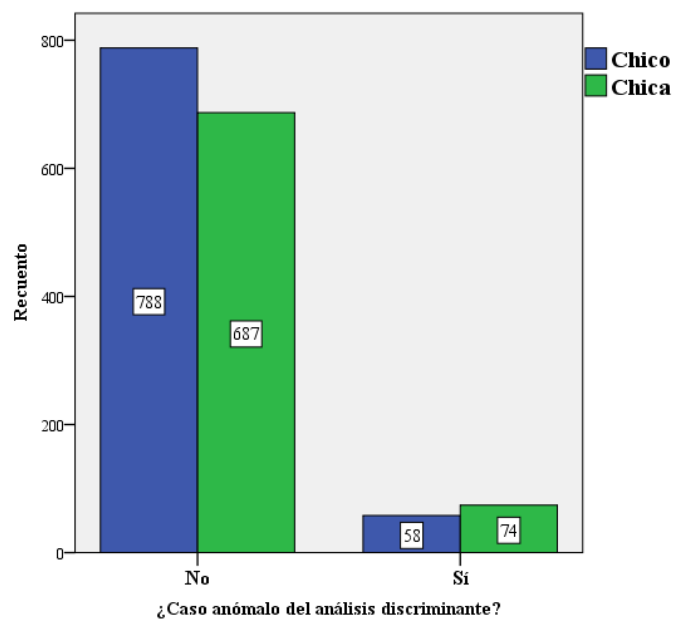


Figura 5.46. Distribución de casos anómalos del análisis discriminante por sexo

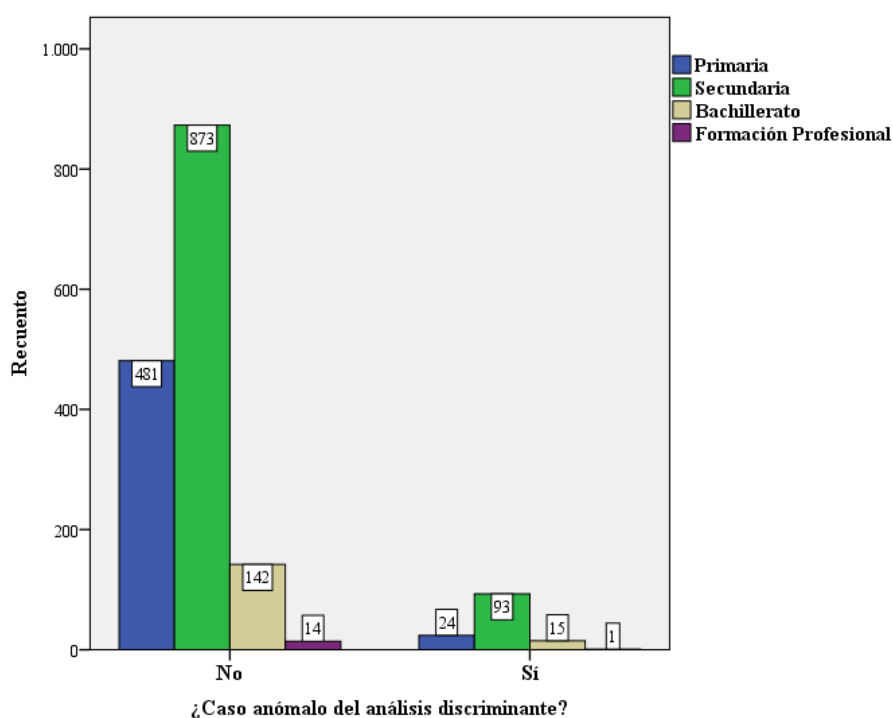


Figura 5.47. Distribución de casos anómalos del análisis discriminante por etapa educativa

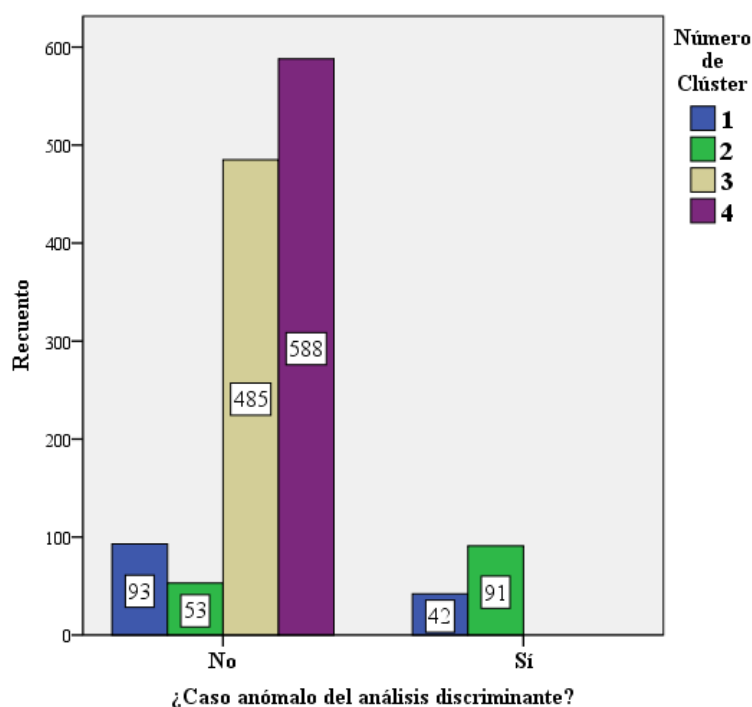


Figura 5.48. Distribución de casos anómalos del análisis discriminante por clúster de pertenencia

Con respecto a la **Pregunta 10** del cuestionario [*“P10. En caso afirmativo a la respuesta anterior, existen en Internet diversas plataformas-tutoriales para el aprendizaje de la ‘programación informática’ (‘coding’), ¿Cuál/es de ellas conoces?”*], que aborda el objetivo específico de investigación **O_{2Ej}**: en la siguiente Tabla 5.91 se muestran los porcentajes de penetración de cada una de ellas, tanto en relación al total de la muestra (N = 1.662), como en relación a la sub-muestra que contestó afirmativamente a la Pregunta 9 (N = 1.154).

Tabla 5.91. Porcentajes de penetración de las distintas plataformas-tutoriales ‘on-line’ de aprendizaje del ‘coding’

Plataformas-tutoriales on-line de aprendizaje del ‘coding’	Frecuencia	Penetración (%) sobre el total de la muestra (N=1662)	Penetración (%) sobre la sub-muestra con motivación por seguir aprendiendo ‘coding’ online (N=1154)
Code.org (http://code.org)*	682	41,03%	59,10%
Scratch (http://scratch.mit.edu)*	513	30,87%	44,45%
Light-Bot (http://light-bot.com)*	156	9,39%	13,52%
App Inventor (http://appinventor.mit.edu/explore)*	143	8,60%	12,39%
Khan Academy (https://www.khanacademy.org/cs)*	113	6,80%	9,79%
Codecademy (http://www.codecademy.com)*	104	6,26%	9,01%
Tynker (http://www.tynker.com)*	91	5,48%	7,89%
Google Blockly (http://code.google.com/p/blockly)*	76	4,57%	6,59%
CodeHS (http://codehs.com)*	75	4,51%	6,50%

Plataformas-tutoriales on-line de aprendizaje del 'coding'	Frecuencia	Penetración (%) sobre el total de la muestra (N=1662)	Penetración (%) sobre la sub-muestra con motivación por seguir aprendiendo 'coding' online (N=1154)
Pivot Animator (https://en.wikipedia.org/wiki/Pivot_Animator)**	6	0,36%	0,52%
Game Maker (http://www.yoyogames.com/studio)**	4	0,24%	0,35%
Lego (http://education.lego.com/es-ar/learn)**	4	0,24%	0,35%
The Foos (http://thefoos.com/)**	4	0,24%	0,35%
Agame (http://www.agame.com/)**	2	0,12%	0,17%
Blender (https://www.blender.org/)**	2	0,12%	0,17%
Unity 3D (https://unity3d.com/es)**	2	0,12%	0,17%
W3Schools (http://www.w3schools.com/)**	2	0,12%	0,17%
3D RAD (http://www.3drad.com/)**	1	0,06%	0,09%
Alice (http://www.alice.org/index.php)**	1	0,06%	0,09%
Android Studio (http://developer.android.com/tools/studio/index.html)**	1	0,06%	0,09%
Eclipse (http://www.eclipse.org/)**	1	0,06%	0,09%
GrokLearning (https://groklearning.com/)**	1	0,06%	0,09%
Hello Processing (http://hello.processing.org/)**	1	0,06%	0,09%
Just Basic (http://justbasic.com/)**	1	0,06%	0,09%
Kodable (https://www.kodable.com/)**	1	0,06%	0,09%
Kodu (http://www.kodugamelab.com/)**	1	0,06%	0,09%
Make School (https://www.makeschool.com/)**	1	0,06%	0,09%
Minecraft (https://minecraft.net/)**	1	0,06%	0,09%
Moway (http://moway-robot.com/)**	1	0,06%	0,09%
RPG Maker (http://www.rpgmakerweb.com/)**	1	0,06%	0,09%
Small Basic (http://smallbasic.com/)**	1	0,06%	0,09%
Treehouse (https://teamtreehouse.com/)**	1	0,06%	0,09%
XCode (https://developer.apple.com/xcode/)**	1	0,06%	0,09%

* Plataforma-tutorial de programación explícitamente listado entre las posibles opciones a marcar en la Pregunta 10

** Plataformas-tutoriales no explícitamente listados, que se mencionaron por los sujetos en la categoría 'Otros'

Tal y como se puede ver en la siguiente Figura 5.49, se detectan 3 grupos de plataformas-tutoriales según su porcentaje de penetración: a) el formado por Code.org y Scratch, con una alta penetración que se sitúa en el orden del 30-40% del total de la muestra; b) el formado por Light-Bot, App Inventor, Khan Academy, Codecademy, Tynker, Blockly y CodeHS, con una penetración moderada en el orden del 5-10%; y c) el resto de plataformas-tutoriales, con una baja penetración que no llega al 1% sobre el total de la muestra (si bien hay que recordar que ninguna de ellas estaba listada explícitamente en el cuestionario)

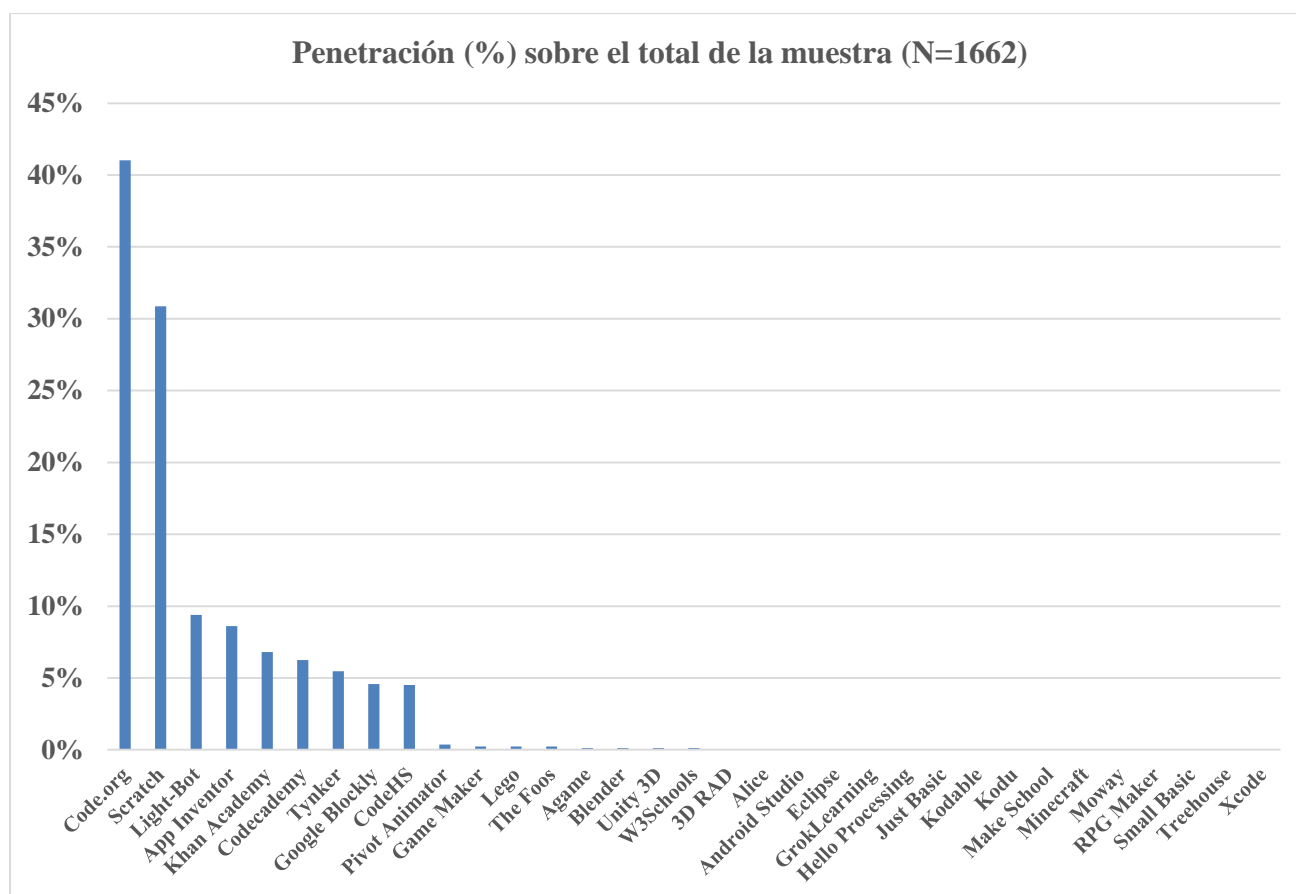


Figura 5.49. Porcentajes de penetración de las distintas plataformas-tutoriales ‘on-line’ sobre total de muestra (N=1662)

En este punto consideramos de interés realizar algunas comparaciones entre el grupo de estudiantes que declaró conocer únicamente la plataforma Code.org (les llamaremos ‘coders’) y el que hizo lo propio con la plataforma Scratch (les llamaremos ‘scratchers’). Esta comparación asume, consecuentemente, que los ‘coders’ hicieron la actividad HoC propuesta por Code.org³⁴³, y los ‘scratchers’ hicieron la actividad HoC propuesta por Scratch³⁴⁴; y se apoya en dos hechos importantes:

- Que las plataformas Code.org y Scratch son las que tienen, de largo, un mayor porcentaje de penetración en la muestra de estudiantes; y, por tanto, son las que merecen una atención más detallada por parte de nuestra investigación.
- Que los tutoriales, y metodología asociada, que proponen Code.org y Scratch son muy diferentes:
 - En el caso de Code.org se trata de puzzles-retos de programación (Figura 5.50), es decir, entornos de resolución de problemas cerrados o ‘close-ended’ (p.e. laberintos) en los que el estudiante debe formular la solución a través de código informático; avanzando por sucesivos puzzles de dificultad progresiva por ensayo y error, y obteniendo *feed-back* inmediato. Son tutoriales ‘self-paced’ (es decir, autoguiados al ritmo del propio estudiante), y según nos relataron los propios profesores, son

³⁴³ <http://hourofcode.com/code>

³⁴⁴ <http://hourofcode.com/sc>

realizados en muchas ocasiones por parejas de estudiantes (*'pair programming'*) que comparten dispositivo. La metodología recomendada desde Code.org para su implantación en el aula es la denominada *'Ask 3, then me'*: si un estudiante no sabe cómo resolver un determinado puzle, debe preguntar al menos a 3 compañeros antes de acudir al profesor.

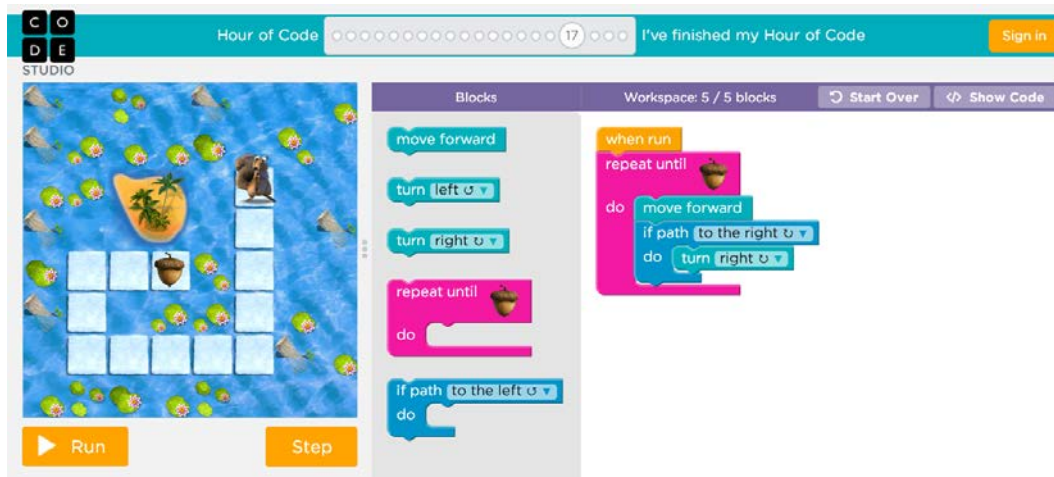


Figura 5.50. Tutorial HoC de Code.org. A la izquierda, el puzle de programación propuesto (llevar a la ardilla hasta la bellota); a la derecha, entorno de trabajo del estudiante en el que formula la solución a través de bloques de programación

- En el caso de Scratch se trata de proyectos creativos *'open-ended'* (Figura 5.51), es decir, el estudiante parte de un *'lienzo en blanco'* en el cual, a partir de bloques de código, puede programar animaciones, historias, videojuegos, etc... Para ello, sobre todo con estudiantes que todavía no han adquirido un nivel mínimo de competencia en Scratch, el aprendiz suele guiarse por plantillas de otros proyectos o secuencias guiadas *'de pasos'* que en muchas ocasiones son proyectadas por el profesor para todo el aula. En ese sentido, podríamos decir que es un entorno más *'teacher-paced'* (guiado por el ritmo del profesor que va instruyendo al estudiante sobre los pasos que debe seguir para programar un proyecto en Scratch); aunque, según los estudiantes ganan en competencia, pueden potencialmente irse liberando de esta guía docente.

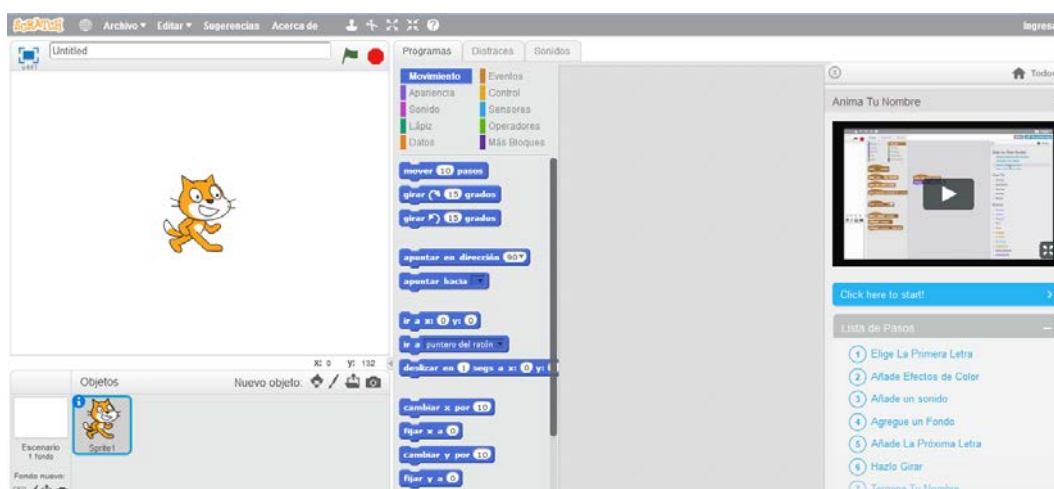


Figura 5.51. Tutorial HoC de Scratch. A la izquierda-centro, área de trabajo del estudiante; a la derecha, video-tutorial explicativo y lista de pasos a seguir para la elaboración guiada de un proyecto.

En la Figura 5.52 (y Tabla D.34 del anexo) se muestra la distribución de ‘coders’ y ‘scratchers’ por sexo, en donde no se encuentran diferencias significativas ($\chi^2 = 2,557$; $p_{(\chi^2)} = 0,110 > 0,05$; $C = 0,064$; $p_{(C)} = 0,110 > 0,05$). En la Figura 5.53 (y Tabla D.35 del anexo) se muestra la distribución de ‘coders’ y ‘scratchers’ por etapa educativa, en este caso sí que encontramos diferencias estadísticamente muy significativas ($\chi^2 = 32,726$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,222$; $p_{(C)} = 0,000 \ll 0,05$). En síntesis, mientras que encontramos ‘coders’ en Primaria y Secundaria; los ‘scratchers’ sólo aparecen masivamente a partir de Secundaria.

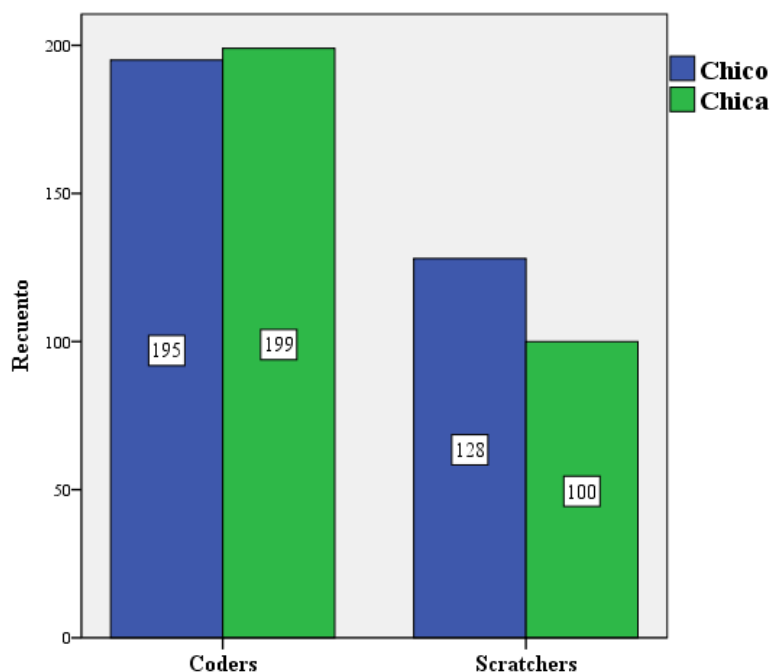


Figura 5.52. Distribución de ‘coders’ y ‘scratchers’ según sexo

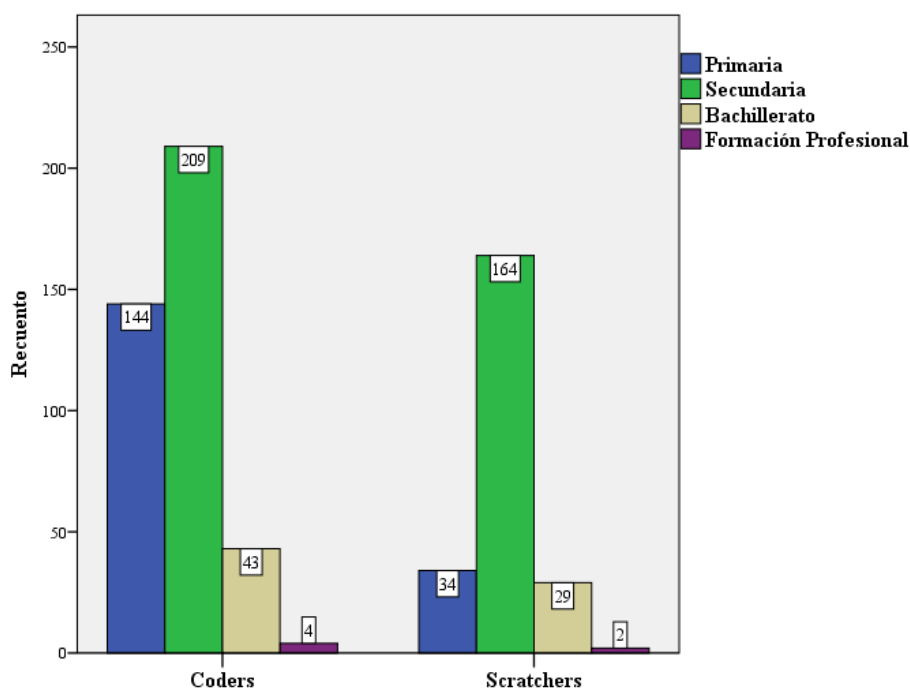


Figura 5.53. Distribución de ‘coders’ y ‘scratchers’ según etapa educativa

En la siguiente Tabla 5.92 y Figura 5.54 se muestran las medias de ‘coders’ y ‘scratchers’ en sus respuestas a la Pregunta 4 [“P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?”]. Tal y como puede observarse, se encuentran diferencias significativas, a favor de los ‘coders’, en la percepción de cómo la actividad ha contribuido a desarrollar el ‘pensamiento lógico’ ($p_{(t)} = 0,028 < 0,05$); y muy significativas ($p_{(t)} \ll 0,05$), igualmente a favor de los ‘coders’, en ‘resolución de problemas’, ‘trabajo en equipo’, ‘persistencia-perseverancia’ y ‘capacidad de atención’. En la única aptitud-habilidad en la cual no se encuentran diferencias significativas es, precisamente, en ‘creatividad’ ($p_{(t)} = 0,106 > 0,05$), consistentemente con nuestra descripción previa de la naturaleza creativa del entorno Scratch.

Tabla 5.92. Diferencias en la Pregunta 4, en función de ser ‘coder’ o ‘scratcher’

P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?		N	Media	Desviación estándar	t	P(t)	d de Cohen
Pensamiento Lógico	Coders	397	3,81	1,052	2,203*	,028	0,18
	Scratchers	228	3,62	1,090			
Creatividad	Coders	389	3,77	1,108	1,620	,106	0,13
	Scratchers	224	3,62	1,161			
Resolución de Problemas	Coders	387	3,68	1,142	3,120**	,002	0,25
	Scratchers	223	3,39	1,137			
Trabajo en Equipo	Coders	389	3,35	1,477	4,214**	,000	0,35
	Scratchers	228	2,84	1,405			
Persistencia-Perseverancia	Coders	386	3,67	1,161	3,873**	,000	0,33
	Scratchers	224	3,29	1,149			
Capacidad de Atención	Coders	390	3,98	1,128	3,235**	,001	0,27
	Scratchers	228	3,66	1,218			

* Significativo al nivel $p_{(t)} < 0,05$

** Significativo al nivel $p_{(t)} < 0,01$

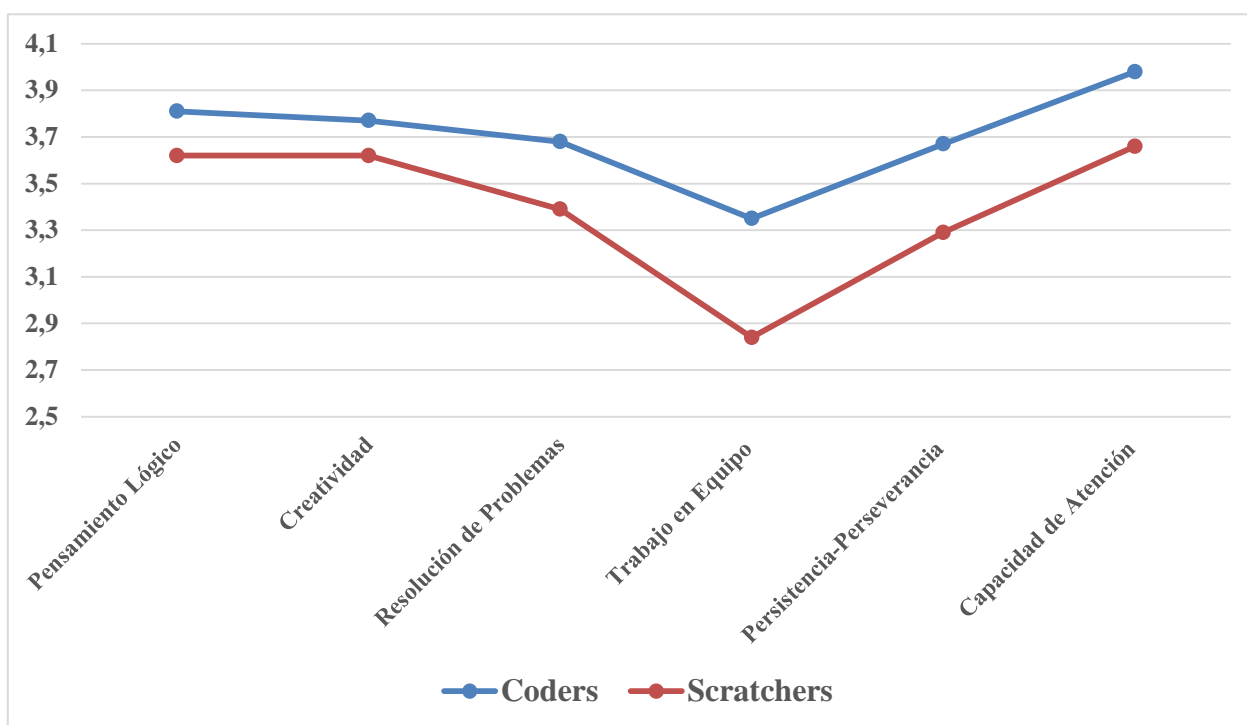


Figura 5.54. Medias en la Pregunta 4 para ‘coders’ y ‘scratchers’

Si segmentamos la gráfica anterior para Primaria y Secundaria (Figura 5.55), se comprueba que las diferencias son especialmente reseñables en la etapa de Primaria. Más concretamente, sólo se encuentran diferencias estadísticamente significativas en Primaria, a favor de los ‘coders’, en ‘trabajo en equipo’ ($t = 2,262$; $p_{(t)} = 0,025 < 0,05$) y, muy significativa, en ‘persistencia-perseverancia’ ($t = 2,825$; $p_{(t)} = 0,005 \ll 0,05$). En Secundaria sólo se halla una diferencia en el límite de la significatividad, a favor de los ‘coders’, en ‘trabajo en equipo’ ($t = 1,902$; $p_{(t)} = 0,058 \approx 0,05$),

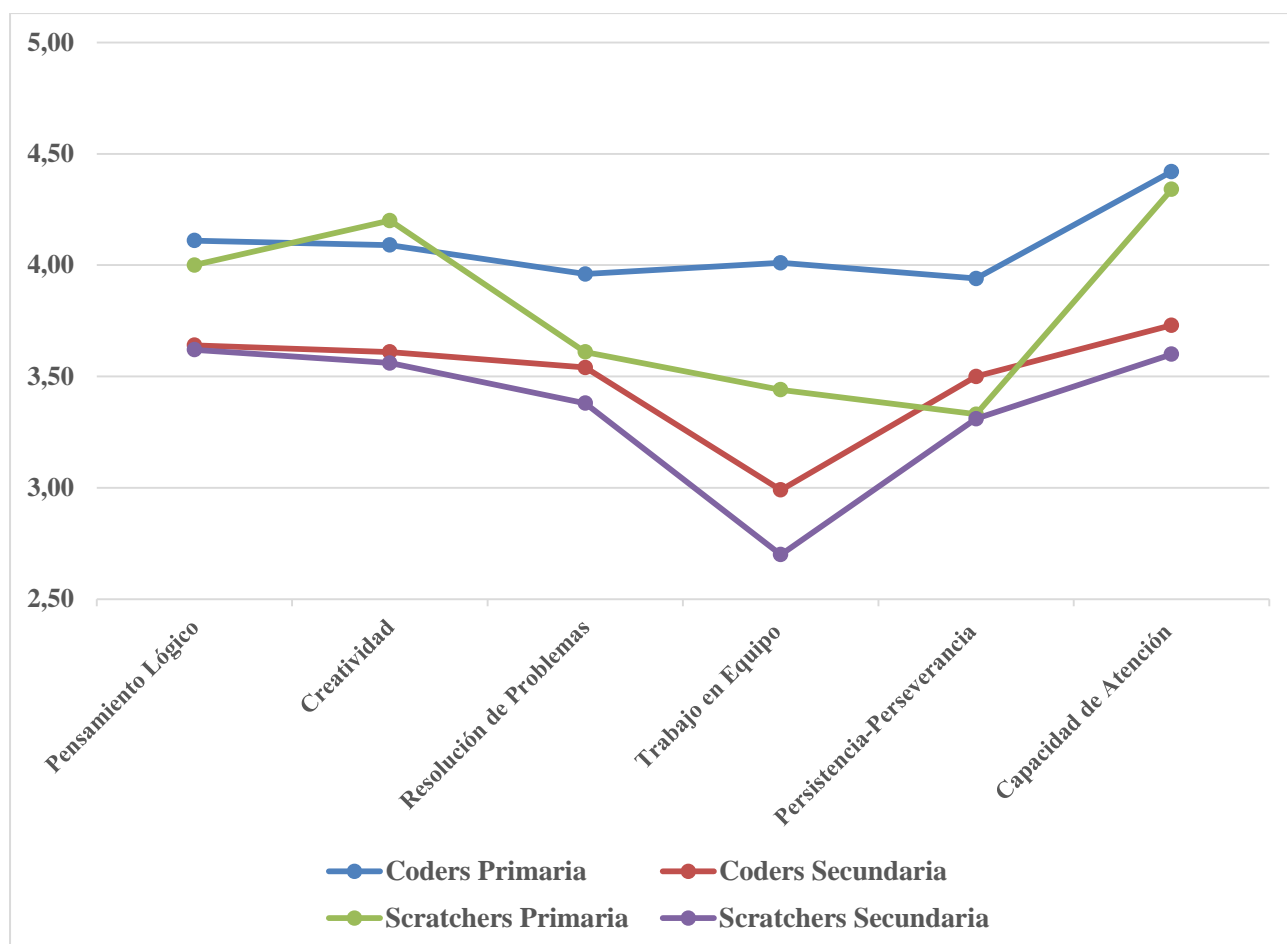


Figura 5.55. Medias en la Pregunta 4 para ‘coders’ y ‘scratchers’, diferenciado para Primaria y Secundaria

En esta misma línea de análisis, en la Tabla 5.93 se comparan las medias de ‘coders’ y ‘scratchers’ en sus respuestas a las preguntas 5, 8 y 9.

Tabla 5.93. Diferencias en las preguntas 5, 8 y 9; en función de ser ‘coder’ o ‘scratcher’

	Coders vs. Scratchers	N	Media	Desviación estándar	t	p _(t)	d de Cohen
P5. Autoeficacia ‘coding’	Coders	395	4,01	,820	3,282**	,001	0,28
	Scratchers	226	3,77	,895			
P8. Motivación proseguir aprendizaje ‘coding’	Coders	396	3,99	,915	3,806**	,000	0,32
	Scratchers	227	3,68	1,008			
P9. Determinación aprender ‘coding’ on-line	Coders	397	,96	,191	2,826**	,005	0,24
	Scratchers	230	,90	,301			

** Significativo al nivel $p_{(t)} < 0,01$

Tal y como puede observarse, encontramos diferencias estadísticamente muy significativas ($p_{(t)} \ll 0,05$), a favor del grupo de *'coders'*, tanto en percepción de autoeficacia posterior a la actividad, como en motivación por continuar aprendiendo programación, y en hacerlo particularmente *'on-line'*. En ese sentido, parece que los tutoriales de Code.org generaron una mejor respuesta de autoeficacia y motivación subsiguiente en los estudiantes, que los tutoriales de Scratch (Figura 5.56).

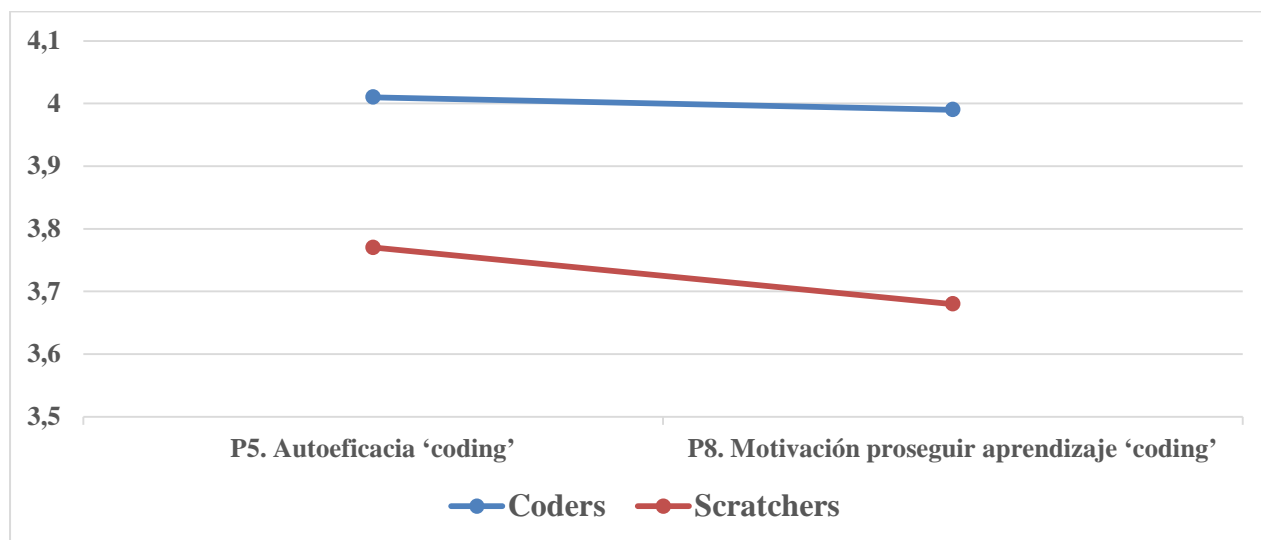


Figura 5.56. Medias en las preguntas 5 y 8 para *'coders'* y *'scratchers'*

Segmentada la gráfica anterior por etapa educativa (Figura 5.57), comprobamos que dichas diferencias a favor de los *'coders'* emergen en Secundaria, que es la única etapa en la que fueron estadísticamente significativas, tanto para la pregunta 5 de autoeficacia ($t = 2,220$; $p_{(t)} = 0,027 < 0,05$), como en la pregunta 8 de motivación ($t = 2,409$; $p_{(t)} = 0,016 < 0,05$).

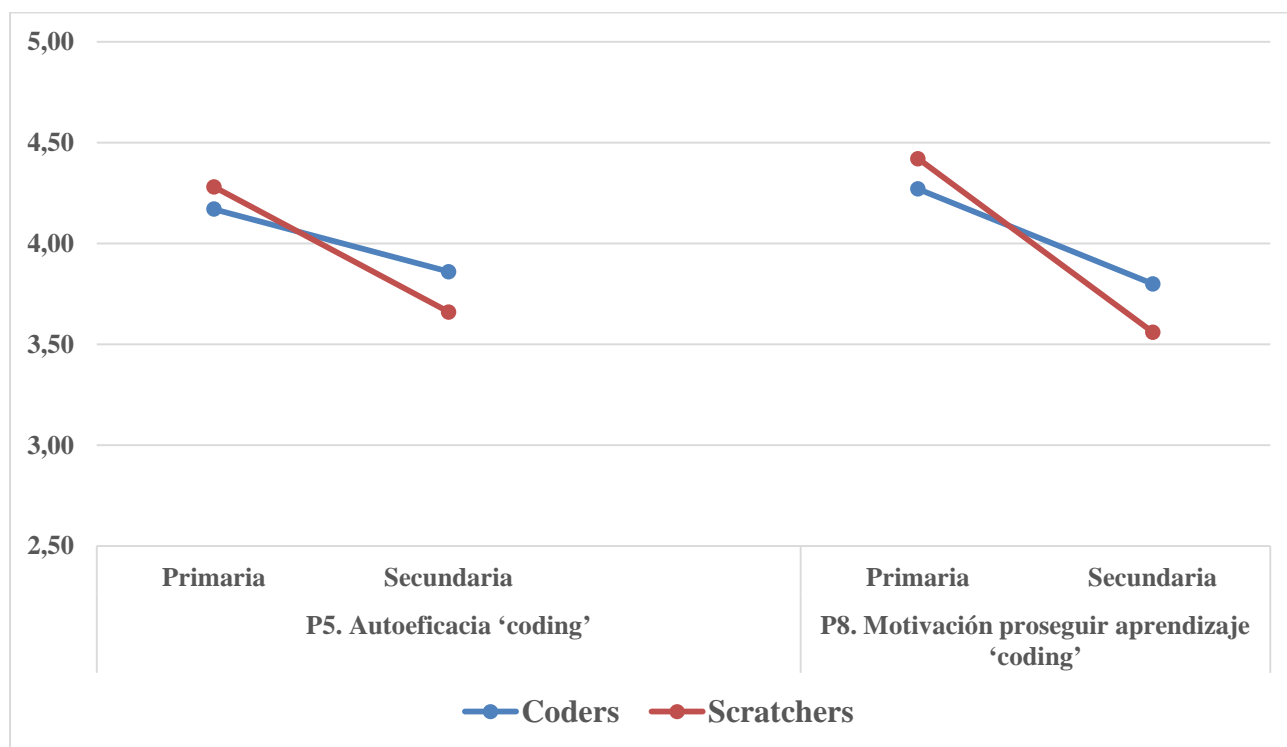


Figura 5.57. Medias en las preguntas 5 y 8 para *'coders'* y *'scratchers'*, diferenciado para Primaria y Secundaria

Con respecto a la **Pregunta 11** [*“P11. ¿Consideras importante la ‘programación informática’ (‘coding’) para tu formación personal, como ciudadano del siglo XXI? ¿Sí, No? ¿Por qué?”*], y la **Pregunta 12** del cuestionario [*“P12. ¿Consideras importante la ‘programación informática’ (‘coding’) para tu formación profesional, como trabajador del siglo XXI? ¿Sí, No? ¿Por qué?”*]; que abordan respectivamente los objetivos de investigación **O_{2Ek}** y **O_{2El}**: ambas son preguntas de respuesta abierta. Se aportan los enlaces a sendos documentos con el conjunto de respuestas emitidas por los estudiantes a cada una de las preguntas³⁴⁵; codificadas y ordenadas según se sitúen en el entorno del ‘sí’, en el entorno del ‘no’, o en el entorno del ‘tal vez / no sé / depende’ (Figura 5.58)



Figura 5.58. Códigos QR a sendos PDF con el conjunto de respuestas emitidas en las preguntas 11 y 12, codificadas y ordenadas según se sitúen en el ‘sí’, el ‘no’, o el ‘tal vez / no sé / depende’

Desde una aproximación cuantitativa, en las Tablas 5.94 y 5.95 (y Figuras D.37 y D.38 del anexo), se presentan las frecuencias de respuesta respectivas a cada pregunta.

Tabla 5.94. Distribución de frecuencias para la Pregunta 11

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
<i>P11. ¿Consideras importante la 'programación informática' ('coding') para tu formación personal, como ciudadano del siglo XXI? ¿Sí, No? ¿Por qué?</i>					
Válido	No	137	8,2	10,5	10,5
	Tal vez / No lo sé / Depende	24	1,4	1,8	12,4
	Sí	1142	68,7	87,6	100,0
	Total	1303	78,4	100,0	
Perdidos	Sistema	359	21,6		
Total		1662	100,0		

Tabla 5.95. Distribución de frecuencias para la Pregunta 12

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
<i>P12. ¿Consideras importante la 'programación informática' ('coding') para tu formación profesional, como trabajador del siglo XXI? ¿Sí, No? ¿Por qué?</i>					
Válido	No	154	9,3	13,2	13,2
	Tal vez / No lo sé / Depende	54	3,2	4,6	17,9
	Sí	957	57,6	82,1	100,0
	Total	1165	70,1	100,0	
Perdidos	Sistema	497	29,9		
Total		1662	100,0		

³⁴⁵ Entre ambos documentos (uno para las respuestas a la Pregunta 11: <https://db.tt/Xc0tFhc9>, y otro para las respuestas a la Pregunta 12: <https://db.tt/TggPuFZQ>) suman casi 100 páginas, por lo que se ha optado en ubicarlos en la nube de manera accesible para el lector; y no como anexos de la tesis doctoral.

Tal y como puede observarse, el porcentaje del 'sí' es abrumador en ambas preguntas, si bien es algo superior en la relativa a la relevancia del 'coding' para la formación personal (como ciudadano) del estudiante (87,6%); que en la relativa a la relevancia del 'coding' para la formación profesional (como trabajador) del estudiante (baja al 82,1%).

Tal y como era de esperar, encontramos una asociación estadísticamente muy significativa entre las respuestas codificadas a ambas preguntas ($\chi^2 = 225,693$; $p_{(\chi^2)} = 0,000 \ll 0,05$; $C = 0,405$; $p_{(C)} = 0,000 \ll 0,05$). Especialmente se encuentra contingencia entre los que responden que 'sí' a ambas preguntas (Tabla 5.96 y Figura 5.59).

Tabla 5.96. Tabla de contingencia Pregunta 11 * Pregunta 12

		P12. Relevancia 'coding' para formación profesional			Total	
		No	Tal vez / No lo sé / Depende	Sí		
P11. Relevancia 'coding' para formación personal	No	Recuento	63	9	51	123
		% dentro de P11	51,2%	7,3%	41,5%	100,0%
	Tal vez / No lo sé / Depende	Recuento	5	7	9	21
		% dentro de P11	23,8%	33,3%	42,9%	100,0%
	Sí	Recuento	85	38	884	1007
		% dentro de P11	8,4%	3,8%	87,8%	100,0%
Total	Recuento	153	54	944	1151	
	% dentro de P11	13,3%	4,7%	82,0%	100,0%	

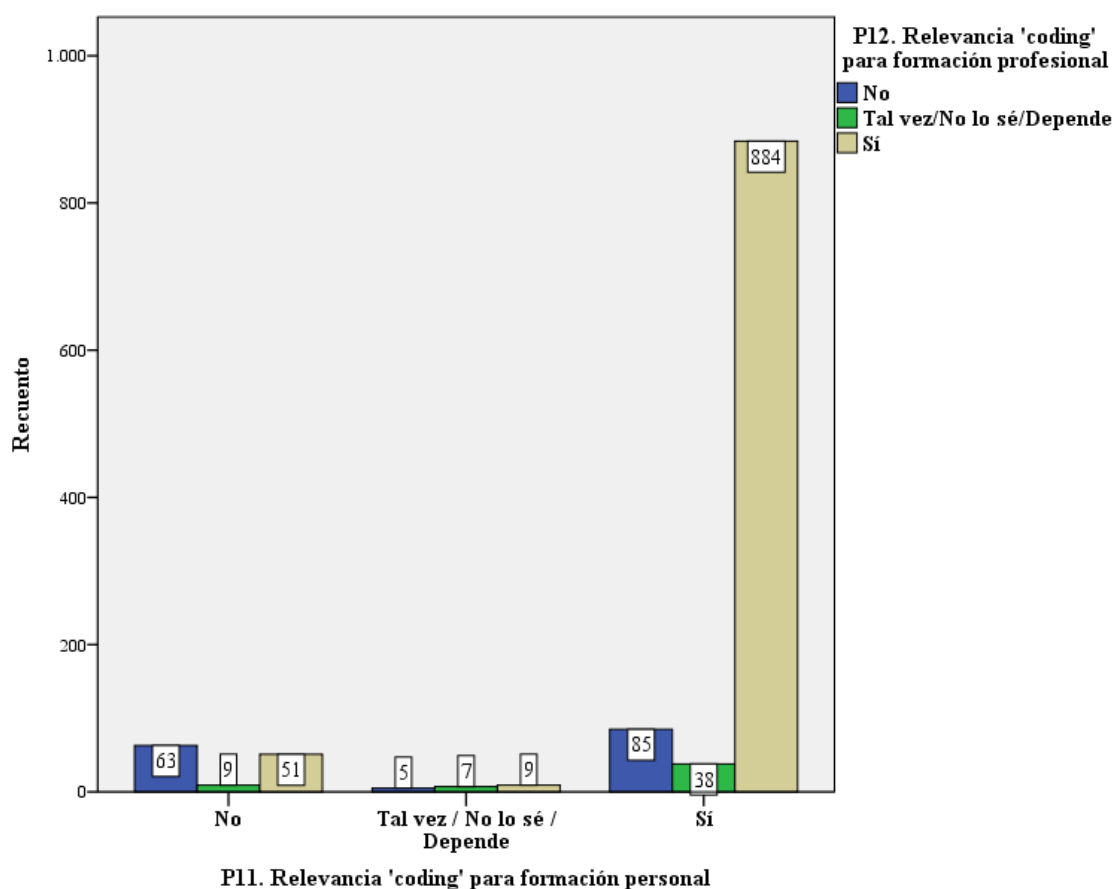


Figura 5.59. Frecuencias en la Pregunta 11 en función de la Pregunta 12

Así las cosas, con mayoría abrumadora del ‘sí’, y con fuerte redundancia en el sentido de las respuestas afirmativas a las preguntas 11 y 12; decidimos abordar un análisis cualitativo más profundo focalizado explícitamente sobre el contenido de las respuestas afirmativas a la Pregunta 11, con el objetivo de conocer en detalle las razones que dan los estudiantes para justificar la relevancia del ‘coding’ en su formación personal, como ciudadanos digitales del siglo XXI.

Para ello, se analizaron las 1.142 respuestas afirmativas válidas a la Pregunta 11. De dicho conjunto de respuestas, se encontraron 944 (82,6%) en las que el sujeto aportó alguna razón a su afirmación. Dichas razones fueron codificadas con el software de análisis cualitativo Atlas.ti (versión 7) en alguna de las siguientes 8 categorías³⁴⁶:

- **Razones utilitarias:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ en función de un beneficio futuro (‘el coding es importante para hacer...’; ‘para ser...’; ‘para ganar...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“sí, me parece algo serio porque así podré estudiar animación 3D”... “es importante porque nos ayuda si queremos ser programadores”... “I think that it’s important because in the future, people will have to know how to code for work or anything”... “Importantísima porque para cualquier trabajo hace falta un mínimo de programación”... “Sí, porque sirve para muchas cosas y para progresar en materia de informática: mejores videojuegos, etc.”... “Si creo que es bueno aprenderla porque es uno de los trabajos mejor pagados de todos actualmente”... “Sí, es importante, porque ahora para usar muchas cosas hace falta la programación informática”... “Sí, porque si alguien te pregunta sobre programación y te puede servir de mucho”.*
- **Razones sociológicas:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ en base a la premisa de que el mundo es (y será) tecnológico-digital (‘el coding es importante porque el mundo...’; ‘porque la tecnología...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“absolutely, because the new world we live in is based so much on the internet and social media, sooner or later the world will be completely controlled by the things we use most”... “ahora que la tecnología está tan avanzada, muchas cosas funcionan alrededor de ella así que es importante aprender a manejar los programas adecuadamente para el futuro”... “claro que sí; hoy en día todo está digitalizado”... “es importante porque hace falta saber programación, ya que en el futuro casi todo va a tratar alrededor de la tecnología”... “I do, since now a days computers are in our every day lives”... “la considero importante porque utilizamos la tecnología a diario”... “sí, porque en el futuro puede que todas las cosas sean de tecnología”... “sí, ahora nuestra vida cotidiana se encuentra rodeada de tecnología”... “sí, lo considero importante porque el día de mañana la tecnología será la base del mundo”.*
- **Razones expresivo-comunicativas:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ en base a que posibilita nuevas formas de expresión y comunicación (‘con el coding puedo crear...’; ‘puedo comunicarme...’; ‘puedo hackear...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“claramente, me parece un trabajo creativo*

³⁴⁶ Las categorías no se establecieron *a priori*, sino en paralelo al análisis de las respuestas; hasta llegar a un sistema comprensivo y comprensible.

para la gente que esté interesada y tenga planes de futuro en crear ideas que ayuden a las personas a través de la tecnología”... “está muy guay saber de código y poder crear cosas a partir de una página en blanco”... “it is important because with coding you can make games, applications, websites”... “sí, porque si quiero hacer un juego o una aplicación necesito saber programar”... “sí, para hackear juegos te puede servir y también para hackear las monedas de juegos, y crear aplicaciones para ti mismo”... “el saber programar te permite ser creativo con lo que utilizas diariamente y además poder crear nuevas aplicaciones para uso personal o para darlo a conocer”... “sí, es importante para nuestra creatividad”.

- **Razones escolares:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ por ser necesario para afrontar la vida escolar con éxito (‘el coding es importante en el colegio...’; ‘para los trabajos de clase...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“now a days in school we do almost everything in school (sic) with computers and we need to know how they work”... “bastante, hoy en día lo estudian en muchos coles”... “I think it is important to know computer programming, especially today because more and more of schoolwork and media is moving online”... “sí, porque para hacer trabajos importantes del colegio”... “sí, te ayuda en las notas y en matemáticas”.*
- **Razones emotivas:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ en base a los sentimientos positivos que genera (‘el coding me gusta...’; ‘disfruto...’; ‘me siento bien...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“I think it’s important for me, because firstly i enjoy programming”... “sí, porque me parece bastante divertido y aprendo mejor con el ordenador”... “sí, porque es educativo y divertido”... “sí, porque me encanta programar”... “sí, porque es muy divertido”... “sí, porque me gusta mucho”... “sí, porque te diviertes aprendiendo”.*
- **Razones de desarrollo:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ en la medida que promueve el desarrollo de aptitudes y habilidades (‘con el coding desarrollo la aptitud de...’; ‘aprendo a...’; ‘mejoro en...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“also programming teaches you how to solve problems which is very helpful for math”... “la considero importante porque te ayuda a mejorar tu lógica y te ayuda a progresar”... “la programación informática me ayudaría a resolver problemas”... “me parece muy importante para mi formación personal ya que ayuda a fomentar muchos campos poco trabajados como la perseverancia o ayuda a concentrarse; también fomenta el trabajo en equipo”... “sí, porque con esto aprendemos lógica”... “sí, para desarrollar todas las partes de nuestro cerebro”... “sí, para nuestro desarrollo”... “sí, porque es muy bueno para la mente”... “sí, porque la informática es una parte fundamental en nuestro desarrollo intelectual ya que la utilizamos diariamente”.*
- **Razones críticas:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ en la medida que nos permite tener una visión más crítica del mundo digital en el que vivimos, y de las estructuras de poder y control que en él operan (‘hay que programar en vez de ser programados...’; ‘saber coding es estar alfabetizado...’; ‘ser un ciudadano digital consciente y responsable...’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: *“I think it is important because when people use the computer, they think that it is all controlled*

by machines but in this way, by learning about computer programing, you learn the origins about computers and how it is actually a very real human invention/creation”... “la programación es algo muy importante para entender la sociedad en la que vivimos ya que el campo de la informática cada día ocupa más temas de conversación, es algo que en poco tiempo se ha hecho con el dominio del mercado y sobre todo de la cultura”... “por supuesto que sí, porque si no llegaremos a ser analfabetos del siglo XXI”... “sí, porque nos ayuda a ser más responsables con el iPad”... “sí, porque si no seríamos analfabetos informáticos”... “yes, because it helps us to understand what actually goes on behind all the scenes (...) it helps my personal training because I will understand things more thoroughly and see how everything works out! Thus, this can help me make decisions, and be wiser and have a greater knowledge”.

- **Razones genéricas:** respuestas que justifican la relevancia del aprendizaje del ‘coding’ de manera genérica o superficial (‘el coding es muy importante...’; ‘es muy necesario’). Algunos ejemplos de respuestas que fueron asignadas a esta categoría: “sí, porque es muy importante”... “lo considero muy importante para la vida”... “sí, pues me parece muy bien y lo recomiendo”

En cuanto a la frecuencia de aparición de cada tipo de razones, se ilustra en la siguiente Figura 5.60. Tal y como puede observarse, las razones más esgrimidas por los estudiantes para justificar la relevancia del aprendizaje del ‘coding’ fueron las razones sociológicas (212 respuestas asignadas a esta categoría), seguidas por las razones utilitarias (188), de desarrollo (180), genéricas (144), expresivo-comunicativas (76), emotivas (68), escolares (44), y, por último, las razones críticas (sólo 32 respuestas asignadas a esta categoría).

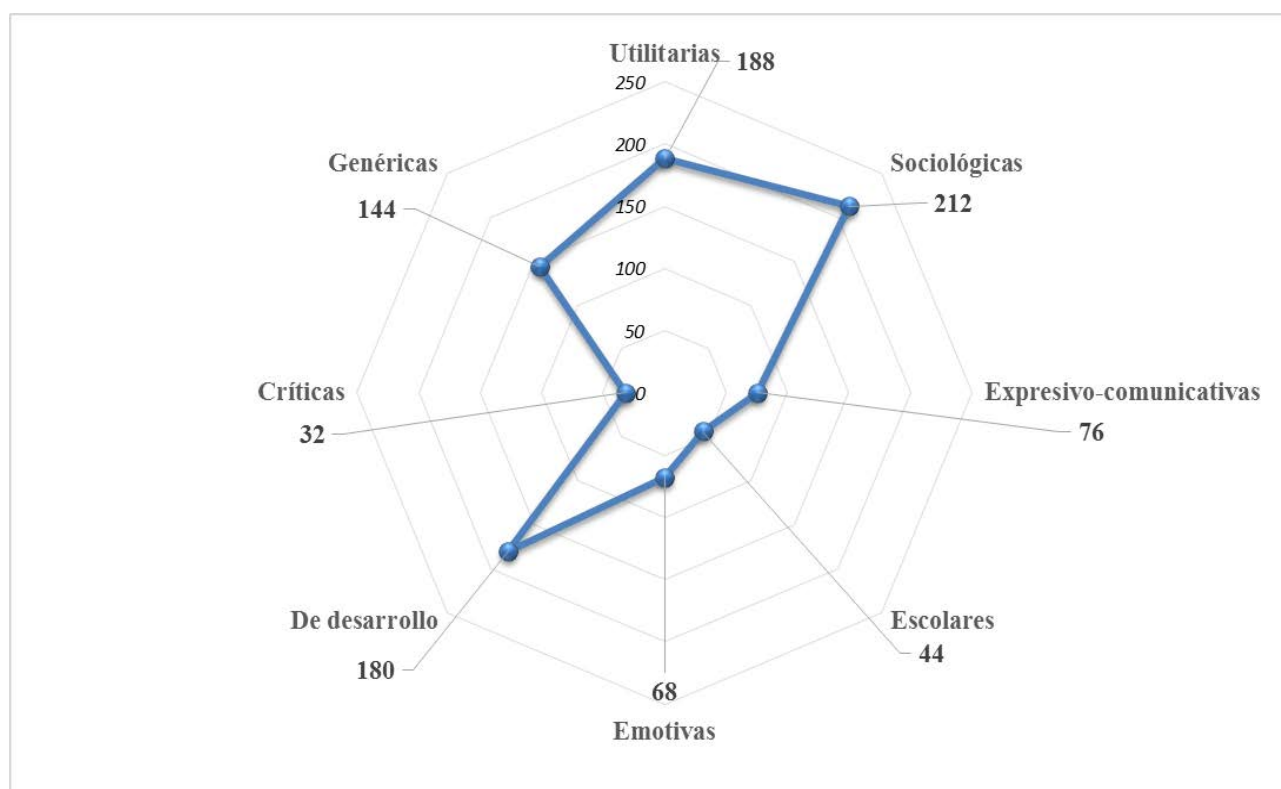


Figura 5.60. Gráfico radial con la frecuencia de aparición relativa a cada tipo de razones para aprender ‘coding’

5.5. Discusión y conclusiones parciales

Dentro del grupo de objetivos específicos de investigación O_{1*} podemos concluir que:

- Hay un fortísimo crecimiento del evento ‘La Hora del Código’ en España, multiplicándose por 5 el número de centros educativos españoles registrados en el mismo de la edición HoC-2013 a la edición HoC-2014.³⁴⁷
- El evento penetra inicialmente por el área Este de España (la mayoritaria en la HoC-2013), y se extiende posteriormente al área Noroeste y Centro (que suben en la HoC-2014). El área Sur-Canarias es minoritario en la HoC-2013, pero no despegaba en la HoC-2014. En cualquier caso, estas variaciones por área geográfica no fueron estadísticamente significativas y hay que tomarlas con prudencia.
- Destacamos que el despegue del evento en el área Noroeste en la HoC-2014 puede atribuirse fundamentalmente a Castilla y León (que pasó de 0 centros en la HoC-2013 a 6 centros invitados en la HoC-2014), y a Galicia (que pasó de 0 a 7 centros invitados).
- Por otra parte, el impacto del evento se extiende desde los centros privados/concertados (que suman mayoría en la HoC-2013), hacia los centros públicos (que ya son claramente mayoritarios en la HoC-2014). Esta variación por titularidad está en el límite de la significación estadística.
- Finalmente, puede afirmarse que el evento penetra inicialmente a través de los centros ‘Multietapa’ (privados o concertados), que son mayoría en la HoC-2013, y se extiende posteriormente hacia los CEIP (Colegios Públicos de Educación Infantil y/o Primaria), que son mayoría en la HoC-2014. Los IES (Institutos Públicos de Educación Secundaria, Bachillerato y/o FP) ya están presentes en la HoC-2013 y se mantienen en la HoC-2014. Esta variación por tipología sí fue significativamente estadística.

Así, globalmente, el evento ‘La Hora del Código’ irrumpe en España en el año 2013 por la zona Este de la mano de colegios privados y concertados Multietapa (algunos de ellos son colegios internacionales y bilingües: tiene sentido que precisamente fueran este tipo de centros los primeros en hacerse eco del evento, dado que el movimiento de aprendizaje del código proviene de Estados Unidos y los centros privados suelen tener menos resistencias a la innovación). En 2014, ‘La Hora del Código’ se extiende hacia otros territorios (especialmente el área Noroeste y Centro), y se incorpora al mismo masivamente el sector público (especialmente de la mano del espectacular aumento de los CEIP; los IES ya estaban presentes en 2013). En la Figura 5.61 se representa aún más sintéticamente esta evolución.

³⁴⁷ En las semanas previas al depósito de esta tesis doctoral, hemos tenido noticia del número de centros registrados en España en la edición HoC-2015, que ha ascendido a 1.581 centros. Ello supone que el número de centros registrados en nuestro país se ha multiplicado por más de 15 veces en tan sólo 2 años (de diciembre de 2013 a diciembre de 2015)

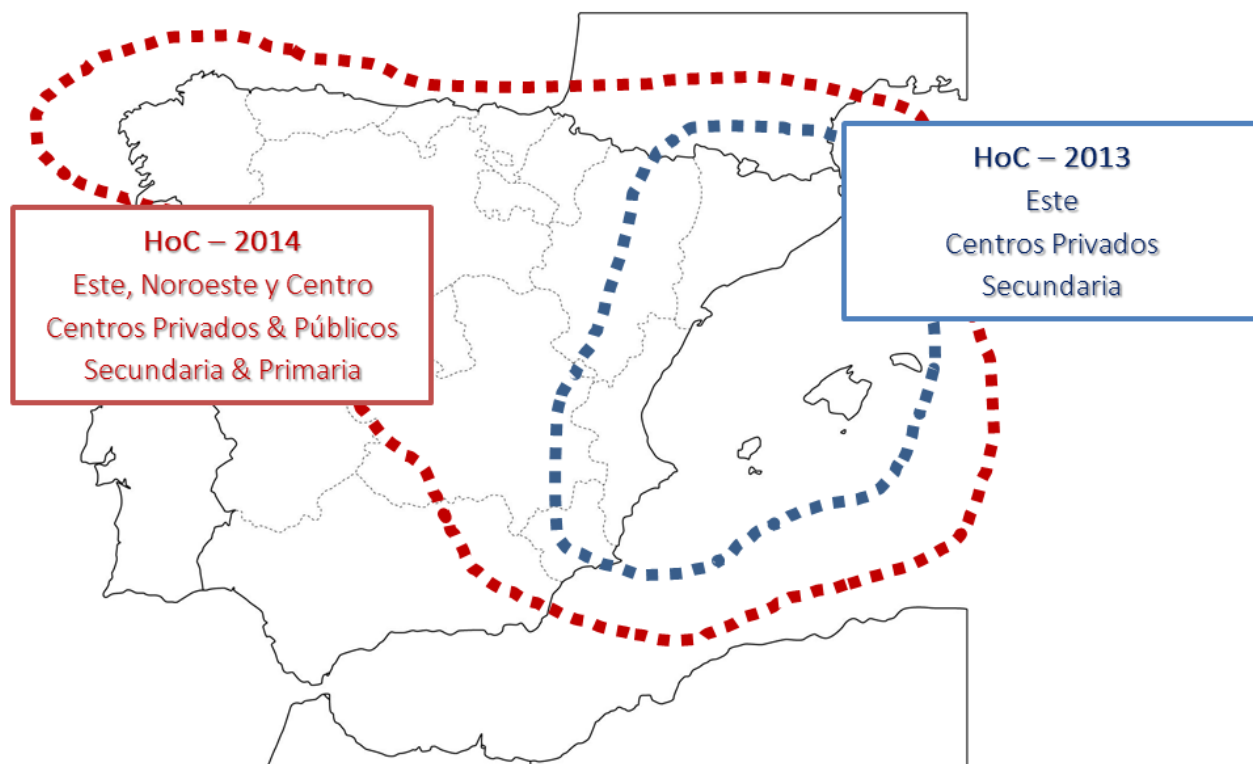


Figura 5.61. Evolución de ‘La Hora del Código’ a lo largo de las ediciones 2013 y 2014

Dentro del grupo de objetivos específicos de investigación $O_{2Pa} \dots O_{2Pc}$, en el marco de la hipótesis directiva de *adecuación*, podemos concluir que:

- Los profesores consideran que la programación informática o ‘coding’ desarrolla un amplio abanico de aptitudes y habilidades. Aunque destacan las relacionadas con el razonamiento lógico y la resolución de problemas, se nombran muchas otras como aptitudes espaciales, atencionales, de motivación e iniciativa, creatividad o trabajo en equipo. Adicionalmente, se señala su potencial para la atención a la diversidad y, aunque prácticamente no se hace mención literal del término ‘pensamiento computacional’, sí se indica que la programación informática debería utilizarse no como fin en sí mismo sino como medio para desarrollar una amplia gama de aptitudes y habilidades, transferibles a diversos campos.
- Hay un consenso generalizado entre los profesores sobre que el aprendizaje del ‘coding’ es adecuado para todo tipo de alumnos, independientemente de su edad, género o capacidad. Se concibe, pues, como un aprendizaje transversal que debe promoverse con las adaptaciones necesarias para cada colectivo, pero sin dejar fuera a ninguno de ellos.
- Moderadamente, los profesores esperan correlación entre el rendimiento de sus alumnos en el área de matemáticas y su nivel de ejecución en tareas de ‘coding’; si bien consideran que el ‘coding’ tiene algunos aspectos visuales propios, y que puede ser motivador para todos los estudiantes independientemente de su rendimiento en matemáticas u otras áreas.

Dentro del grupo de objetivos de investigación $O_{2Pd} \dots O_{2Pg}$, en el marco de la hipótesis directiva de *viabilidad*, podemos concluir que:

- Los profesores son conocedores de un amplio abanico de recursos y plataformas-tutoriales ‘on-line’ para el aprendizaje del ‘coding’. Se refieren más de una veintena de recursos al respecto. Globalmente, y utilizando la terminología que empleamos en la taxonomía del [epígrafe 2.4.1](#), los profesores señalan para Educación Infantil y comienzos de Educación Primaria recursos de aprendizaje que utilizan ‘lenguajes visuales por flechas’; para finales de Educación Primaria y comienzos de Educación Secundaria se señalan recursos basados en ‘lenguajes visuales por bloques’; y para finales de Educación Secundaria y Bachillerato se indican recursos basados en ‘lenguajes textuales’.
- El tutorial más utilizado durante la HoC es el propuesto por Code.org basado en los personajes de ‘Angry Birds’, por su atractivo visual y sencillez, por el efecto motivador de los personajes en los estudiantes, y por estar bien diseñado para un tiempo de aprendizaje de 1 hora.
- En conjunto, los profesores declaran que con una actividad de 1 hora es posible introducir a los estudiantes en el ‘coding’, despertando su curiosidad para proseguir con un aprendizaje más profundo. Además, declaran que este tipo de actividades y eventos son una forma de mostrar a la comunidad educativa la accesibilidad de la programación informática para todo tipo de alumnos, desmitificando su dificultad extrema; apoyándose para ello en los nuevos recursos y lenguajes visuales, que bajan las ‘barreras de entrada’ a la programación y que permiten generar productos en breve tiempo. Estas conclusiones van en la línea de lo señalado por Lye y Koh en su reciente revisión (2014), acerca de cómo la proliferación y disponibilidad de los nuevos lenguajes visuales está revitalizando el interés en las escuelas sobre la programación informática.
- Los profesores informan de un excelente caldo de cultivo en la comunidad educativa para celebrar el evento de ‘La Hora del Código’: muchos apoyos y casi ninguna resistencia. Parece que hay una asunción colectiva de padres, madres, estudiantes, equipos directivos y profesorado, de la importancia de la programación como habilidad digital clave en el siglo XXI.

Dentro del grupo de objetivos de investigación $O_{2Ph} \dots O_{2Pi}$, en el marco de la hipótesis directiva de *relevancia*, podemos concluir que:

- De manera casi unánime, los profesores manifiestan que el ‘coding’ es una habilidad relevante para el futuro de sus estudiantes como ciudadanos digitales. Las principales razones que se esgrimen tienen que ver con la omnipresencia de los objetos digitales en la sociedad actual y futura, y con el desarrollo de habilidades cognitivas superiores de resolución de problemas (incluidas menciones expresas al ‘pensamiento computacional’) al programar. En menor medida, aparecen razones de corte creativo-expresivo o sociocríticas.
- También con rotundidad, la gran mayoría de profesores expresan su opinión de que el ‘coding’ es una habilidad relevante para el futuro de sus estudiantes como trabajadores digitales. Ya sea por la creciente demanda explícita de programadores, ya sea por la utilidad del ‘coding’ para afrontar de manera más eficaz un amplio abanico de profesiones en la sociedad digital.

Dentro del grupo de objetivos de investigación $O_{2Pj} \dots O_{2Pñ}$ podemos concluir que:

- Las actividades inicialmente planificadas en los centros para ‘La Hora del Código’ abarcan un amplio rango de edades (desde Educación Infantil hasta Formación Profesional), apoyándose en un variado abanico de recursos y tutoriales en línea. En general, los eventos organizados tienen un carácter masivo, involucrando a todo el centro, lo cual refleja la facilidad con que se pueden ‘escalar’ (= ampliar el factor de escala) dichos recursos de aprendizaje del ‘coding’. Son eventos que se difunden a toda la comunidad educativa. Habitualmente se celebran en el aula de informática y a través de equipos PC’s fijos, si bien ya aparece el uso de algunos dispositivos portátiles y móviles. Son eventos de marcado carácter lúdico, y sin evaluación formal. La metodología es activa, ‘*learning by doing*’ con el apoyo de los tutoriales ‘*on-line*’ autoguiados; pero promoviendo que los estudiantes se ayuden mutuamente (metodología ‘*ask 3, then me*’).
- Las actividades finalmente desarrolladas siguen en su gran mayoría la planificación inicial, a excepción de casos puntuales por problemas de temporalización u organización escolar. Es destacable que, cuando es necesario, se modifica sobre la marcha el dispositivo y/o tutorial de aprendizaje inicialmente escogido por el profesor (p.e. por resultar excesivamente fácil o difícil), lo que da idea de la flexibilidad y adaptabilidad de estos sistemas al nivel de ‘coding’ de cada estudiante o grupo de estudiantes. No se realizan evaluaciones formales, lo cual sugiere cierto vacío en este terreno que trataremos de cubrir en el Capítulo 6 con el diseño y validación de nuestro ‘Test de Pensamiento Computacional’ (TPC)
- Con respecto a los aspectos más positivos y más negativos derivados de la celebración del evento, según los profesores, se sintetizan en la siguiente Figura 5.62.

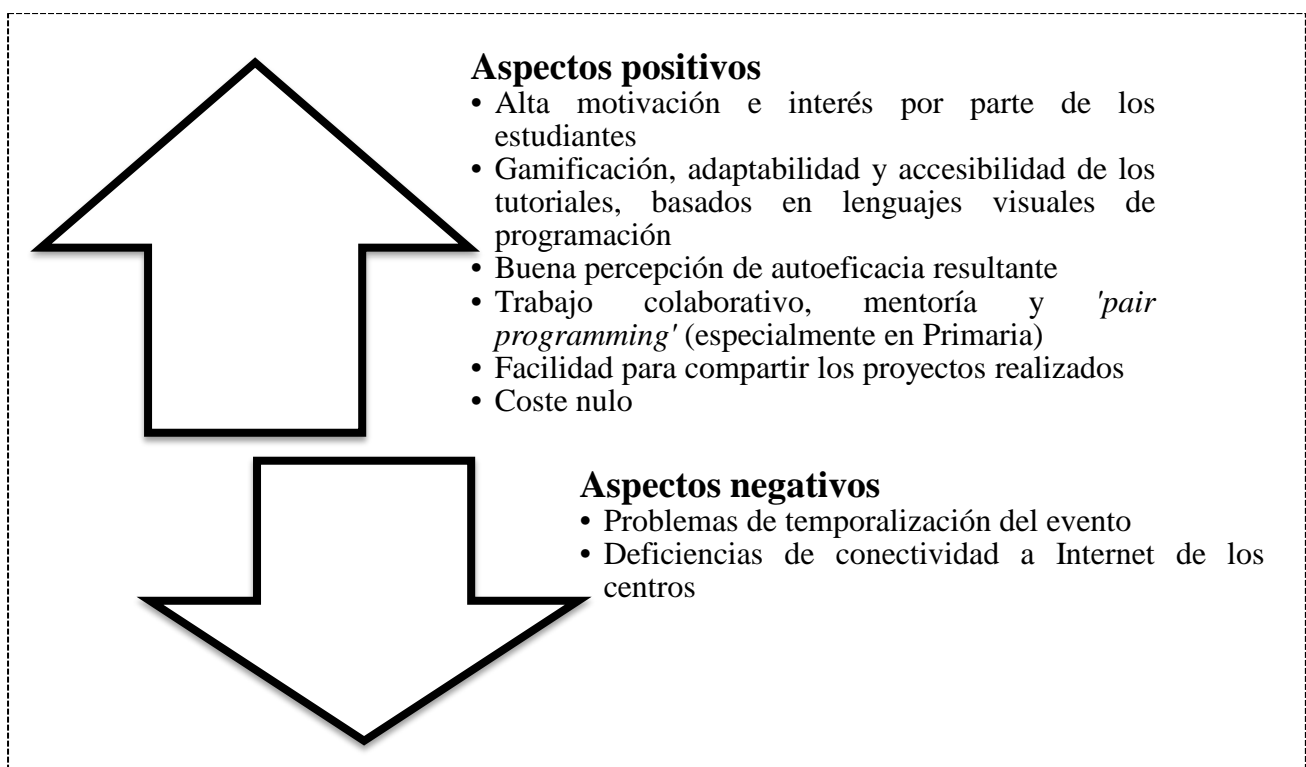


Figura 5.62. Aspectos más positivos y más negativos reportados por los profesores en el cuestionario posterior a ‘La Hora del Código’

- El anecdotario o casuística excepcional del evento, referida por los profesores, gira alrededor de: estudiantes que demostraron un ‘talento especial’ a través de un desempeño excelente en las tareas de ‘coding’, incluso sin experiencia previa; estudiantes (o grupos) habitualmente disruptivos, inquietos o con problemas de atención, que se engancharon de forma sostenida a las tareas de programación propuestas; estudiantes con desfase curricular en otras áreas que mostraron un rendimiento inesperadamente alto en el ‘coding’; chicas con prejuicios negativos alrededor de su aptitud para programar, que se vieron confrontadas al comprobar su competencia con las tareas de ‘coding’ propuestas; o el excelente funcionamiento de la metodología ‘pair programming’.
- Fundamentalmente, se detectan iniciativas de proseguir con currículos más sistemáticos, prolongados y avanzados de aprendizaje del ‘coding’ a final de la etapa de Primaria y Secundaria, con el curso trimestral ‘K-8 Intro to Computer Science’ (Code.org, 2015a). Y también intenciones de profundización trimestral con Scratch en Secundaria y con AppInventor en Secundaria y Bachillerato. Ello nos pone sobre la pista, y en contacto con los centros adecuados, para llevar a cabo la evaluación de programas que se describe en el Capítulo 7 de esta tesis doctoral.

Dentro del grupo de objetivos $O_{2Ea} \dots O_{2Ec}$ podemos concluir que:

- Algo más de la mitad de los estudiantes encuestados (53,0%) ha oído hablar de la programación informática o ‘coding’ con anterioridad al evento de ‘La Hora del Código’. Ese porcentaje es significativamente mayor según se avanza en las etapas del sistema educativo; y significativamente mayor en chicos (59,0%) que en chicas (46,7%). La mayor brecha de género a favor de los chicos (+13,3%) en su exposición al término ‘programación informática’ o ‘coding’ se da en Educación Secundaria. Y, adicionalmente, hay un porcentaje significativamente mayor de estudiantes que han oído hablar del ‘coding’ en los centros privados-concertados ‘multietapa’ (incluyen estudiantes de Primaria y Secundaria) frente a los institutos o colegios públicos.
- Algo menos de la mitad de los estudiantes encuestados (43,0%) declara haber programado con anterioridad al evento. Este porcentaje es significativamente mayor según se avanza en las etapas del sistema educativo; y significativamente mayor en chicos (49,2%) que en chicas (37,0%). La mayor brecha de género a favor de los chicos (+24,4%) en su experiencia previa en programación se da en Bachillerato. Y, adicionalmente, hay un porcentaje significativamente mayor de estudiantes que ha tenido experiencia previa en ‘coding’ en los centros privados-concertados ‘multietapa’ (incluyen estudiantes de Primaria y Secundaria) frente a los institutos o colegios públicos.
- Analizando conjuntamente los dos puntos anteriores (Figura 5.63), vemos que los chicos mantienen una diferencia con respecto a las chicas aproximadamente constante $\approx +10\%$ a lo largo de todas las etapas en lo relativo a haber oído hablar del ‘coding’. Sin embargo, mientras que los chicos van dando progresivamente el paso de ‘oír hablar’ al de ‘tener experiencia’ con la programación; las chicas dejan de dar dicho paso a partir de la Secundaria. Dicho de otra manera, la brecha de género tiene un doble componente: no es sólo que las chicas estén menos expuestas a la idea del ‘coding’, sino que además se deciden menos a probarlo.

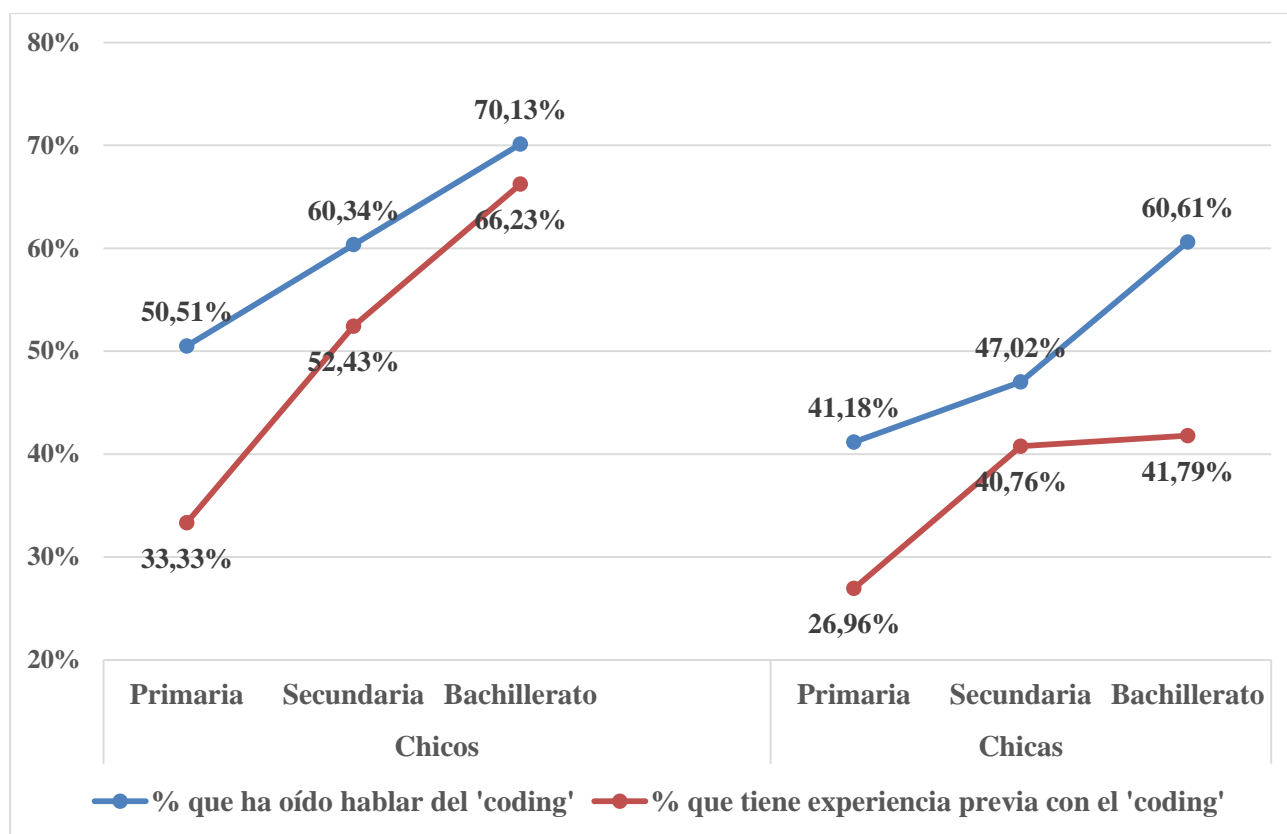


Figura 5.63. Análisis combinado de la brecha de género (se ha omitido aquí la etapa de FP por su reducida muestra)

- Los lenguajes de programación con los que un mayor porcentaje de estudiantes declara haber tenido experiencia son (de más a menos y sólo reflejando aquellos con una penetración >2% sobre el total de la muestra): HTML (14,02%), Scratch (12,03%), JavaScript (9,63%), CSS (4,45%), Python (3,61%), Lightbot (2,77%) y Blockly (2,17%). Se observa una clara trayectoria evolutiva y de género en el uso de los lenguajes: así, los lenguajes visuales encuentran su mayor uso en Primaria (Lightbot, visual por flechas) y en Secundaria (Scratch y Blockly, visuales por bloques), con un índice de penetración similar entre sexos; los 'pseudolenguajes' de etiquetas y estilo (HTML y CSS) tienen su mayor uso en el Bachillerato, con un factor de género $\approx 1,5$ veces superior en chicos; y los lenguajes textuales puros (JavaScript y Python) encuentran su mayor uso en la Formación Profesional, con un factor de género $\approx 2,5-3,0$ veces superior en los chicos. Así, la progresiva brecha de género parece que corre en paralelo a la transición de un tipo de lenguajes a otros.

Con respecto al objetivo **O_{2Ed}**, en el marco de la hipótesis directiva de **adecuación**, podemos concluir que:

- En términos generales, los estudiantes consideran que la actividad introductoria de 'coding' les ha servido para desarrollar en un nivel medio-alto (medianas de 4 sobre 5 en la escala Likert): 'capacidad de atención', 'pensamiento lógico', 'creatividad', 'resolución de problemas' y 'persistencia-perseverancia'. Y en un nivel medio (mediana de 3 sobre 5 en la escala Likert) el 'trabajo en equipo'.

- Precisamente la aptitud-habilidad ‘trabajo en equipo’ presenta una distribución de frecuencias distinta al resto; es una distribución casi bimodal en los extremos de la escala Likert: así pues parece una variable de ‘todo o nada’, que presumiblemente sería dependiente de valores contextuales (en función de la metodología de aprendizaje del ‘coding’ seguida por cada profesor, se percibe ‘nada’ o ‘mucho’ trabajo en equipo). El resto de variables presentan distribuciones con asimetría negativa, con la mayoría de las respuestas acumuladas en la parte superior de la escala; parecen, presumiblemente, variables más independientes de valores contextuales, y vinculadas a la tarea de programación en sí.
- No se encuentran diferencias por sexo en la percepción de cómo la actividad introductoria de ‘coding’ ha contribuido a desarrollar las distintas aptitudes-habilidades. Sí se encuentran diferencias estadísticamente significativas en función de la etapa educativa, a favor de Primaria frente a Secundaria y Bachillerato, especialmente en la ‘capacidad de atención’ y el ‘trabajo en equipo’ (en ambas igualmente aparecen diferencias significativas en función de tener o no experiencia previa en ‘coding’, favorables a estos últimos). Parece, pues, que los estudiantes más pequeños (en mayor medida sin experiencia en ‘coding’) perciben más intensamente la relación del ‘coding’ con el desarrollo de la capacidad de atención; a medida que son más mayores (y va habiendo más estudiantes con experiencia previa en ‘coding’) resulta menos saliente el aspecto atencional, pues posiblemente haya partes de la tarea que ya se han automatizado. Adicionalmente, y en consistencia con los cuestionarios de profesores, parece que el ‘trabajo en equipo’ fue promovido especialmente en las actividades de ‘coding’ en Primaria, frente a Secundaria y Bachillerato.
- La percepción que tienen los estudiantes sobre en qué medida la actividad de ‘coding’ contribuye a desarrollar sus aptitudes y habilidades, se organiza alrededor de 3 factores relativamente independientes: 1) el principal, que tiene que ver con ‘el acto de programar en sí’, intrínseco a cualquier tarea de ‘coding’, y que aglutina dos factores de segundo orden: uno cognitivo (‘pensamiento lógico’ y ‘resolución de problemas’) y otro motivacional-atencional (‘persistencia-perseverancia’ y ‘capacidad de atención’); 2) factor de menor peso, que tiene que ver con la carga creativa del ‘entorno-tarea de programación’ utilizado: coherentemente con una de las dimensiones de la taxonomía planteada en el [epígrafe 2.4.1](#) de la parte teórica, hay entornos de ‘coding’ poco creativos consistentes en la resolución de puzzles o problemas cerrados de programación (‘close-ended’), frente a entornos altamente creativos donde se programan proyectos abiertos ‘desde cero’ (‘open-ended’) o remezclando creativamente el código de otros; y 3), el factor que tiene que ver con la ‘metodología de aprendizaje de la programación’ utilizada por el docente en el aula, en especial en lo relativo a si el ‘coding’ se realiza de manera colaborativa (p.e. mediante ‘pair programming’) o en solitario.

Con respecto al grupo de objetivos $O_{2Ee} \dots O_{2Ej}$, en el marco de la hipótesis de *viabilidad*, podemos concluir que:

- Tras finalizar la actividad de ‘coding’, la percepción de autoeficacia general es moderadamente alta (mediana de 4 sobre 5). Destaca especialmente que sólo el 6,9% de los estudiantes declara que el ‘coding’ se les da ‘mal’ o ‘muy mal’ tras la experiencia; y se

acumula un 70,2% que declara que se les da ‘bien’ o ‘muy bien’. En este sentido, parece que el evento ‘La Hora del Código’ consigue su objetivo, tal y como vimos en el [sub-epígrafe 2.2.3.1](#) de la parte teórica, de desmitificar la dificultad del aprendizaje del ‘coding’. Con su oferta de tutoriales visuales, intuitivos y adaptables, parece conseguir que la mayoría de estudiantes acaben la actividad con una alta percepción de competencia; convirtiendo así el ‘coding’ en un aprendizaje potencialmente accesible para el grueso de la población escolar (y no sólo para los más dotados).

- 2 de cada 3 estudiantes afirma que pudo completar el 100% de la actividad propuesta en su centro para ‘La Hora del Código’. Es un dato que globalmente nos indica la viabilidad de iniciar y finalizar una actividad introductoria al ‘coding’ en una sola sesión de clase. Aun así es un dato con margen de mejora: 1 de cada 3 estudiantes afirma que no pudo finalizar el 100% de la actividad, algo que ya se anticipaba en las respuestas de los profesores a su cuestionario (*post*) en donde se reportaron algunos problemas de temporalización.
- El hecho de que el estudiante consiga acabar, o no acabar, la actividad de una hora propuesta tiene un efecto moderado sobre la percepción de autoeficacia ulterior (en el caso de acabarla, la percepción de autoeficacia es luego significativamente mayor). Es decir, hay evidencia empírica que respalda la importancia de planificar actividades de aprendizaje que puedan abordarse y finalizarse en el tiempo previsto; de cara a una adecuada sensación de competencia por parte del estudiante, que presumiblemente le proporciona una buena base motivacional para aprendizajes posteriores.
- Existen diferencias significativas en función del sexo, tanto en el porcentaje de sujetos que consiguió finalizar la actividad HoC como en la percepción de ‘autoeficacia’ subsiguiente, ambas a favor de los chicos. Sin embargo, dichas diferencias desaparecen cuando se controla la experiencia previa en programación.
- También existen diferencias significativas en función de la etapa educativa, tanto en el porcentaje de sujetos que consiguió finalizar la actividad HoC como en la percepción de ‘autoeficacia’ subsiguiente. Educación Secundaria es la etapa en la cual ambas variables presentan sus valores más bajos; y, por tanto, en donde el ajuste global de la actividad planteada a los estudiantes fue más deficiente. Triangulando con la información aportada por los profesores, es posible que ello tenga que ver con que en esos cursos se produce la transición o salto desde los lenguajes visuales por bloques a los lenguajes textuales (Weintrop & Wilensky, 2015a, 2015b); ocasionando ciertos desajustes en el nivel de dificultad que las actividades plantean a los estudiantes.
- Los estudiantes que consiguen finalizar la actividad HoC, y que en consecuencia tienden a adquirir una percepción de autoeficacia mayor en ‘coding’, tienden igualmente a valorar en mayor medida la contribución del ‘coding’ al desarrollo de sus distintas aptitudes y habilidades. Por el contrario, los estudiantes que no consiguen finalizar la actividad, con menor percepción de autoeficacia, desvalorizan el papel del ‘coding’ para su desarrollo. Esta relación podría ser explicada en términos de disonancia cognitiva (Festinger, 1957), según la cual tendemos a generar sistemas de creencias y percepciones que encajen entre sí, que

guarden cierta coherencia interna: “*si acabo la actividad HoC, me siento más competente en ‘coding’, y valoro más sus efectos sobre mis aptitudes*” (muestra de coherencia cognitiva); “*si soy incapaz de finalizar la actividad HoC, percibo que se me da mal el ‘coding’, y lo desvalorizo*” (para evitar disonancia cognitiva).

- El rendimiento académico informado es significativamente superior en las chicas frente a los chicos para las asignaturas de Lengua, Música y Plástica; y se iguala entre ambos sexos para Matemáticas y Tecnología/Informática. El rendimiento académico informado sufre un descenso significativo desde Educación Primaria a Educación Secundaria, con una posterior recuperación en Bachillerato debida fundamentalmente al colectivo femenino.
- Más relevante para nuestra investigación, encontramos una correlación significativa entre tener experiencia previa en ‘coding’ y el rendimiento académico informado; especialmente en las asignaturas de Tecnología/Informática y Matemáticas, tal y como era de esperar. Además, dicha correlación es más fuerte en el colectivo masculino. Así, el tener contacto con la programación informática podría estar funcionando como catalizador del buen rendimiento académico, especialmente en las asignaturas más afines y con un efecto acelerador especialmente intenso entre los chicos (las chicas tienden a rendir alto en cualquier caso).
- En la misma línea, encontramos una correlación significativa entre la percepción de autoeficacia en ‘coding’ tras realizar la actividad y el rendimiento académico informado; otra vez especialmente intensa en las asignaturas de Tecnología/Informática y Matemáticas. Pero esta relación es más acusada en el colectivo masculino que en el femenino: parece que los chicos establecen una relación más directa entre su rendimiento típico (hecho ‘objetivo’) y su percepción de autoeficacia (hecho ‘subjetivo’); no tanto así las chicas, en donde es más común encontrar casos en los que, a pesar de un alto rendimiento, se declara una percepción de autoeficacia baja o moderada (Figura 5.64). Este hallazgo es consistente con la investigación sobre diferencias de género en ‘pesimismo defensivo’ (ver p.e., [Rodríguez, Cabanach, Valle, Núñez, & González-Pineda, 2004](#); [Suárez Riveiro & Fernández Suárez, 2005](#)), que repetidamente muestra que las chicas son más proclives a declararse poco eficaces en una tarea aunque tengan evidencia objetiva de su rendimiento en la misma.
- Igualmente, encontramos una correlación significativa entre el hecho de haber conseguido acabar, o no, la actividad HoC y el rendimiento académico informado; y otra vez especialmente intensa en las asignaturas de Tecnología/Informática y Matemática. En este caso la relación es similar en ambos sexos, lo cual es explicable dado que ahora se correlacionan dos hechos objetivos; mientras que en el punto anterior se correlacionaba una percepción subjetiva con un hecho objetivo, dando lugar a efectos diferenciales por sexo.

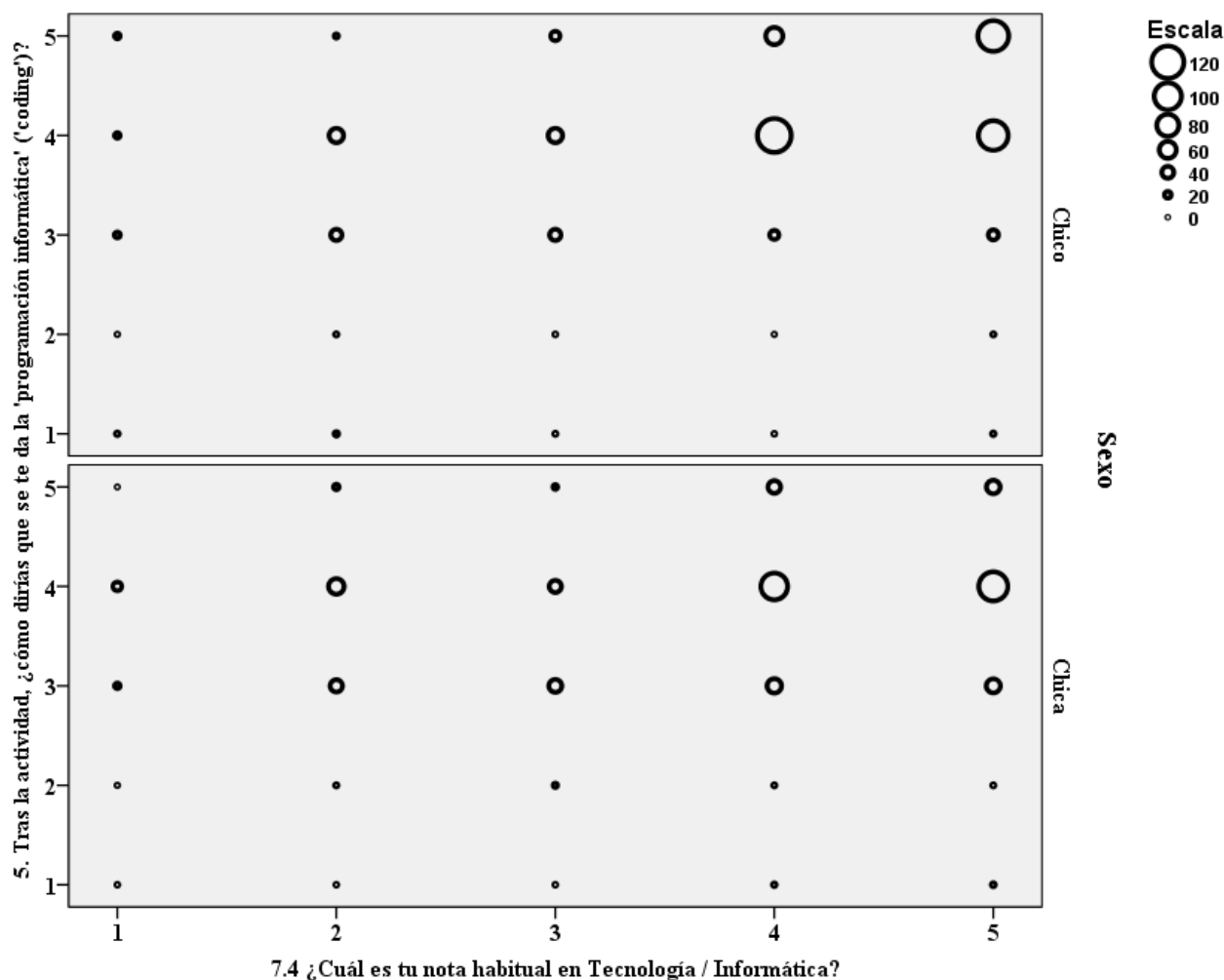


Figura 5.64. Diagrama de dispersión entre el rendimiento informado en Tecnología/Informática y la percepción de autoeficacia en 'coding', diferenciado por sexo. Los chicos establecen una relación más directa entre ambas variables, mientras que hay chicas con alto rendimiento que se sitúan en niveles sólo medios de autoeficacia.

- Globalmente, el grado de motivación para proseguir el aprendizaje del 'coding' tras el evento, manifestado por los estudiantes, es bastante alto (mediana y moda de 4, en una escala de 1 a 5). Tan sólo un 12,6% se declara 'nada motivado' o 'poco motivado' para continuar aprendiendo 'coding'; el 27,2% se declara 'algo motivado'; y un 60,2% dice estar 'bastante motivado' o 'muy motivado' para seguir con el aprendizaje del 'coding' tras su experiencia en el evento de 'La Hora del Código'. Además, 8 de cada 10 estudiantes declara que le gustaría seguir aprendiendo 'coding' a través de entornos 'on-line' (plataformas y tutoriales en Internet que permiten al estudiante avanzar a su propio ritmo). Así pues, diríamos que el evento 'La Hora del Código' logra su objetivo de enganchar a una mayoría de estudiantes para que, a partir de la actividad introductoria realizada, se interesen en proseguir y profundizar en el aprendizaje del 'coding'.
- Tanto el grado de motivación para proseguir al aprendizaje del 'coding', como la determinación para hacerlo 'on-line', es significativamente mayor en chicos y en chicas; y significativamente mayor en Primaria que en Secundaria (en donde se produce un brusco descenso, atribuible fundamentalmente al colectivo femenino). Además, el hecho de haber o

no haber conseguido completar la actividad HoC y la percepción de autoeficacia subsiguiente, tienen un efecto significativo en estas variables motivacionales: a mayor autoeficacia, de manera casi lineal, aparece mayor motivación para proseguir el aprendizaje del 'coding'; en el caso específico del aprendizaje a través de plataformas 'on-line' se observa un techo atribuible a un pequeño (pero resistente) grupo de estudiantes que, aún con buena autoeficacia y buena motivación para el 'coding', rechazan el sistema en línea.

- Así pues, se detecta una efecto motivacional 'en cascada' desde el hecho de conseguir completar la actividad HoC, tener una buena percepción de autoeficacia subsiguiente en tareas de 'coding', manifestar consistentemente una buena motivación para continuar con su aprendizaje y la determinación para hacerlo a través de plataformas-tutoriales 'on-line'. Parece que 'La Hora del Código' consigue introducir a una mayoría de estudiantes en este círculo virtuoso (Figura 5.65): la mayor densidad de sujetos se concentra en el área del cubo correspondiente a "alta autoeficacia * alta motivación * determinación a aprender 'coding' on-line"; a partir de su oferta de recursos de aprendizaje atractivos, visuales y muy bien ajustados en su grado de dificultad.

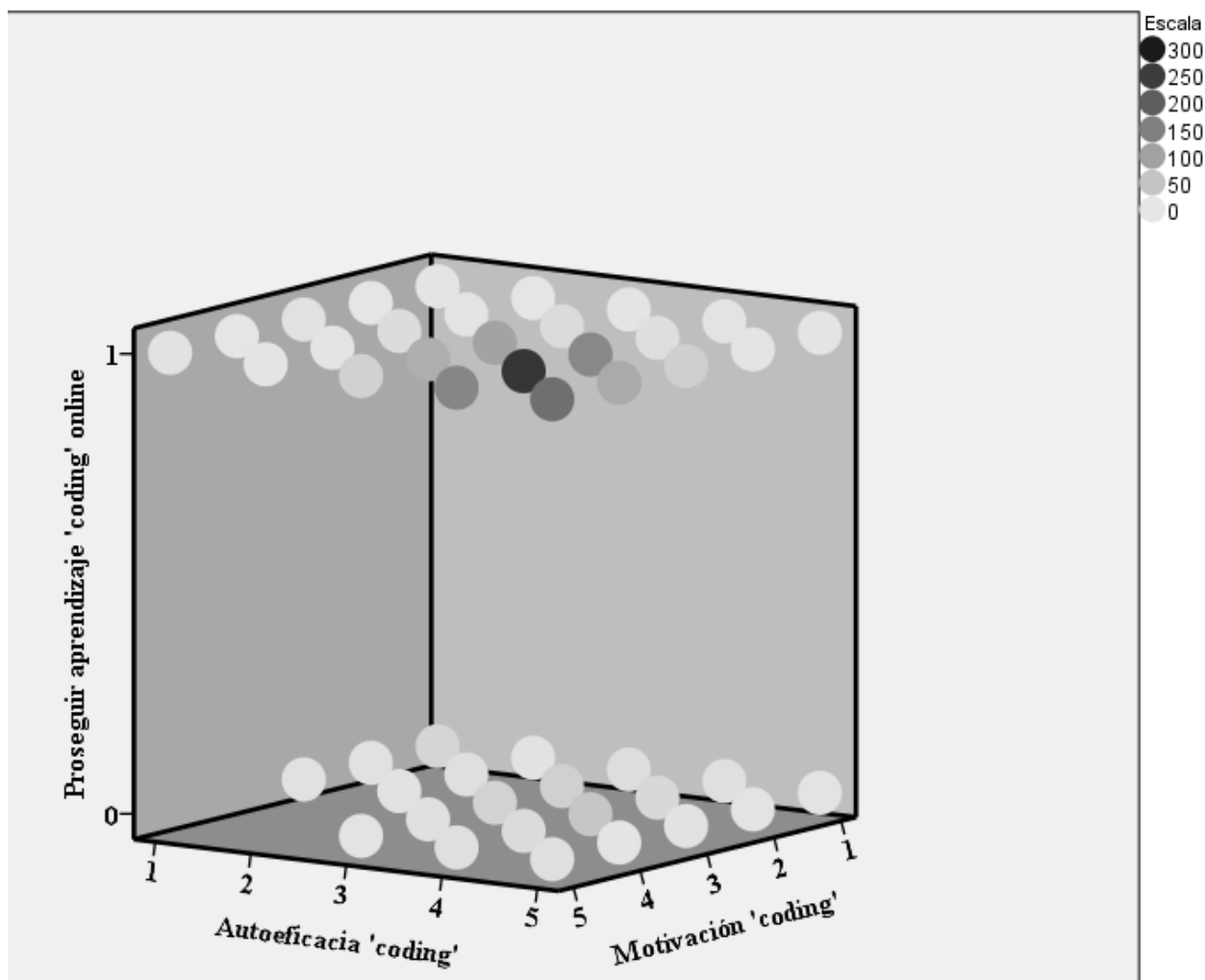


Figura 5.65. Diagrama de dispersión tridimensional: la mayor densidad de sujetos se concentra en el área de "alta autoeficacia * alta motivación * sí a proseguir aprendizaje del 'coding' on-line"

- En esta ‘cascada motivacional virtuosa’, unos pocos sujetos van quedándose por el camino (p.e. sujetos que, aunque hayan finalizado la actividad con una percepción de autoeficacia media-alta, luego declaran baja motivación para proseguir con el aprendizaje del ‘coding’). Lo más relevante es que este abandono de la ‘cascada motivacional virtuosa’ es más acusado en chicas que en chicos, se manifiesta especialmente en Secundaria, y particularmente en la determinación para aprender ‘coding on-line’ (Tabla 5.97, y Figuras 5.66, 5.67, y 5.68). Estos resultados son consistentes con hallazgos previos que señalan la menor propensión de las chicas hacia los entornos de aprendizaje ‘on-line’ (Yang *et al.*, 2011)

Tabla 5.97. Evolución de la ‘cascada motivacional virtuosa’, en función del sexo

	<i>P5. Autoeficacia ‘coding’</i> (% acumulado medio-alto ³⁴⁸)	<i>P8. Motivación proseguir aprendizaje ‘coding’</i> (% acumulado medio-alto)	<i>P9. Determinación aprender ‘coding’ on-line</i> (% afirmativo)
<i>Chicos</i>	93,72%	88,32%	83,38%
<i>Chicas</i>	92,85%	86,27%	77,38%
<i>Total</i>	93,13%	87,41%	80,36%

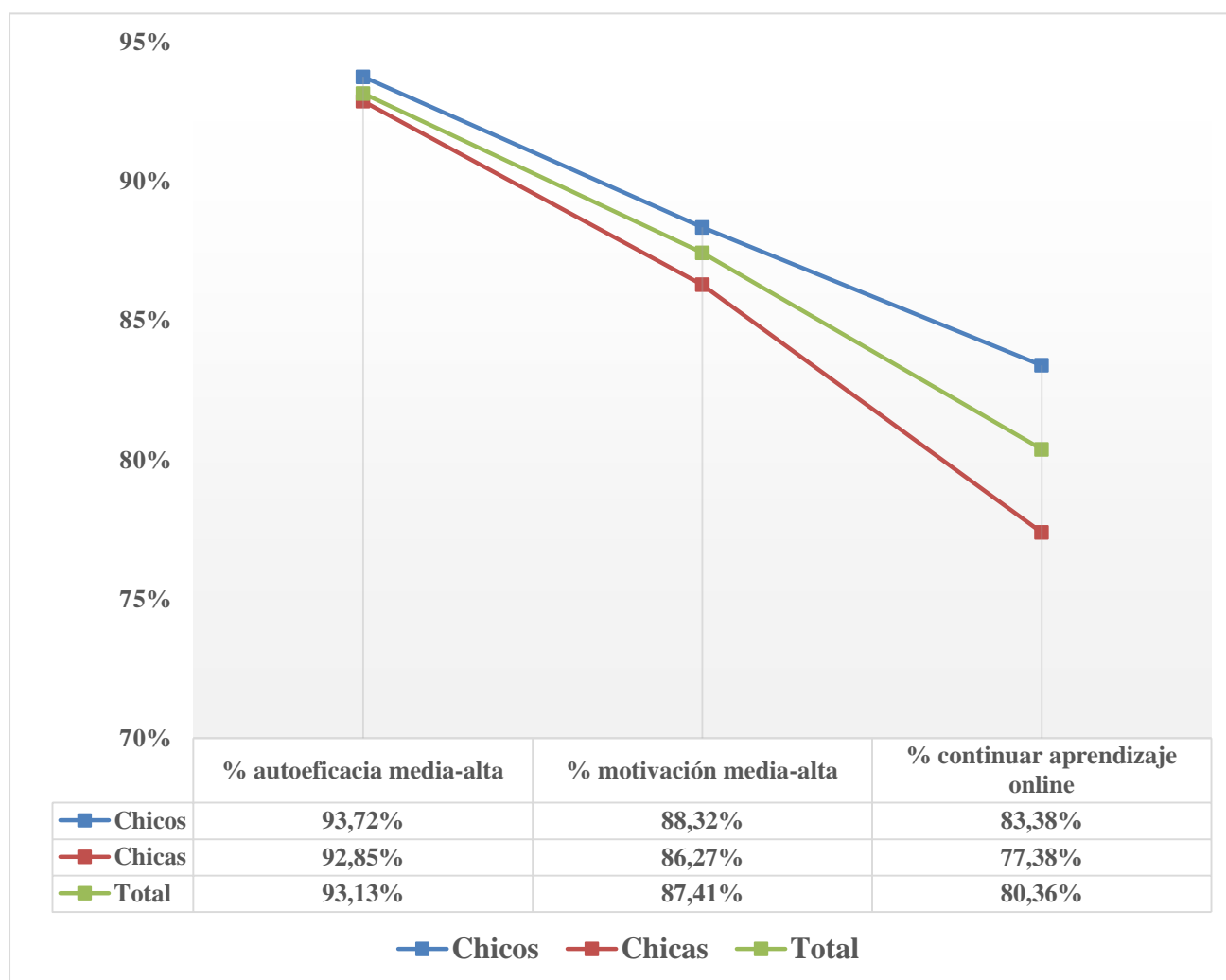


Figura 5.66. Evolución de la ‘cascada motivacional virtuosa’, en función del sexo

³⁴⁸ Se refiere al porcentaje acumulado de sujetos que se sitúan en la parte medio-alta de la escala (puntos 3, 4 o 5 de la escala Likert)

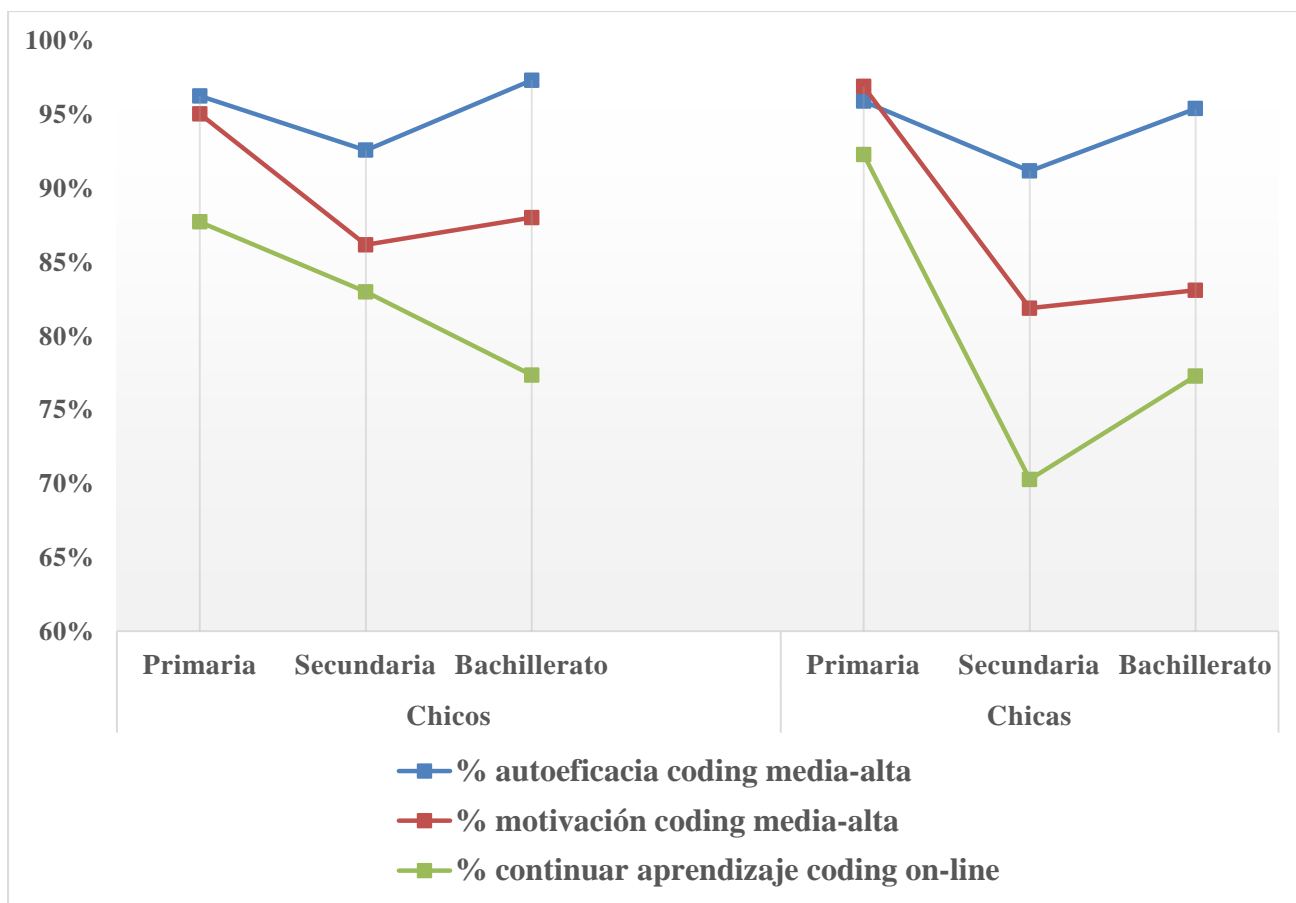


Figura 5.67. Evolución de la ‘cascada motivacional virtuosa’ por sexo y etapa educativa (I)

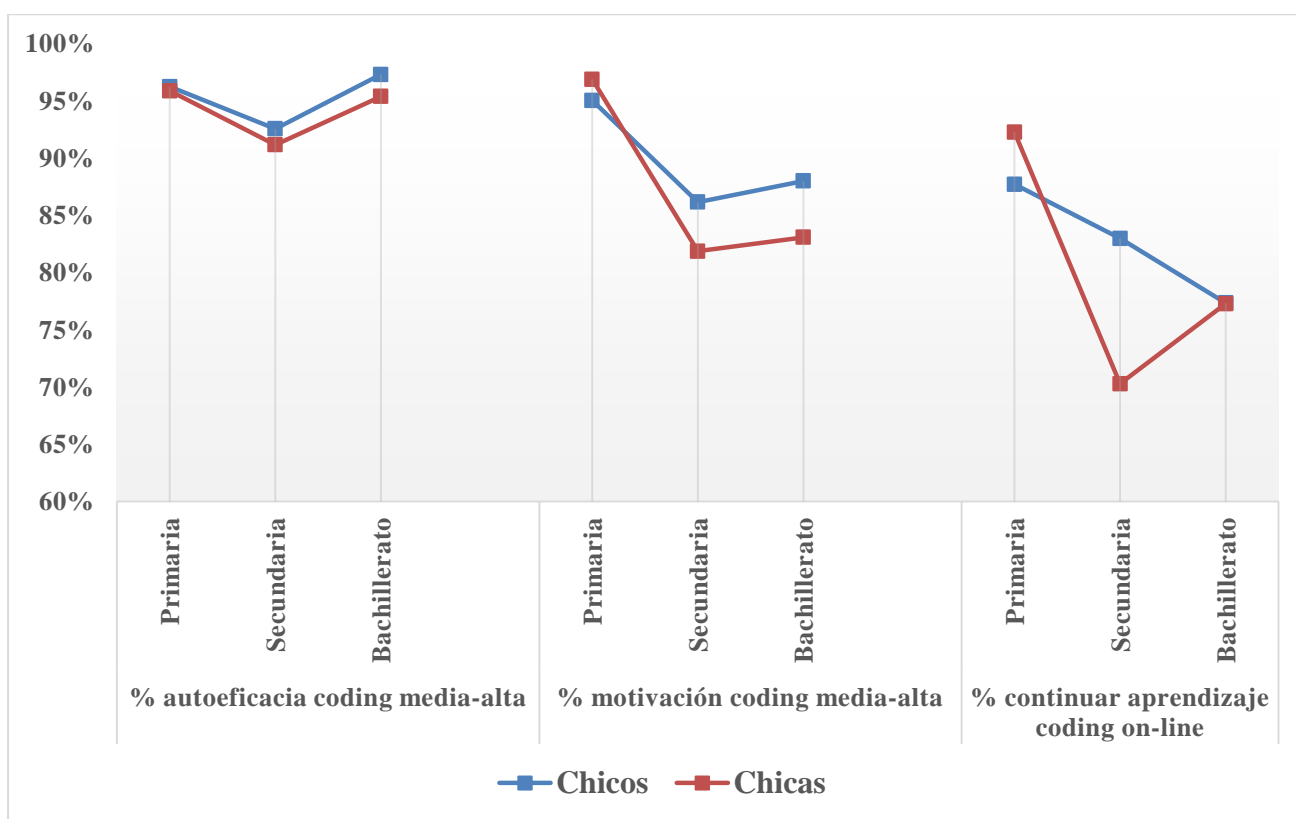


Figura 5.68. Evolución de la ‘cascada motivacional virtuosa’ por sexo y etapa educativa (II)

- El análisis de clúster revela la existencia de cuatro conglomerados de sujetos en función de tener o no tener exposición previa al *'coding'*, y de tener alta o baja autoeficacia-motivación para el *'coding'* tras la actividad HoC. En concreto, los clústeres encontrados son: 1) *'novatos de baja autoeficacia-motivación'* o, simplemente, *'no interesados'*; 2) *'veteranos de autoeficacia media y baja motivación'* o, simplemente, *'desencantados'*; 3) *'veteranos de alta autoeficacia-motivación'* o, simplemente, *'enganchados'*; 4) *'novatos de autoeficacia media y alta motivación'* o, simplemente, *'entusiasmados'*. Casi 8 de cada 10 sujetos (un 79,4%) se sitúa en los clústeres 3 y 4, *'enganchados'* y *'entusiasmados'* respectivamente, que tienen en común su alta motivación para proseguir el aprendizaje del *'coding'*; lo cual puede considerarse un éxito del evento HoC. Sólo un 20,6% de sujetos se sitúan en los clústeres de baja motivación, el clúster 1 (10,0%) de *'no interesados'* y el clúster 2 (10,7%) de *'desencantados'*. Los clústeres 1, 2 y 4 son de ligero predominio femenino, mientras que el clúster 3 es de claro predominio masculino; esta distribución diferencial por sexo no existe en la etapa de Primaria, emergiendo en Secundaria. Adicionalmente, en los centros privados hay predominio de sujetos del clúster 3 (*'enganchados'*) mientras que en los centros públicos lo hay del clúster 4 (*'entusiasmados'*); resultado coherente con el hecho de que en los centros privados hay mayor proporción de estudiantes con experiencia previa en *'coding'* (recordemos que el *'coding'* comenzó a penetrar en el sistema educativo fundamentalmente a través de estos centros privados)
- A través de un análisis de regresión múltiple, encontramos un modelo explicativo-predictivo parsimonioso de la motivación de los estudiantes para proseguir su aprendizaje con el *'coding'* tras la actividad HoC. El 42,1% de la varianza de dicha motivación queda explicado de manera directa, y de mayor a menor peso relativo, por la autoeficacia percibida en *'coding'* tras la actividad, por la percepción de que el *'coding'* contribuye a desarrollar la creatividad y la resolución de problemas; y, en sentido inverso, por la etapa educativa del estudiante. En síntesis, para promover una alta motivación de los estudiantes en continuar aprendiendo *'coding'* es crítico plantear tareas que sean capaces de acometer con éxito y que les planteen problemas significativos que puedan abordarse desde un plano creativo; ello parece estar sucediendo más adecuadamente en Primaria que en Secundaria.
- En el caso más específico de estar o no determinado a continuar el aprendizaje del *'coding'* a través de plataformas-tutoriales *'on-line'* autoguiados (al propio ritmo, de manera interactiva...), obtenemos una función discriminante que es capaz de predecir correctamente el 86,9% de los casos. Dicha predicción se hace en base a cuatro variables: la motivación del estudiante por seguir aprendiendo *'coding'* y a su percepción de autoeficacia, como era esperable; pero también son críticas en la función discriminante, la experiencia previa en *'coding'* del estudiante y su percepción de que éste contribuye al desarrollo del pensamiento lógico. Parece, pues, que el perfil más específico del estudiante que se declara proclive al aprendizaje del *'coding on-line'*, es el de un sujeto que ya ha tenido contacto con la programación informática y que es especialmente sensible a sus aspectos formales-lógicos.
- La anterior función discriminante se muestra poco eficaz en un grupo concreto de 135 sujetos que, aún con niveles suficientes de motivación, autoeficacia y experiencia previa en *'coding'*; finalmente declaran no estar dispuestos a proseguir su aprendizaje mediante el sistema de

plataformas-tutoriales ‘*on-line*’ autoguiados. Dicho grupo está compuesto mayoritariamente por chicas de Educación Secundaria (del clúster 2 o ‘*desencantadas*’), que parecen estar demandando otros contextos o metodologías diferentes, más significativos y cercanos a sus intereses, para seguir aprendiendo ‘*coding*’. Estos resultados son consistentes con estudios previos que afirman que las chicas en edades ‘*middle school*’ (6º Primaria a 2º ESO) se ven más atraídas hacia un aprendizaje del ‘*coding*’ que incluya entornos ‘*off-line*’ desde una aproximación kinestésica (p.e. aprender secuencias y bucles a través de coreografías de baile) (Daily, Leonard, Jörg, Babu, & Gundersen, 2014; Leonard *et al.*, 2015)

- Con respecto al grado de conocimiento de los estudiantes de las distintas plataformas-tutoriales ‘*on-line*’ de aprendizaje del ‘*coding*’, se detectan 3 grupos: a) el formado por Code.org y Scratch, con una alta penetración que se sitúa en el orden del 30-40% del total de la muestra; b) el formado por Light-Bot, App Inventor, Khan Academy, Codecademy, Tynker, Blockly y CodeHS, con una penetración moderada en el orden del 5-10%; y c) el resto de plataformas-tutoriales, con una baja penetración que no llega al 1% sobre el total de la muestra.
- Al comparar entre los estudiantes que siguieron el tutorial HoC de Code.org (o ‘*coders*’), y los que siguieron el tutorial HoC de Scratch (o ‘*scratchers*’) encontramos algunas diferencias de interés. Hay ‘*coders*’ tanto en Primaria como en Secundaria, mientras que los ‘*scratchers*’ sólo emergen masivamente en Secundaria. Los ‘*coders*’ perciben en mayor medida que los ‘*scratchers*’ cómo la programación contribuye a desarrollar sus aptitudes-habilidades, especialmente el trabajo en equipo y la persistencia-perseverancia; y acaban la actividad con niveles significativamente mayores de autoeficacia y motivación para seguir aprendiendo programación, también si es a través de entornos ‘*on-line*’. Así, globalmente, parece que los tutoriales de Code.org, en el contexto de ‘La Hora del Código’, ofrecen ventajas respecto a los de Scratch. Este hecho es consistente con el diseño de los tutoriales de Code.org, muy orientados a la solución progresiva de puzzles breves de programación, en un entorno gamificado y con *feed-back* inmediato que consigue, en sólo 1 hora, elevar la sensación de competencia y motivación a corto plazo del aprendiz (habrá que ver por cuánto tiempo un sujeto podría continuar aprendiendo sólo a través de este sistema ‘a corto plazo’); mientras que Scratch se orienta a la creación de proyectos abiertos a través del código, exigiendo procesos de aprendizaje más largos (y, al principio del proceso, bastante dirigidos por el docente)

Con respecto al grupo de objetivos $O_{2EK} \dots O_{2EI}$, en el marco de la hipótesis de **relevancia**, podemos concluir que:

- Desde un punto de vista cuantitativo, hay una mayoría abrumadora de estudiantes, superior al 80%, que afirman la importancia del ‘*coding*’ para su futuro. Si bien dicho porcentaje es algo superior cuando se les pregunta sobre la relevancia del ‘*coding*’ para su formación personal, como ciudadanos (87,6%); que cuando se les pregunta sobre lo mismo para su formación profesional, como trabajadores (82,1%)

- Desde un punto de vista cualitativo, las razones esgrimidas por los estudiantes para justificar la relevancia del aprendizaje del *'coding'* son, de mayor a menor frecuencia de aparición: las razones sociológicas ('el *coding* como respuesta a un mundo-sociedad tecnológico y digital), seguidas por las razones utilitarias ('el *coding* como herramienta para conseguir beneficios personales y profesionales'), razones de desarrollo ('el *coding* como potenciador de una mente lógica y solucionadora de problemas), razones genéricas, razones expresivo-comunicativas ('el *coding* como vía de expresión a través de la creación de objetos digitales'), emotivas ('el *coding* como fuente de diversión y satisfacción'), escolares ('el *coding* como parte de un emergente currículum escolar'), y, por último, las razones críticas ('el *coding* como elemento de empoderamiento y alfabetización crítica en la sociedad digital; programar para evitar ser programado'). Esta ordenación es consistente con los mensajes lanzados desde los *leitmotifs*, lemas y *merchandising* del evento 'La Hora del Código'; según vimos en el [sub-epígrafe 2.2.3.1](#).
- En conjunto, la inmensa mayoría de los estudiantes se muestran sensibles a la relevancia del *'coding'* para su futuro; con lo cual, parece lograrse otra de las finalidades principales del evento HoC. Ahora bien, cuando entramos en el análisis cualitativo de las razones que sustentan dicha afirmación encontramos un panorama, algo preocupante, excesivamente utilitarista y con tintes de determinismo sociológico ('el mundo es y será digital; y debo saber *'coding'* para aprovechar y sacar beneficio de esa circunstancia'); en detrimento de los aspectos creativos, comunicativos y críticos, que deberían promoverse activamente desde el sistema educativo. Estas cuestiones han sido señaladas recientemente por Williamson (2015), que llama la atención, de manera muy crítica, sobre el currículum oculto de parte del movimiento *'code literacy'* y de las iniciativas *'learning to code'*. En su opinión, detrás de estos acontecimientos, se esconde la intención de que las nuevas generaciones interioricen un sistema de pensamiento que asuma que cualquier problema, incluidos los problemas políticos, puede ser resuelto si se aplica el algoritmo o código correcto (Williamson, 2015). Esta idea serviría de justificación y soporte a las emergentes formas de 'gobernanza digital o algorítmica' (Sadin, 2015), en la cual los gobiernos rehúyen sus responsabilidades desplazándolas a problemas técnicos que pueden resueltos por algoritmos, si a éstos se les proporcionan los datos suficientes (y de ahí la justificación de obtener datos continuamente de los ciudadanos). En nuestra opinión, el movimiento *'code literacy'* sería una oportunidad precisamente para lo contrario: que los estudiantes no den por hecho el funcionamiento de las pantallas con las que interactúan a diario, sino que aprendan que dichas pantallas están programadas, con arreglo a unos intereses dados, y que ellos tienen la capacidad de reprogramarlas llegado el caso. El campo de batalla está abierto.

CAPÍTULO 6

SEGUNDO ESTUDIO: EL TEST DE PENSAMIENTO COMPUTACIONAL (TPC)

6.1. Antecedentes

Tal y como justificamos en la parte teórica de esta tesis doctoral (Capítulo 3), el pensamiento computacional (PC) se viene situando en el foco de la innovación educativa como un conjunto de habilidades de solución de problemas que debe ser adquirido por las nuevas generaciones de estudiantes. Sin embargo, no existe aún consenso internacional sobre cómo definir el PC, ni tampoco una idea clara sobre cómo incorporarlo a los sistemas educativos en sus distintas etapas. Igualmente, hay un enorme vacío sobre cómo medir y evaluar el PC.

En respuesta, abordamos la construcción y validación de un ‘Test de Pensamiento Computacional’ (TPC) específicamente diseñado para población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO), pero con aplicaciones complementarias en el último ciclo de Educación Primaria (5º Primaria – 6º Primaria) y en el segundo ciclo de la Educación Secundaria Obligatoria (3º ESO y 4º ESO).

Para ello, y previa enunciación explícita de los objetivos específicos que nos planteamos en este estudio, dividiremos el proceso en tres fases bien diferenciadas:

- ✓ Fase I: acerca de la construcción-diseño del TPC; y su validación de contenido a través del juicio de expertos.
- ✓ Fases II: acerca de la virtualización-digitalización del TPC; y su validación aparente.
- ✓ Fase III: acerca de la puesta en servicio y aplicación del TPC; y el análisis de su psicometría general (descriptivos), fiabilidad (consistencia interna y estabilidad) y validez (criterial, discriminante, convergente, y factorial)

El capítulo finaliza con la discusión de los resultados encontrados a lo largo de las 3 fases anteriores, y la enunciación de algunas conclusiones parciales al respecto.

6.2. Objetivos específicos

En este Capítulo 6 se aborda el objetivo general **O₃** [‘Diseñar y validar un instrumento que mida el nivel de desarrollo del pensamiento computacional, o Test de Pensamiento Computacional (TPC), en población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO)’]. En el marco de este objetivo general, nos planteamos los siguientes objetivos específicos:

- **O_{3a}**: Diseñar un Test de Pensamiento Computacional (TPC) que siga las directrices fijadas por Buffum *et al.* (2015)³⁴⁹ sobre la construcción de instrumentos de medida en el área de las Ciencias de la Computación para edades ‘middle school’.
- **O_{3b}**: Someter al TPC a un proceso de validación, y depuración subsiguiente, de su contenido a través del juicio de expertos.
- **O_{3c}**: Virtualizar-digitalizar el TPC, de manera que pueda ser accesible a través de dispositivos fijos o móviles con conexión a Internet.
- **O_{3d}**: Someter al TPC a un proceso de validación aparente, que garantice su correcta legibilidad y navegación.
- **O_{3e}**: Aplicar el TPC sobre una muestra representativa y de tamaño suficiente de la población objetivo.
- **O_{3f}**: Describir la psicometría general del TPC: estadísticos descriptivos, baremos, y análisis de ítems.
- **O_{3g}**: Estudiar la fiabilidad del TPC como consistencia interna.
- **O_{3h}**: Estudiar la fiabilidad del TPC como estabilidad.
- **O_{3i}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘razonamiento lógico’.
- **O_{3j}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘espacial’.
- **O_{3k}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘numérica’.
- **O_{3l}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘verbal’.
- **O_{3m}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘resolución de problemas’.
- **O_{3n}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘perceptivo-atencional’.
- **O_{3ñ}**: Estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de personalidad.
- **O_{3o}**: Estudiar la validez criterial (predictiva) del TPC con respecto al rendimiento académico en varias asignaturas.

³⁴⁹ Recordamos al lector que dichas directrices se revisaron en el epígrafe 3.3.2 (Capítulo 3) de la parte teórica.

- **O_{3p}**: Estudiar la validez criterial (predictiva) del TPC con respecto a las estadísticas de desempeño en el curso ‘*K-8 Intro to Computer Science*’ ([Code.org, 2015a](#)).
- **O_{3q}**: Estudiar la validez discriminante del TPC para detectar sujetos de alta capacidad computacional.
- **O_{3r}**: Estudiar la validez convergente del TPC con respecto a una selección de tareas Bebras³⁵⁰
- **O_{3s}**: Estudiar la validez convergente del TPC con respecto a la herramienta Dr. Scratch³⁵¹
- **O_{3t}**: Estudiar la validez factorial del TPC.

6.3. Fase I: Diseño y validación de contenido

La Fase I arranca con el diseño de una primera versión del Test de Pensamiento Computacional (TPC), siguiendo las directrices fijadas por Buffum *et al.* (2015) sobre la construcción de instrumentos de medida en el área de las Ciencias de la Computación para edades ‘*middle school*’; cubriendo así el objetivo específico **O_{3a}**. Por tanto, nuestro TPC es diseñado inicialmente (*versión 1.0, de octubre de 2014*) bajo los siguientes principios y directrices (Román-González, 2015a, 2015b):

- **Objetivo:** el TPC pretende medir el nivel de desarrollo del pensamiento computacional (PC) en el sujeto.
- **Definición operativa del constructo medido**³⁵²: el pensamiento computacional es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica inherente a los lenguajes informáticos de programación: secuencias o direcciones básicas, bucles, condicionales, funciones, y variables.
- **Población objetivo:** el TPC está diseñado y dirigido, como foco central, a población escolar española de entre 12 y 13 años (1º y 2º ESO).
- **Tipo de instrumento:** prueba objetiva de elección múltiple con 4 opciones de respuesta (sólo 1 correcta).
- **Longitud:** 40 ítems.
- **Tiempo estimado de realización:** 45 minutos.

Cada uno de los ítems está diseñado y caracterizado a través de los siguientes cinco ejes de diseño: ‘concepto computacional abordado’; ‘entorno-interfaz del ítem’; ‘estilo de las alternativas de respuesta’; ‘existencia o inexistencia de anidamiento’; y ‘tarea requerida’. A continuación, describimos con más detalle cada uno de estos cinco ejes de diseño, acompañándolos de una breve justificación que remite a lo ya visto en la parte teórica.

³⁵⁰ Recordamos al lector que las tareas Bebras se describieron en el sub-epígrafe 3.3.2.1 (Capítulo 3) de la parte teórica.

³⁵¹ Recordamos que la herramienta Dr. Scratch se describió en el sub-epígrafe 3.3.2.2 (Capítulo 3) de la parte teórica.

³⁵² Así ya enunciada en el epígrafe 3.1.5 (Capítulo 3) de la parte teórica.

- **Concepto computacional abordado:** coherentemente con la definición operativa del PC dada más arriba, cada ítem aborda uno o más de los siguientes 8 conceptos computacionales, ordenados en una dificultad creciente: Direcciones o secuencias básicas (5 ítems); Bucles – ‘repetir veces’ (5 ítems); Bucles – ‘repetir hasta’ (5 ítems); Condicional simple – ‘if’ (5 ítems); Condicional compuesto – ‘if/else’ (5 ítems); Mientras que – ‘while’ (5 ítems); Funciones simples (5 ítems); Funciones con parámetros³⁵³ (5 ítems).

Justificación: Estos conceptos están alineados con los estándares de aprendizaje que fija la ‘Computer Science Teachers Association’ (CSTA) para la educación en ciencias de la computación en estas edades (CSTA, 2011). En concreto, en su Nivel 2 (‘Level 2’, correspondiente a los Grados 6-9 en el sistema norteamericano, equivalentes a 6º Primaria - 3º ESO en el sistema educativo español), se fija el siguiente estándar en lo relativo a los conceptos computacionales que deben dominarse en estas edades: “CPP.L2-05: Implement problem solutions using a programming language, including: looping behavior, conditional statements, logic expressions, variables, and functions” (CSTA, 2011, p. 17), es decir, implementar soluciones a problemas utilizando (los conceptos inherentes a) un lenguaje de programación: bucles, sentencias condicionales, expresiones lógicas, variables y funciones.

Además, los conceptos computacionales abordados por el TPC se alinean con los que proponen los 2 instrumentos específicamente revisados en el epígrafe 3.3.2 (Capítulo 3): a saber, el ‘Test for Measuring Basic Programming Abilities’ (Mühling, Ruf, & Hubwieser, 2015), y el ‘Commutative Assessment’ (Weintrop & Wilensky, 2015a).

- **Entorno-Interfaz del ítem:** los ítems del TPC se presentan en alguno de los siguientes dos entornos gráficos o interfaces: ‘El Laberinto’ (31 ítems); ‘El Lienzo’ (9 ítems).

Justificación: entre las interfaces utilizadas para el aprendizaje del ‘coding’ que revisamos en el epígrafe 2.4.1 (Capítulo 2) de la parte teórica, las más recurrentes, por su funcionalidad para representar la mayoría de los conceptos computacionales, son los laberintos o similares (Figura 2.19) y los lienzos o pizarras en los que se programan diseños geométricos computacionales (Figura 2.20)]

- **Estilo de las alternativas de respuesta:** en cada ítem, las alternativas de respuesta se pueden presentar en alguno de estos tres estilos: ‘Visual por flechas’ (8 ítems); ‘Visual por bloques’ (24 ítems); ‘Textual’ (8 ítems)

Justificación: al revisar en el epígrafe 2.4.1 (Capítulo 2) los tipos de recursos existentes para el aprendizaje del ‘coding’, vimos que éstos se presentaban a través de tres tipos de lenguajes, de menor a mayor dificultad: lenguajes visuales ‘por flechas’ (aptos incluso para pre-lectores); lenguajes visuales ‘por bloques’; y lenguajes ‘textuales’. La solución a un determinado problema computacional puede representarse en cualquiera de ellos (Figura 6.1).

³⁵³ Las ‘funciones con parámetros’ son la manera habitual de introducir las ‘variables’. Por ejemplo, mientras que una ‘función simple’ puede expresar el algoritmo para dibujar un cuadrado de lado 100 cm. como “Cuadrado = 4 * [avanzar 100 cm. + girar derecha 90°]”; una ‘función con parámetros’ puede expresar el algoritmo general para dibujar cualquier polígono regular de N lados (variable), de 100 cm. cada uno, como “Polígono_(N lados) = N * [avanzar 100 cm. + girar derecha (360/N)°]”

Adicionalmente, el hecho de presentar a lo largo del TPC las alternativas de respuesta en diferentes formatos se alinea con el estándar CSTA siguiente: “*CT.L2-07: Represent data in a variety of ways including text, sounds, pictures, and numbers*” (CSTA, 2011, p. 16), es decir, representar los datos en múltiple variedad de formas incluyendo imágenes, texto, números y sonidos. Por último, el ‘*Commutative Assessment*’ (Weintrop & Wilensky, 2015a) reconoce explícitamente la importancia de presentar alternativamente estilos de respuesta visuales y textuales.

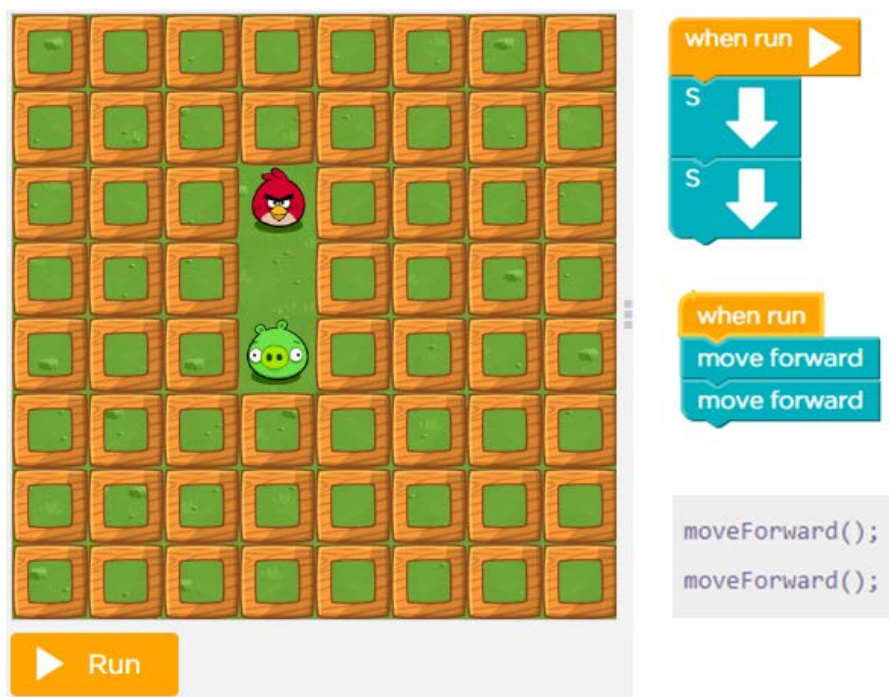


Figura 6.1. Un problema computacional (a la izquierda, planteado en interfaz tipo ‘laberinto’), y su solución expresada en lenguaje ‘por flechas’ (arriba derecha), visual ‘por bloques’ (centro derecha) y ‘textual’ (abajo derecha)

- **Existencia o inexistencia de anidamiento:** dependiendo de si la solución del ítem involucra una secuencia de comandos-órdenes con (29 ítems) o sin (11 ítems) conceptos computacionales anidados (un concepto embebido en otro concepto en un orden de jerarquía superior)

Justificación: para manejar la progresión natural en complejidad de los programas, éstos se articulan en estructuras jerárquicas y anidadas. Por ello, es importante el dominio de los conceptos computacionales en un primer momento ‘sin anidar’, para posteriormente manejarlos de manera ‘anidada’. Así se reconoce en los estándares CSTA: “*CT.L2-12. Use abstraction to decompose a problem into sub problems; CT.L2-13. Understand the notion of hierarchy and abstraction in computing including high-level languages, translation, instruction set, and logic circuits*” (CSTA, 2011, p. 16). Además, el ‘*Test for Measuring Basic Programming Abilities*’ (Mühling, Ruf, & Hubwieser, 2015) reconoce explícitamente la importancia de introducir el anidamiento de conceptos computacionales, para medir el nivel de PC en los estudiantes.

- **Tarea requerida:** dependiendo de cuál de las siguientes tareas cognitivas es necesaria para la resolución del ítem: ‘Secuenciación’, el estudiante debe enunciar de manera ordenada una serie de comandos-órdenes (18 ítems); ‘Completamiento’, el estudiante debe completar un conjunto incompleto de comandos previamente dado (14 ítems); ‘Depuración’, el estudiante debe depurar (*debug*) un conjunto incorrecto de comandos previamente dado (8 ítems).

Justificación: dentro de las limitaciones inherentes a una prueba de elección múltiple como la que planteamos con el TPC, en la que el sujeto finalmente lo que hace es ‘elegir’ entre una serie de opciones dadas; tratamos de diversificar la tarea cognitiva requerida a través de ítems de ‘secuenciación’, ‘completamiento’ y ‘depuración’. Este trío de tareas cognitivas requeridas a lo largo del TPC resuenan respectivamente a varias de las ‘prácticas computacionales’ (*computational thinking practices*) del modelo MIT-Harvard (Brennan & Resnick, 2012) que revisamos en la parte teórica, en concreto: ‘secuenciación’ \approx ‘experimentación e iteración’; ‘completamiento’ \approx ‘reutilización y remezcla’; ‘depuración’ \approx ‘evaluación y depuración’. Adicionalmente, con ello conseguimos alinearnos con el estándar principal sobre pensamiento computacional para estas edades fijado por la CSTA: “*CT.L2-01: Use the basic steps in algorithmic problem-solving to design solutions (e.g., problem statement and exploration, examination of sample instances, design, implementing a solution, testing, evaluation)*” (CSTA, 2011, p.16), es decir, utilizar secuencias básicas de pasos para la resolución algorítmica de problemas (lo que incluye el diseño de algoritmos, su revisión, completamiento, evaluación y depuración)

En la Figura 6.2 se muestran las especificaciones de los 40 ítems de esta versión inicial del TPC (*versión 1.0*), que son caracterizados en los 5 ejes de diseño señalados anteriormente.

PARTE EMPÍRICA / Capítulo 6 / Román-González (2016)

	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
			Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Item 1	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Secuenciación
Item 2	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Completamiento
Item 3	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Depuración
Item 4	Laberinto	Textual	Sí	No	No	No	No	No	No	No	No	Secuenciación
Item 5	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Secuenciación
Item 6	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Secuenciación
Item 7	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Completamiento
Item 8	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación
Item 9	Lienzo	Textual	Sí	Sí	No	No	No	No	No	No	No	Depuración
Item 10	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Completamiento
Item 11	Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No	No	Secuenciación
Item 12	Laberinto	Visual por bloques	Sí	No	Sí	No	No	No	No	No	No	Completamiento
Item 13	Laberinto	Textual	Sí	No	Sí	No	No	No	No	No	No	Secuenciación
Item 14	Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	Sí	Depuración
Item 15	Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	Sí	Secuenciación
Item 16	Laberinto	Visual por flechas	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación
Item 17	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación
Item 18	Laberinto	Textual	Sí	No	Sí	Sí	No	No	No	No	Sí	Completamiento
Item 19	Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	Sí	Completamiento
Item 20	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Depuración
Item 21	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación
Item 22	Laberinto	Textual	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación
Item 23	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Depuración
Item 24	Laberinto	Textual	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento
Item 25	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento
Item 26	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación
Item 27	Laberinto	Textual	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación
Item 28	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Depuración
Item 29	Laberinto	Textual	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento
Item 30	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento
Item 31	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación
Item 32	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento
Item 33	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Depuración
Item 34	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación
Item 35	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento
Item 36	Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Secuenciación
Item 37	Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Completamiento
Item 38	Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Depuración
Item 39	Lienzo	Visual por bloques	No	No	No	No	No	No	No	Sí	Sí	Secuenciación
Item 40	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	Sí	Sí	Completamiento

Figura 6.2. Cuadro resumen de especificaciones de los 40 ítems del TPC (versión 1.0) en los 5 ejes de diseño

En el **Anexo E** se aportan en detalle los 40 ítems del TPC (*versión 1.0*), así como tabla de especificaciones para cada uno de ellos. Aquí baste con presentar 4 de estos ítems como ejemplos que ilustren el proceso de diseño seguido (Figuras 6.3 a 6.6)

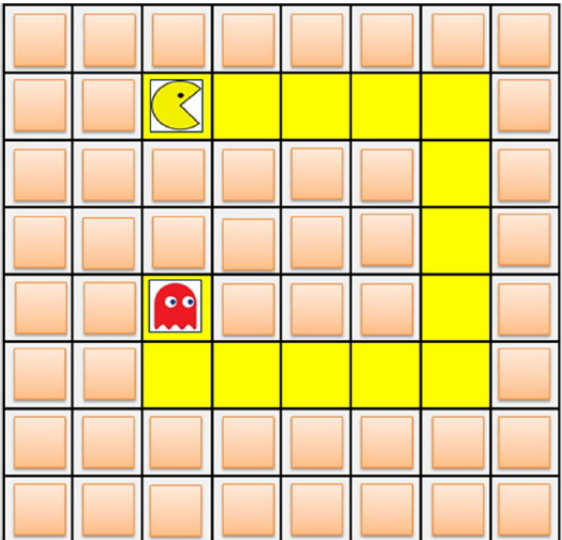

<p>¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?</p> 	<p>Opción A</p> <pre> repetir 4 veces haz repetir 3 veces haz avanzar girar a la derecha avanzar </pre>	<p>Opción B </p> <pre> repetir 3 veces haz repetir 4 veces haz avanzar girar a la derecha avanzar </pre>
	<p>Opción C</p> <pre> repetir 3 veces haz repetir 4 veces haz avanzar girar a la derecha avanzar </pre>	<p>Opción D</p> <pre> repetir 4 veces haz avanzar repetir 3 veces haz girar a la derecha avanzar </pre>

Figura 6.3. Se corresponde con el ítem 8 del TPC (*versión 1.0*) [Bucles – repetir veces; ‘El Laberinto’; Visual por bloques; Con anidamiento; Secuenciación]

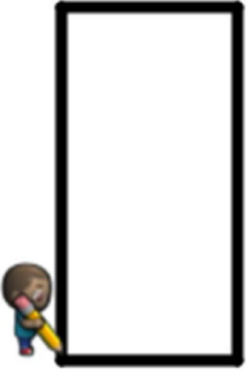

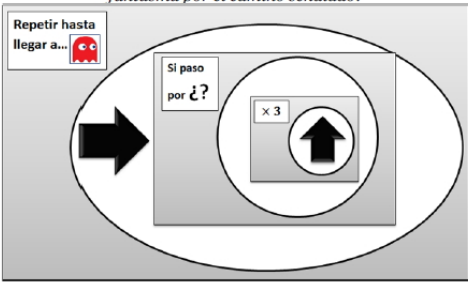
<p>Para que el artista dibuje una vez el siguiente rectángulo (50 píxeles de ancho y 100 píxeles de alto), ¿en qué paso de la siguiente secuencia de órdenes hay un error?</p> 	<p>Opción A </p> <p>4 × [Avanzar 50 px. + Girar a la izquierda + Avanzar 100 px. + Girar a la izquierda]</p> <hr/> <p>Opción B</p> <p>4 × [Avanzar 50 px.] + Girar a la izquierda + Avanzar 100 px. + Girar a la izquierda]</p> <hr/> <p>Opción C</p> <p>4 × [Avanzar 50 px. + Girar a la izquierda] + Avanzar 100 px. + Girar a la izquierda]</p> <hr/> <p>Opción D</p> <p>4 × [Avanzar 50 px. + Girar a la izquierda + Avanzar 100 px.] + Girar a la izquierda]</p>
--	--

Figura 6.4. Se corresponde con el ítem 9 del TPC (*versión 1.0*) [Bucles – repetir veces; ‘El Lienzo’; Textual; Sin anidamiento; Depuración]

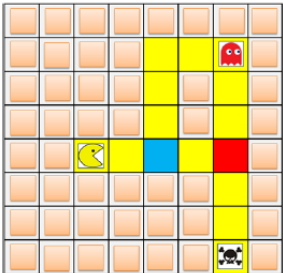
¿Qué falta en la siguiente secuencia de órdenes para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?




Repetir hasta llegar a...

Si paso por ¿?


x 3




Opción A



Opción B



Opción C



Opción D

Tanto la opción A como la opción C son correctas



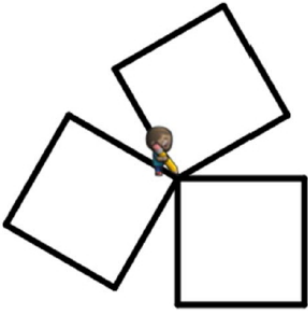


Figura 6.5. Se corresponde con el ítem 19 del TPC (versión 1.0) [Bucles – repetir hasta + Condicional simple; ‘El Laberinto’; Visual por flechas; Con anidamiento; Completamiento]

Si tenemos el siguiente conjunto de órdenes, al que llamamos ‘my function’:



¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? Cada uno de los lados de cada cuadrado mide 100 píxeles.



Opción A

repetir 3 veces

haz my function

girar a la derecha por 120 grados

Opción B

repetir 3 veces

haz my function

girar a la derecha por 120 grados

Opción C

repetir 4 veces

haz my function

girar a la derecha por 90 grados

Opción D

repetir 4 veces

haz my function

girar a la derecha por 90 grados




Figura 6.6. Se corresponde con el ítem 31 del TPC (versión 1.0) [Bucles – repetir veces + Funciones simples; ‘El Lienzo’; Visual por bloques; Con anidamiento; Secuenciación]

6.3.1. Método

Para abordar nuestro objetivo específico O_{3b} , el ‘Test de Pensamiento Computacional’ (versión 1.0) es sometido a un proceso de validación de su contenido a través del método ‘juicio de expertos’. A continuación, se presenta la muestra de expertos participantes en dicho proceso; los instrumentos, y el procedimiento de validación utilizado con detalle; y, con posterioridad, los resultados del mismo que cristalizan finalmente en el ‘Test de Pensamiento Computacional’ (versión 2.0)

6.3.1.1. *Muestra*

Para el proceso de validación de contenido de nuestro TPC (*versión 1.0*), 39 expertos son invitados inicialmente a colaborar en la emisión de juicio y valoración sobre el instrumento. Siguiendo las recomendaciones de Buffum *et al.* (2015), los expertos invitados son principalmente profesores del área de Informática, tanto de etapa universitaria como pre-universitaria.

Finalmente, 20 expertos aceptan. Este panel de expertos tiene las siguientes características: 14 hombres y 6 mujeres; edad media 36,9 años; pertenecientes a los siguientes colectivos profesionales (es posible pertenecer a más de uno simultáneamente), tal y como se muestra en la Tabla 6.1 y 6.2.

Tabla 6.1. Panel de expertos aceptantes para participar en el proceso de validación de contenido

<i>Apellidos</i>	<i>Nombre</i>	<i>Sexo</i>	<i>Edad</i>	<i>¿A cuál/es de los siguientes colectivos pertenece?</i> ³⁵⁴
Sáez López	José Manuel	Hombre	37	Uni
López Sánchez	Joaquín	Hombre	44	ESO
Coloma Brotons	Antonio	Hombre	40	ESO, Bach
Stabile Rodríguez-Vergés	Lidia	Mujer	53	ESO, Bach
Escribano Álvarez	David	Hombre	35	ESO, FP, Progr, APICV
Mollá Vayá	Gustavo	Hombre	44	ESO, Bach, FP
Casas Barrado	Manuel	Hombre	23	Apps
López de la Hoz	Sandra	Mujer	35	ESO, Bach, FP, APICV
Hernán	Isidoro	Hombre	43	Uni
Navarro Bayo	María Amparo	Mujer	41	ESO, Bach
Bernad Nebot	Carmen	Mujer	51	Bach
Gutiérrez Rojas	Israel	Hombre	33	Uni, Progr
De la Rosa Martín	Adrián	Hombre	19	Apps
Martínez Martínez	Vicente	Hombre	39	ESO, APICV
Pérez Ibáñez	Aránzazu	Mujer	48	ESO, Bach, FP
López Ibarra	David	Hombre	38	ESO, Bach
Martín Gutiérrez	Sergio	Hombre	34	Uni
Fernández Rodríguez	Eva	Mujer	25	Apps
Prudencio Fernández	Jorge	Hombre	22	Apps
Moreno León	Jesús	Hombre	34	ESO, FP, Uni, Progr

Tabla 6.2. Perfil profesional del panel de expertos

<i>Colectivo profesional</i>	<i>Nº de expertos</i>
Profesores de Informática de Educación Secundaria	11
Profesores de Informática de Bachillerato	8
Profesores de Informática de Formación Profesional	5
Profesores de Informática de Universidad	5

³⁵⁴ Abreviaturas utilizadas: Uni = Profesor/a de Universidad; ESO = Profesor/a de Educación Secundaria; Bach = Profesor/a de Bachillerato; FP = Profesor/a de Formación Profesional; Apps = Ganador/a del concurso 'I Edición Premios Apps Fundación Telefónica'; Progr = Miembro del colectivo 'Programamos.es'; APICV = Miembro de la 'Asociación de Profesores de Informática de la Comunidad Valenciana'

<i>Colectivo profesional</i>	<i>Nº de expertos</i>
Ganadores de la 'I Edición Premios Apps Fundación Telefónica' ³⁵⁵	4
Miembros del colectivo Programamos.es ³⁵⁶	3
Miembros de la Asociación de Profesores de Informática de la Comunidad Valenciana ³⁵⁷	3

6.3.1.2. Instrumentos

El TPC (*versión 1.0*³⁵⁸, de octubre 2014) (Román-González, 2014d) es enviado a los 20 expertos aceptantes, a los que se solicita su valoración del mismo a través de 3 cuestionarios en línea (Román-González, 2014e, 2014f, 2014g), soportados con la tecnología de *Google Drive Forms*³⁵⁹ (Figura 6.7)



Figura 6.7. Códigos QR de enlace a los 3 cuestionarios en línea utilizados para el procedimiento de validación de contenido (Román-González, 2014e, 2014f, 2014g)

Las valoraciones que se solicitan a los expertos sobre el TPC (*versión 1.0*) a través de los anteriores cuestionarios son, tal y como recomiendan Buffum *et al.* (2015), una combinación de escalas tipo Likert, preguntas dicotómicas, y preguntas abiertas. En concreto, se solicitan las siguientes valoraciones:

- **Para cada uno de los 40 ítems:** nivel de dificultad del ítem en relación a la población objetivo (escala Likert de 10 puntos, desde '1=Muy fácil' a '10=Muy difícil'); pertinencia del ítem para medir el pensamiento computacional (escala Likert de 10 puntos, desde '1=Nada pertinente' a '10=Absolutamente pertinente'); si incluir o no el ítem en la versión final del TPC (respuesta dicotómica, 'Sí lo incluiría' o 'No lo incluiría'); observaciones y sugerencias abiertas para la mejora del ítem.
- **Para cada uno de los 5 ejes de diseño:** adecuación de haber considerado el eje para el diseño del TPC (escala Likert de 10 puntos, desde '1=Nada adecuado' a '10=Absolutamente

³⁵⁵ En 2014 se convocó la 'I Edición de los Premios Apps de la Fundación Telefónica', un concurso destinado a chicos y chicas de entre 18 y 25 años, para crear-programar aplicaciones móviles; lo que da una idea de la actualidad y relevancia del tema: http://www.fundaciontelefonica.com/es/educacion_innovacion/premios_apps

³⁵⁶ Programamos.es (<http://programamos.es/>) es un proyecto sin ánimo de lucro que pretende acercar la programación a edades tempranas de una manera lúdica; y modificar la forma en la que los estudiantes se relacionan con la tecnología, pasando de ser consumidores a creadores tecnológicos a través de la programación de videojuegos y el desarrollo de aplicaciones para dispositivos móviles. Recientemente vienen desarrollando la herramienta analítica Dr. Scratch (<http://drscratch.programamos.es/>) en colaboración con la ETSI Informática de la Universidad Rey Juan Carlos.

³⁵⁷ <https://www.apicv.es/>

³⁵⁸ Disponible como documento PDF en <https://db.tt/66FJPURK>. En ese momento del proceso, el instrumento aún se denominaba 'Escala de Pensamiento Computacional'. Precisamente por las sugerencias de varios expertos, tal y como se verá más adelante, la denominación se modificó a 'Test de Pensamiento Computacional'

³⁵⁹ [<http://goo.gl/6p1gcR>][<http://goo.gl/JSrHSD>][<http://goo.gl/xCZIU8>]

adecuado’); si incluir o no el eje en el diseño final del TPC (respuesta dicotómica, ‘Sí’ o ‘No’); observaciones y sugerencias abiertas acerca de los ejes de diseño.

- **Para el instrumento en su conjunto:** valoración sobre la longitud del TPC (escala Likert de 5 puntos, desde ‘1=Muy corta’ a ‘5=Muy larga’) y longitud ideal estimada para la versión final (respuesta abierta); valoración del tiempo estimado de realización del TPC (escala Likert de 5 puntos, desde ‘1=Muy corto’ a ‘5=Muy largo’) y tiempo ideal de realización en relación a la longitud ideal (respuesta abierta); valoración global del TPC (escala Likert de 10 puntos, desde ‘1=Muy deficiente’ a ‘10=Excelente’); y comentarios finales abiertos.

6.3.1.3. Procedimiento

El procedimiento seguido se resume en los siguientes pasos:

- i. Conformación de la muestra de expertos invitados (N=39)³⁶⁰
- ii. Contacto con cada uno de los expertos invitados a través de correo electrónico, solicitando su colaboración en la validación de contenido.
- iii. Conformación de la muestra de expertos aceptantes (N=20)
- iv. Envío a cada uno de los expertos aceptantes, a través de correo electrónico, de un documento PDF que incluye el TPC (*versión 1.0*) y los enlaces a los 3 cuestionarios en línea de valoración. En dicho correo, se ofrecen unas ligeras recomendaciones al experto para acometer su tarea de revisión:
 - a. Se le recomienda imprimir el PDF y hacer una primera lectura global del TPC.
 - b. Se le llama la atención con respecto a la posibilidad de completar los 3 cuestionarios en línea en momentos no consecutivos, para facilitarle la descomposición de su tarea en 3 partes más manejables. Así, a través del primer cuestionario se le solicita la valoración de los ítems 1 a 20; a través del segundo cuestionario se le solicita la de los ítems 21 a 30; y a través del tercero se solicita al experto la valoración de cada uno de los ejes de diseño del TPC y la valoración global del mismo.
 - c. Orientativamente, se indica al experto que la tarea en su conjunto puede llevarle unas 2 o 3 horas, y se le da un plazo aproximado de 15 días desde la recepción del correo para su completamiento.
- v. A lo largo de las 2 semanas siguientes al envío del anterior correo electrónico, cada uno de los expertos, de manera independiente, fue completando los 3 cuestionarios de valoración. Más concretamente, el primero de los cuestionarios fue cumplimentado por los 20 expertos aceptantes (100%), el segundo fue cumplimentado por 19 (95%), y el tercer de los cuestionarios por 17 expertos (85%)

³⁶⁰ El grueso de la muestra de expertos invitados a la validación de contenido está formado por los profesores de etapas pre-universitarias (ESO, Bachillerato y FP) que se mostraron especialmente colaborativos durante el primero de nuestros estudios empíricos, sobre ‘La Hora del Código’

- vi. Transcurrido el plazo de cumplimentación de los cuestionarios, se procedió a descargar las respuestas de los expertos desde la base de datos de *Google Drive* a una hoja de cálculo, que fue posteriormente procesada con Excel.

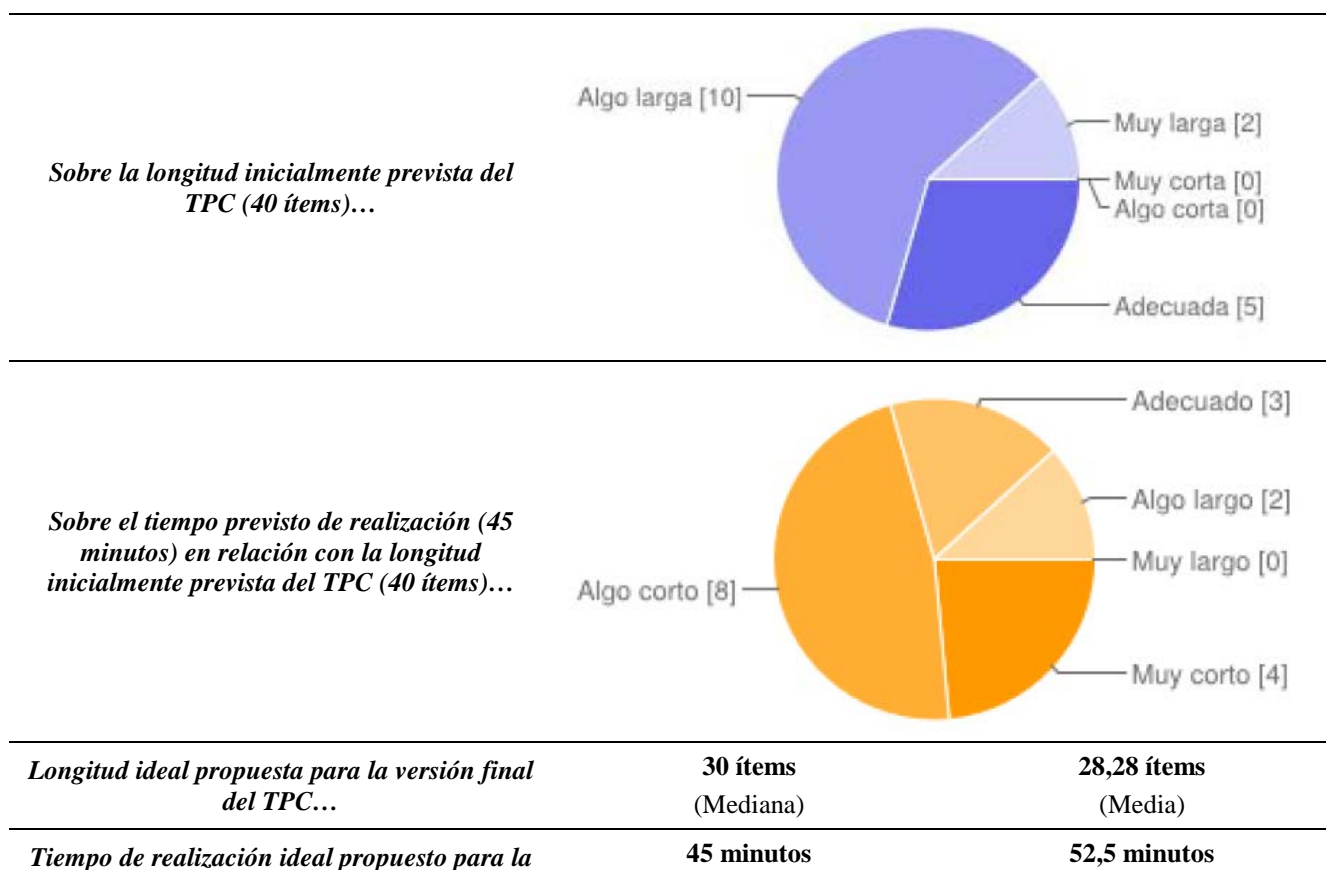
6.3.2. Resultados

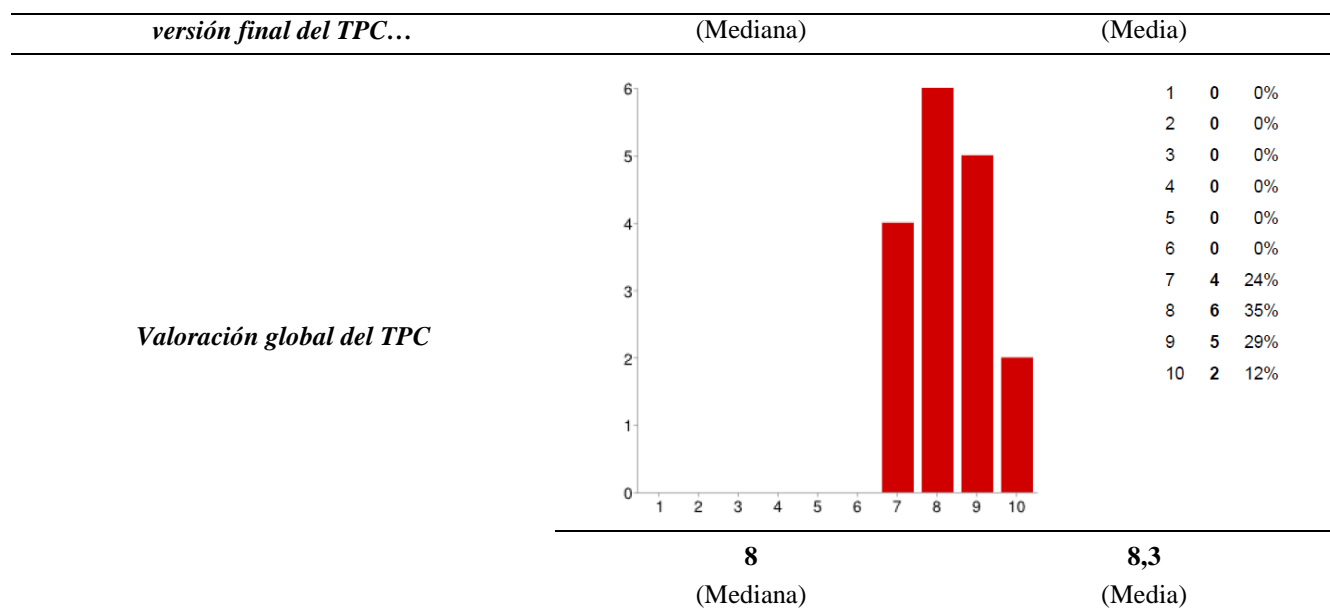
Exponemos los resultados en orden inverso para una mejor comprensión. Así pues, comenzamos por los resultados de la valoración de los expertos sobre el instrumento en su conjunto.

- Acerca de la longitud del TPC (*versión 1.0*), 40 ítems: el 11,8% de los expertos la valoran como ‘*muy larga*’; 58,8% como ‘*algo larga*’; y 29,4% como de longitud ‘*adecuada*’.
- Acerca del tiempo estimado de realización (45 minutos) en relación a la longitud inicialmente prevista (40 ítems): 70,6% de los expertos lo valoran como ‘*muy corto*’ o ‘*algo corto*’; 17,6% como tiempo ‘*adecuado*’; y 11,8% como ‘*algo largo*’.
- La longitud ideal expresada por los expertos para la versión final del TPC es de 28,3 ítems (media) o 30 ítems (mediana); y el tiempo ideal de realización en relación a dicha longitud ideal es de 52,5 minutos (media) o 45 minutos (mediana).
- La valoración global del TPC es de 8 (mediana) u 8,3 (media) sobre 10.

En la Tabla 6.3 se resumen gráficamente estos resultados, y se presenta adicionalmente una síntesis de los comentarios finales abiertos de los expertos.

Tabla 6.3. Resumen de la valoración de los expertos sobre el instrumento en su conjunto





Recomendaciones globales

- Estimar un tiempo de resolución de 1 minuto y medio por ítem (por ejemplo, se recomienda “*un instrumento de 30 ítems para ser resuelto en 45 minutos*”)
- Reformular los ítems que exigen ‘depuración’ de una secuencia errónea de código: se aconseja mostrar sólo una vez el código, y sobre el mismo señalar 4 opciones en las que puede encontrarse el paso erróneo.
- Prescindir del estilo de respuesta ‘textual’ por su excesiva complejidad para la población objetivo.
- Se reconoce la progresiva dificultad del test, y se recomienda prescindir de los ítems finales relativos a las ‘funciones con parámetros’ por su excesiva dificultad para la población objetivo.
- Se advierte frente al uso prematuro en el test de anidamientos, y de su posible complejidad: se debe introducir el anidamiento sólo después de haber tratado y remarcado el concepto computacional correspondiente sin anidar.

Con respecto a los resultados de la valoración de los expertos sobre cada uno de los cinco ejes de diseño del TPC, se sintetizan en la siguiente Tabla 6.4.

Tabla 6.4. Resumen de la valoración de los expertos sobre los ejes de diseño del TPC

<i>Eje de diseño del TPC</i>	<i>Grado de adecuación</i> (desde ‘1=Nada adecuado’ a ‘10=Absolutamente adecuado’)		<i>Índice de aceptación</i> (% de ‘Sí’)	<i>Observaciones y sugerencias abiertas</i>
	<i>Mediana</i>	<i>Media</i>		
	Entorno-Interfaz del reactivo	9		

<i>Eje de diseño del TPC</i>	<i>Grado de adecuación</i> (desde '1=Nada adecuado' a '10=Absolutamente adecuado')		<i>Índice de aceptación</i> (% de 'Sí')	<i>Observaciones y sugerencias abiertas</i>
	<i>Mediana</i>	<i>Media</i>		
<i>Estilo de las alternativas de respuesta</i>	9	8,35	100%	<ul style="list-style-type: none"> ▪ Se estima que los estilos visuales (por flechas y por bloques) son los más adecuados para la edad objetivo ▪ Eliminar el estilo textual (o sustituirlo por 'pseudocódigo' en varias líneas y con sangrado)
<i>Concepto computacional abordado</i>	10	9,00	100%	<ul style="list-style-type: none"> ▪ Excesiva complejidad del último bloque de ítems, en donde se abordan las 'funciones con parámetros'
<i>Existencia o inexistencia de anidamiento</i>	8	8,17	94%	<ul style="list-style-type: none"> ▪ Introducir el anidamiento sólo después de haber introducido y remarcado el concepto correspondiente sin anidar ▪ Eliminar los anidamientos excesivos (dobles y triples anidamientos)
<i>Tarea requerida</i>	10	9,05	100%	<ul style="list-style-type: none"> ▪ Se advierte de la posible complejidad excesiva de la tarea 'depuración' ▪ En los ítems de 'depuración' se aconseja escribir sólo una vez el código, y sobre el mismo señalar 4 opciones en las que puede encontrarse el paso erróneo. ▪ Valorar incluir algún ítem en el que el alumno deba escribir 'desde cero' el algoritmo (evocación)

Con respecto a los resultados de la valoración de los expertos sobre cada uno de los ítems del TPC (*versión 1.0*), se detallan en las siguientes Tablas 6.5 a 6.12. Se presenta una tabla por cada grupo consecutivo de 5 ítems, focalizado en un concepto computacional determinado, incluyendo las decisiones parciales tomadas al respecto de dicho grupo.

Tabla 6.5. Resumen de resultados y decisiones para los ítems 1-5

<i>Número de ítem</i>	<i>Dificultad percibida para la población objetivo</i> (De 1 a 10)		<i>Pertinencia percibida para medir el PC</i> (De 1 a 10)		<i>Índice de aceptación</i> (De 0% a 100%)	<i>Observaciones y sugerencias abiertas</i>
	<i>Mediana</i>	<i>Media</i>	<i>Mediana</i>	<i>Media</i>		
ÍTEM 1	1,0	1,4	8,0	7,4	85%	OK
ÍTEM 2	2,0	2,0	8,5	7,6	90%	OK
ÍTEM 3	3,5	3,9	8,0	7,8	80%	Mostrar una sola vez el código, y marcar sobre el mismo los pasos en los que puede encontrarse el error.
ÍTEM 4	3,0	3,4	8,0	7,6	75%	No se acepta el estilo de respuesta textual
ÍTEM 5	4,0	4,4	8,0	7,9	95%	Incluir alguna clave o pista del sentido inicial de la marcha del lápiz

Decisiones Ítems 1-5 (Concepto computacional medido: ‘Direcciones-secuencias básicas’)

- ✓ En los ítems 1 y 2 se unifica el diseño de la interfaz del laberinto para hacerlo coherente con el resto del instrumento.
- ✓ Se adapta el ítem 3 para que el reactivo incluya ambos sentidos de giro, y que se muestre una sola vez el código a depurar, marcando sobre el mismo los pasos en los cuales puede encontrarse el error.
- ✓ Se elimina el ítem 4 por su bajo índice de aceptación ($\leq 75\%$)
- ✓ Se adapta el ítem 5, incluyendo una clave-pista sobre el sentido inicial de la marcha del lápiz.

Tabla 6.6. Resumen de resultados y decisiones para los ítems 6-10

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 6	2,0	2,3	8,5	7,3	85%	Hacer hincapié en las instrucciones de que ‘Pac-Man’ debe llegar justo hasta el fantasma (sin pasarse), y estrictamente por el camino señalado (sin salirse ni tocar los cuadrados naranjas)
ÍTEM 7	4,0	4,1	9,0	8,1	95%	OK
ÍTEM 8	6,0	6,1	9,0	8,6	100%	Introduce demasiado pronto el anidamiento. Retrasar hasta el final de este grupo de ítems
ÍTEM 9	6,0	6,2	8,0	7,7	65%	No se acepta el estilo de respuesta textual. Al reactivo, tipo lienzo, hay que incluirle una pista del inicio de la marcha de lápiz. Mostrar sólo una vez el código a depurar.
ÍTEM 10	6,0	6,7	9,0	8,3	85%	Excesiva dificultad del reactivo

Decisiones Ítems 6-10 (Concepto computacional medido: ‘Bucles - Repetir veces’)

- ✓ Los ítems 6 y 7 están OK. Pero hay que hacer hincapié en las instrucciones iniciales en que hay que llevar a ‘Pac-Man’ justo hasta la casilla en la que se encuentra el fantasma (sin pasarse), y estrictamente por el camino señalado en amarillo (sin salirse del mismo ni tocar los cuadrados naranjas)
- ✓ El ítem 8, por introducir demasiado pronto el anidamiento de bucles, se retrasa hasta el final de este grupo.
- ✓ Fundimos los ítems 9 y 10: Utilizamos el reactivo del ítem 9 incluyendo la pista sobre el lienzo. Las alternativas son ‘por bloques’. La tarea es depuración, mostrando sólo una vez el código.

Tabla 6.7. Resumen de resultados y decisiones para los ítems 11-15

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 11	4,0	3,9	9,0	7,5	75%	Incluir 2 movimientos en todas las opciones de respuesta para que sean plausibles
ÍTEM 12	4,0	4,0	8,5	7,9	95%	OK
ÍTEM 13	5,5	5,2	7,0	7,2	70%	No se acepta el estilo de respuesta textual. Sobra el último giro a la izquierda
ÍTEM 14	5,5	5,7	7,5	7,6	75%	Mostrar una sola vez el código, y marcar sobre el mismo los pasos en los que puede encontrarse el error.
ÍTEM 15	6,5	6,9	8,0	7,9	85%	Incluir clave visual sobre el sentido inicial del lápiz

Decisiones Ítems 11-15 (Concepto computacional medido: ‘Bucles - Repetir hasta’)

- ✓ En el ítem 11 se incluyen 2 movimientos en todas las opciones de respuesta para que todas las alternativas sean plausibles
- ✓ Ítem 12 se mantiene sin cambios
- ✓ Se elimina el ítem 13 por su bajo índice de aceptación (< 75%)
- ✓ Ítem 14 se adapta para mostrar una sola vez el código erróneo, marcando sobre el mismo los pasos en los que puede encontrarse el error. Se remarca el anidamiento con distintos grosores de línea.
- ✓ En el Ítem 15 se incluye clave visual sobre el sentido inicial del lápiz (y se cambia de opción correcta de la C a la A para re-equilibrar)

Tabla 6.8. Resumen de resultados y decisiones para los ítems 16-20

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 16	5,0	5,5	8,0	7,2	80%	Puede ser difícil comprender el anidamiento con el estilo visual por flechas
ÍTEM 17	6,0	6,0	9,0	8,6	90%	OK
ÍTEM 18	6,5	6,1	8,5	8,0	65%	No se acepta el estilo de respuesta textual
ÍTEM 19	6,5	6,7	8,5	8,0	75%	Puede ser difícil comprender el anidamiento con el estilo visual por flechas
ÍTEM 20	7,0	7,1	8,5	7,8	75%	Mostrar una sola vez el código, y marcar sobre el mismo los pasos en los que puede encontrarse el error.

Decisiones Ítems 16-20 (Concepto computacional medido: ‘Condicional simple (If)’)

- ✓ En el ítem 16 se mejora la representación del anidamiento en el estilo visual por flechas, mediante el uso jerárquico de líneas de grosor
- ✓ Ítem 17 se mantiene sin cambios
- ✓ Se elimina el ítem 18 por su bajo índice de aceptación (< 75%)
- ✓ En el ítem 19 se mejora la representación del anidamiento en el estilo visual por flechas, mediante el uso jerárquico de líneas de grosor
- ✓ El ítem 20 se adapta para mostrar una sola vez el código erróneo, marcando sobre el mismo los pasos en los que puede encontrarse el error.

Tabla 6.9. Resumen de resultados y decisiones para los ítems 21-25

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 21	6,0	5,7	9,0	8,5	89%	OK
ÍTEM 22	6,0	5,7	8,0	8,1	74%	No se acepta el estilo de respuesta textual
ÍTEM 23	8,0	7,2	8,0	7,4	63%	Se señala su excesiva dificultad, atribuible tanto al reactivo como a la tarea de depuración que se exige
ÍTEM 24	7,0	6,7	8,0	8,0	74%	No se acepta el estilo de respuesta textual
ÍTEM 25	7,0	6,9	9,0	8,6	95%	OK

Decisiones Ítems 21-25 (Concepto computacional medido: ‘Condicional compuesto (If/else)’)

- ✓ Los ítems 21 y 25 se mantienen sin cambios
- ✓ Los ítems 22 y 24 se adaptan a un estilo de respuesta visual por bloques (el ítem 24 además se pone en modo ‘depuración’ para reequilibrar la presencia de este grupo de ítems)
- ✓ Se elimina el ítem 23 por su bajo índice de aceptación (<< 75%)

Tabla 6.10. Resumen de resultados y decisiones para los ítems 26-30

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 26	4,0	4,4	8,0	7,9	95%	OK. Aunque todas las opciones podrían cumplir el enunciado, pero sólo una de manera exacta.
ÍTEM 27	6,0	5,8	8,0	7,6	79%	No se admite el estilo de respuesta textual
ÍTEM 28	8,0	7,2	9,0	8,0	74%	Excesivamente difícil por la gran extensión del código a depurar
ÍTEM 29	6,0	5,7	8,0	7,5	79%	No se admite el estilo de respuesta textual
ÍTEM 30	7,0	6,9	8,0	7,9	84%	Incluir alguna explicación adicional sobre el número desconocido '?' de fresas

Decisiones Ítems 26-30 (Concepto computacional medido: 'Mientras que (While)')

- ✓ El ítem 26 se mantiene sin cambios
- ✓ Los ítems 27 y 29 se adaptan a un estilo visual por bloques.
- ✓ Se elimina el ítem 28 por su bajo índice de aceptación (< 75%)
- ✓ El ítem 30 se mantiene sin cambios (sólo debe añadirse una instrucción adicional sobre el número desconocido de fresas '?' que debe comer el fantasma)

Tabla 6.11. Resumen de resultados y decisiones para los ítems 31-35

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 31	8,0	7,6	9,0	8,3	84%	Dar pista de la forma geométrica que dibuja la función
ÍTEM 32	7,0	7,4	9,0	8,7	89%	Dar pista de la forma geométrica que dibuja la función
ÍTEM 33	9,0	8,3	8,0	7,9	63%	Excesiva dificultad del algoritmo que dibuja un círculo
ÍTEM 34	8,0	7,4	9,0	8,7	95%	OK
ÍTEM 35	8,0	7,3	9,0	8,6	89%	OK

Decisiones Ítems 31-35 (Concepto computacional medido: 'Funciones simples')

- ✓ En los ítems 31 y 32 se da una pista adicional de la forma geométrica que dibuja la función 'my function'. Y se añade clave visual acerca del sentido inicial del lápiz.
- ✓ Se elimina el ítem 33 por su excesiva dificultad (índice de aceptación << 75%)
- ✓ Los ítems 34 y 35 se mantienen sin cambios

Tabla 6.12. Resumen de resultados y decisiones para los ítems 36-40

Número de ítem	Dificultad percibida para la población objetivo (De 1 a 10)		Pertinencia percibida para medir el PC (De 1 a 10)		Índice de aceptación (De 0% a 100%)	Observaciones y sugerencias abiertas
	Mediana	Media	Mediana	Media		
ÍTEM 36	9,0	7,7	9,0	8,6	84%	Excesiva complejidad de las 'funciones con parámetros' para la población objetivo
ÍTEM 37	8,0	7,5	9,0	8,7	79%	
ÍTEM 38	9,0	8,2	9,0	8,3	74%	
ÍTEM 39	9,0	8,8	9,0	8,4	63%	
ÍTEM 40	10,0	8,8	9,0	8,3	74%	

Decisiones Ítems 36-40 (Concepto computacional medido: 'Funciones con parámetros')

- ✓ Este grupo de ítems se elimina por su excesiva dificultad

La fiabilidad inter-jueces (*'inter-rater reliability'*) mostrada por los 19 expertos que completaron la valoración de los 40 ítems candidatos fue: $\alpha = 0,968^{**}$ ($p_{(\alpha)} = 0,000 < 0,01$) para la 'dificultad percibida' de los ítems; $\alpha = 0,294^*$ ($p_{(\alpha)} = 0,050 \leq 0,05$) para la 'pertinencia percibida para medir el PC' de los ítems; y $\alpha = 0,467^{**}$ ($p_{(\alpha)} = 0,001 < 0,01$) para la 'aceptación' de los ítems para formar parte de la versión final del TPC. Todos los 'alfa' fueron calculados como coeficientes de correlación intraclase, utilizando el modelo aleatorio bidireccional tipo consistencia, según sugiere Gwet (2014).

Las tablas anteriores se representan de manera gráfica y sintetizada en las siguientes figuras. Así, el nivel de dificultad, en relación con la población objetivo (12 y 13 años $\approx 1^\circ$ y 2° ESO), percibido por los expertos para cada ítem, se muestra en la Figura 6.8. La pertinencia para medir el PC percibida por los expertos para cada ítem, se muestra en la Figura 6.9. Y la apreciación sobre si incluir o no cada ítem en la versión final del TPC, expresado como 'índice de aceptación', se muestra en la Figura 6.10. Del análisis de estos resultados, podemos afirmar que:

- El TPC (*versión 1.0*) presenta una dificultad percibida creciente a lo largo de sus 40 ítems, algo a priori recomendado para cualquier instrumento que pretenda medir aptitudes o habilidades, cubriendo el rango completo de dificultad percibida (de *1=muy fácil* a *10=muy difícil*).
- Los ítems muestran un alto 'índice de aceptación' por los expertos, alrededor del 80-90%, para ser incluidos en la versión final del test; aunque con un pequeño descenso hacia el último grupo de ítems, quizás debido a su excesiva dificultad para la población objetivo.
- Los ítems muestran igualmente un alto nivel de pertinencia percibida para medir el PC, incluso con un ligero incremento en el último grupo de ítems (que mide el concepto computacional de mayor complejidad: las 'funciones con parámetros'), lo cual aconseja conservar estos ítems para una posible versión futura del instrumento destinada a una población de mayor edad.

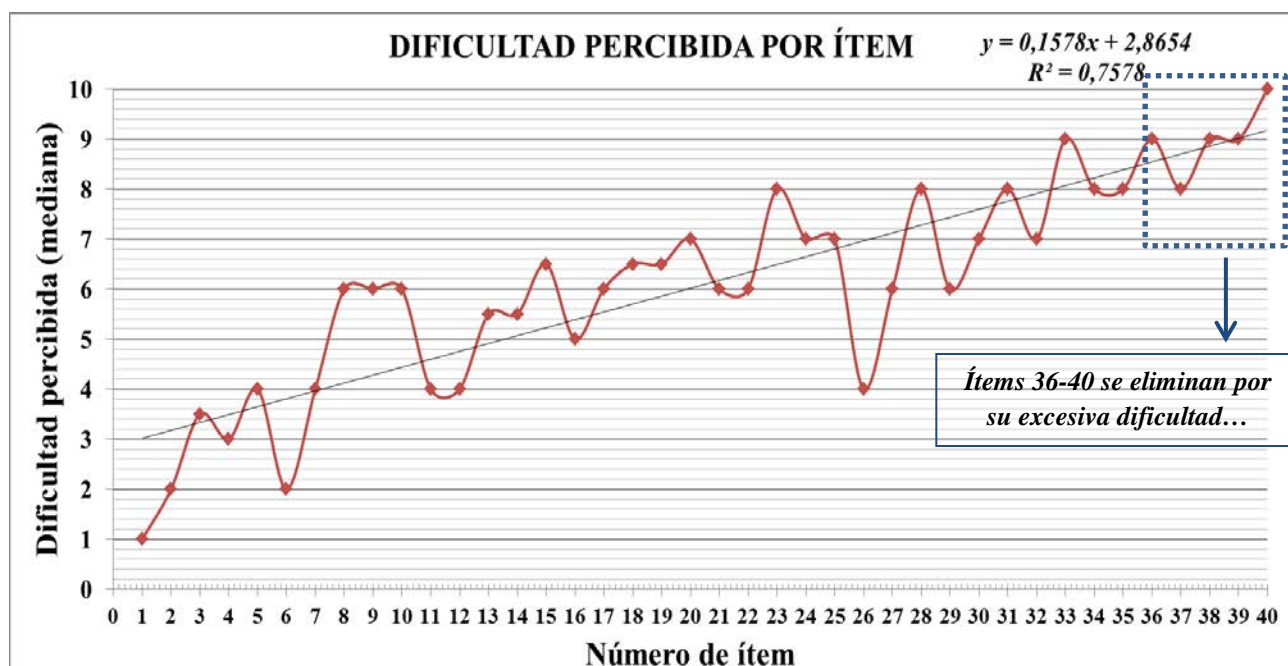


Figura 6.8. Nivel de dificultad percibido por los expertos para cada ítem

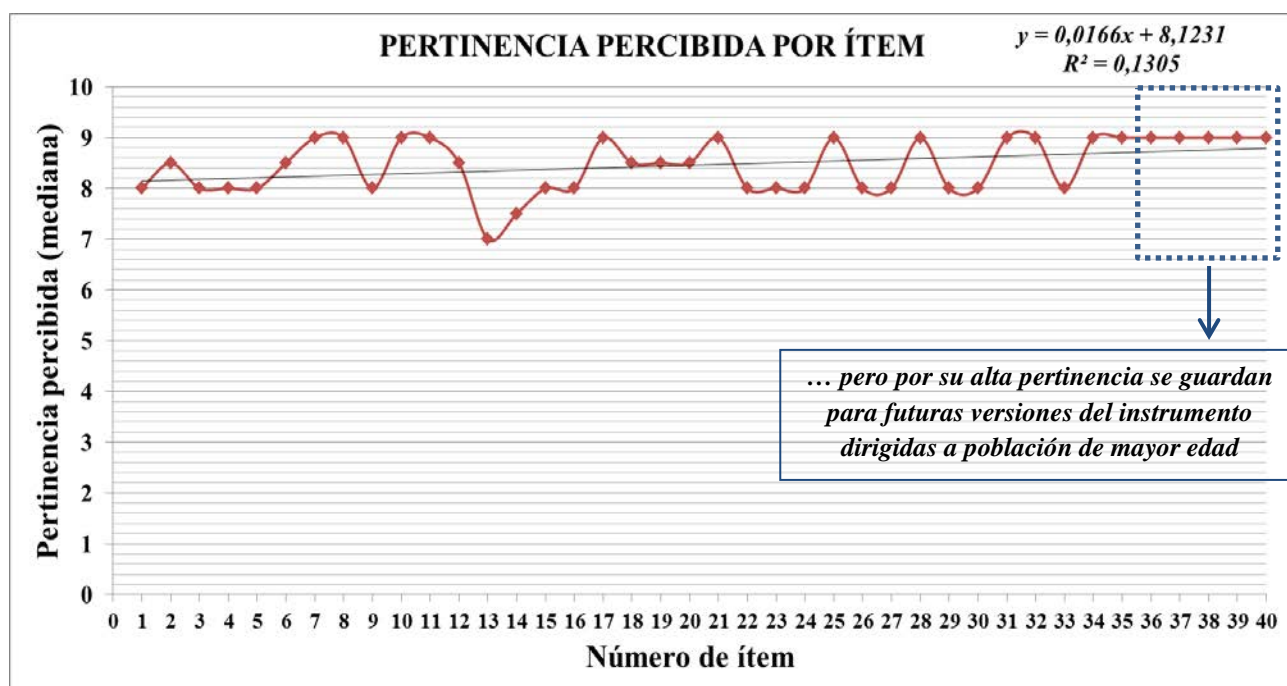


Figura 6.9. Pertinencia para medir el PC percibido por los expertos para cada ítem

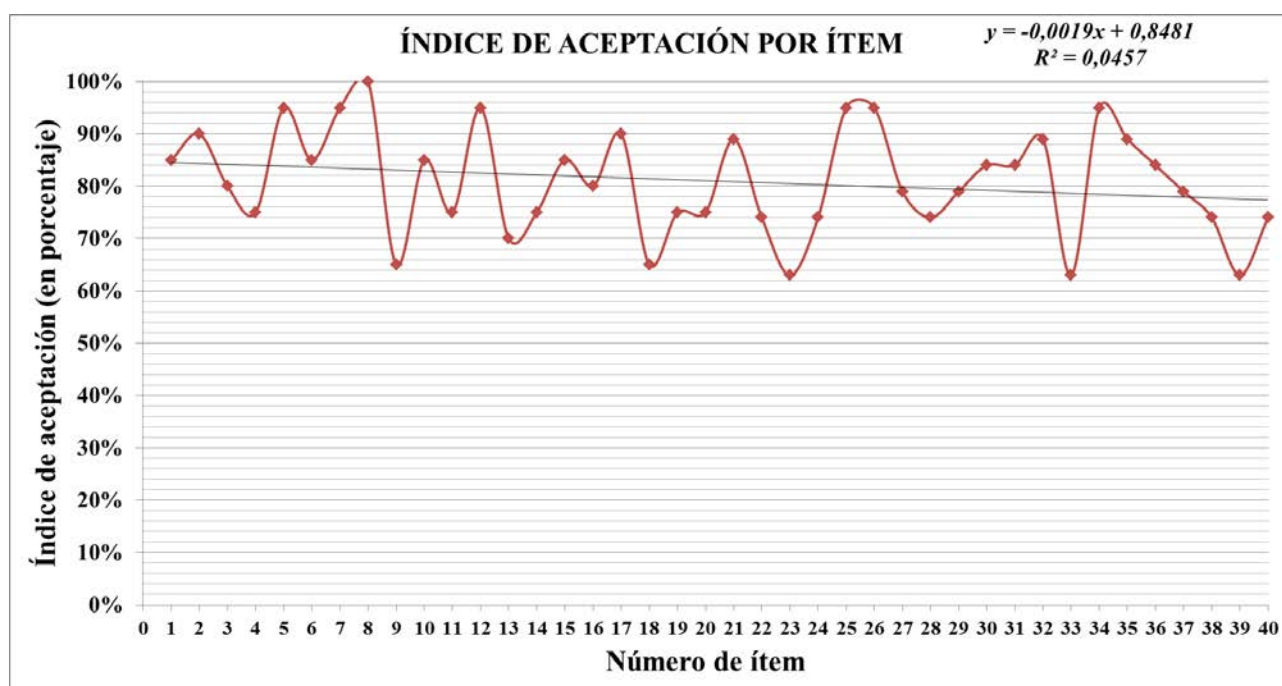


Figura 6.10. 'Índice de aceptación' por ítem

En síntesis, tomando en consideración las sugerencias y comentarios de los expertos acerca de los diferentes ítems y de los ejes de diseño del instrumento, se toman las siguientes decisiones para refinar esta versión inicial del TPC (*versión 1.0*):

- Eliminar el último grupo de 5 ítems, correspondiente al concepto computacional de 'funciones con parámetros', por su excesiva complejidad para la población objetivo.

- De los 35 ítems restantes, 7 grupos de 5 ítems (cada grupo se ocupa de abordar explícitamente un concepto computacional como se ilustró en la Figura 6.2), eliminamos el ítem de cada grupo con menor índice de aceptación por los expertos.
- Prescindir del estilo ‘textual’ en las alternativas de respuesta, dado que los estilos visuales (‘por flechas’ o ‘por bloques’) se consideran más apropiados para la población objetivo.
- Reformular los ítems que requieren ‘depurar’ (‘*debug*’) una secuencia errónea de comandos-órdenes: se mostrará el código solamente una vez, y sobre el mismo se indicarán los 4 posibles pasos en los que puede encontrarse el error.
- Incluir una pista visual acerca de la dirección y sentido inicial del trazo en la interfaz de ‘El Lienzo’.
- Introducir el anidamiento sólo después de haber acometido el concepto computacional correspondiente sin anidamiento; y eliminar los anidamientos excesivos (anidamientos dobles y triples).
- Incluir al comienzo del test unas breves instrucciones y tres ejemplos que sirvan para que los estudiantes se familiaricen con el entorno de trabajo.

En la siguiente Figura 6.11 quedan ilustrados los ítems de la versión inicial del TPC (*versión 1.0*) que se eliminan para la versión final del instrumento (*versión 2.0*)

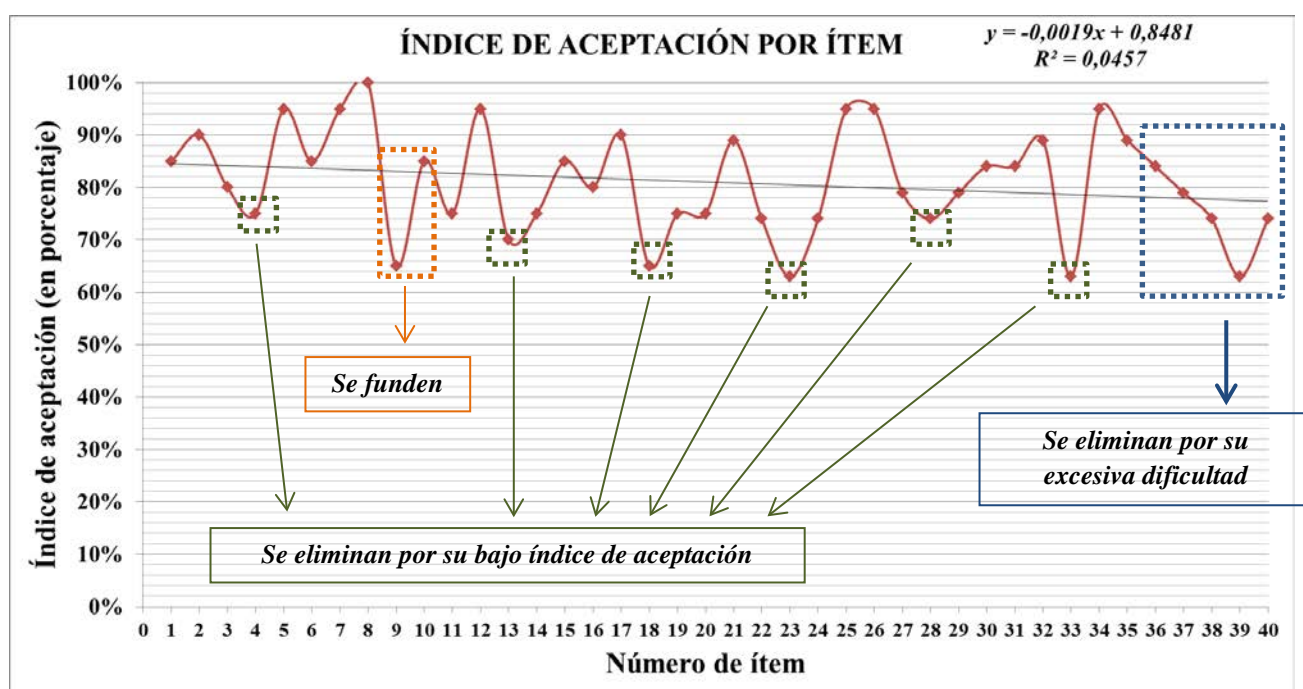


Figura 6.11. ‘Índice de aceptación’ por ítem (+ ítems que son eliminados tras el proceso de validación de contenido)

Aplicando las decisiones tomadas, en la siguiente Figura 6.12 se muestra cómo un ítem de la versión inicial del TPC (ítem 9, ver antes Figura 6.4) queda diseñado definitivamente en la versión final del instrumento:

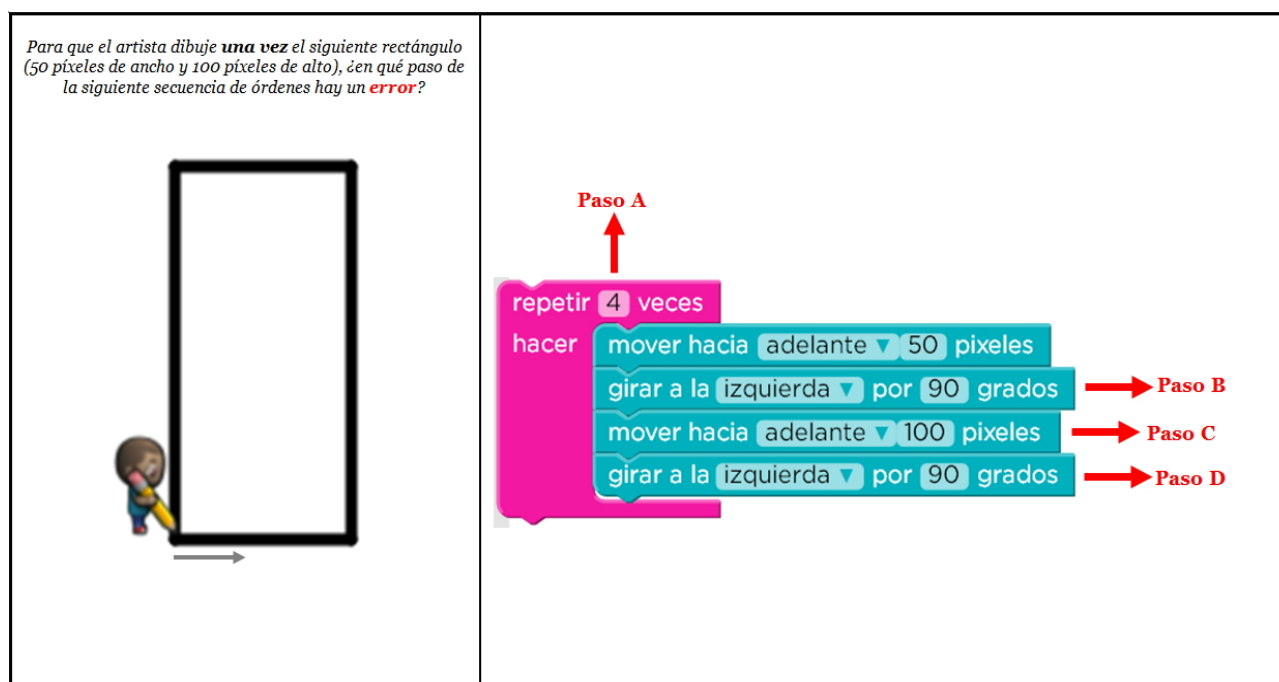


Figura 6.12. Revisión del ítem 9 de la versión original del TPC (ver anterior Figura 6.4), para la versión final del TPC

Como resultado de todo este procedimiento de depuración, se desarrolla la versión final del TPC (versión 2.0³⁶¹, de noviembre de 2014) (Román-González, 2014h), de 28 ítems de longitud; que sigue, por tanto, los siguientes principios (Román-González, Pérez-González, & Jiménez-Fernández, 2015):

- **Objetivo:** el TPC pretende medir el nivel de desarrollo del pensamiento computacional en el sujeto.
- **Definición operativa del constructo medido:** el pensamiento computacional es la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación, y usando la lógica inherente a los lenguajes informáticos de programación: secuencias o direcciones básicas, bucles, condicionales, funciones, y variables.
- **Población objetivo:** el TPC está diseñado y dirigido, como foco central, a población escolar española de entre 12 y 13 años (1º y 2º ESO)
- **Tipo de instrumento:** prueba objetiva de elección múltiple con 4 opciones de respuesta (sólo 1 correcta).
- **Longitud:** 28 ítems.
- **Tiempo máximo de realización:** 45 minutos.

Cada uno de los 28 ítems está diseñado y caracterizado en los siguientes cinco ejes de diseño:

- **Concepto computacional abordado:** cada ítem aborda uno o más de los siguientes 7 conceptos computacionales, ordenados en dificultad creciente: Direcciones o secuencias

³⁶¹ Disponible como documento PDF en <https://db.tt/6hg2seLu>

básicas (4 ítems); Bucles–‘repetir veces’ (4 ítems); Bucles–‘repetir hasta’ (4 ítems); Condicional simple–‘if’ (4 ítems); Condicional compuesto–‘if/else’ (4 ítems); Mientras que–‘while’ (4 ítems); Funciones simples (4 ítems)

- **Entorno-Interfaz del ítem:** los ítems del TPC se presentan en alguno de los siguientes dos entornos gráficos o interfaces: ‘El Laberinto’ (23 ítems); ‘El Lienzo’ (5 ítems)
- **Estilo de las alternativas de respuesta:** en cada ítem, las alternativas de respuesta se pueden presentar en alguno de estos dos estilos: ‘Visual por flechas’ (8 ítems); ‘Visual por bloques’ (20 ítems)
- **Existencia o inexistencia de anidamiento:** dependiendo de si la solución del ítem involucra una secuencia de comandos-órdenes con (19 ítems) o sin (9 ítems) conceptos computacionales anidados (un concepto embebido en otro concepto en un orden de jerarquía superior)
- **Tarea requerida:** dependiendo de cuál de las siguientes tareas cognitivas es necesaria para la resolución del ítem: ‘Secuenciación’, enunciar de manera ordenada una serie de comandos-órdenes (14 ítems); ‘Completamiento’, completar un conjunto incompleto de comandos previamente dado (9 ítems); ‘Depuración’, depurar (‘*debug*’) un conjunto incorrecto de comandos previamente dado (5 ítems).

En la Figura 6.13 se muestran sintéticamente las especificaciones de los 28 ítems de esta versión final del TPC, que son caracterizados en los 5 ejes señalados. Y en el [Anexo F](#) se aportan en detalle los 28 ítems del TPC (*versión 2.0*), así como la tabla de especificaciones para cada uno de ellos.

	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	Opción correcta
			Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Item 1	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Secuenciación	B
Item 2	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Completamiento	C
Item 3	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Depuración	D
Item 4	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Secuenciación	D
Item 5	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Secuenciación	C
Item 6	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Completamiento	D
Item 7	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	No	Depuración	A
Item 8	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación	B
Item 9	Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No	No	Secuenciación	D
Item 10	Laberinto	Visual por bloques	Sí	No	Sí	No	No	No	No	No	No	Completamiento	C
Item 11	Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	Sí	Depuración	C
Item 12	Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	Sí	Secuenciación	A
Item 13	Laberinto	Visual por flechas	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación	B
Item 14	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación	A
Item 15	Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	Sí	Completamiento	D
Item 16	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Depuración	D
Item 17	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación	B
Item 18	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación	A
Item 19	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Depuración	B
Item 20	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento	C
Item 21	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación	A
Item 22	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación	B
Item 23	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento	A
Item 24	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento	C
Item 25	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación	B
Item 26	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento	B
Item 27	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación	A
Item 28	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento	C

Figura 6.13. Cuadro resumen de especificaciones de los 28 ítems del TPC (versión 2.0) en los 5 ejes de diseño

6.4. Fase II: Virtualización y validación aparente

Para la virtualización del TPC (*versión 2.0*), esto es, su digitalización como formulario en línea a través de la tecnología de *Google Drive Forms*, se realiza una prueba previa de virtualización³⁶² (Figura 6.14) sobre el ítem 8.



Figura 6.14. Código QR de enlace al formulario de prueba de virtualización del TPC

Tal y como puede verse, en dicha prueba de virtualización se presenta el ítem 8 de tres maneras diferentes:

- **Modo A:** se virtualiza el ítem a partir de una sola imagen; que contiene el reactivo en la parte izquierda de la misma, y las opciones de respuesta en la parte derecha. El sujeto puede ampliar la imagen con el teclado ‘Ctrl+’
- **Modo B:** se virtualiza el ítem a partir de dos imágenes; una que contiene el reactivo y, a continuación, otra que contiene las opciones de respuesta. El sujeto puede ampliar la imagen con el teclado ‘Ctrl+’
- **Modo C:** se virtualiza el ítem enlazándolo a un documento PDF que contiene el mismo, y que se abre en otra ventana.

Consultados los mismos expertos que participaron en la validación de contenido, por correo electrónico, se recibe una respuesta unánime alrededor del ‘Modo A’ como el más adecuado para la virtualización de los ítems de nuestro instrumento. Se procede pues a una primera virtualización del TPC (*versión 2.0*), siguiendo las siguientes pautas:

- ✓ Se virtualizan los 28 ítems del TPC (*versión 2.0*) según el ‘Modo A’; distribuidos en 7 páginas diferentes, con 4 ítems en cada una ellas (por lo tanto, en cada página aparecen los 4 ítems correspondientes a un ‘concepto computacional’ específico).
- ✓ Se añade una página inicial, en la cual: se da la bienvenida al test al sujeto; se le solicitan al sujeto sus datos personales (‘Nombre’, ‘Apellidos’, ‘Sexo’, ‘Centro Educativo’, y ‘Curso Académico’); se incluyen unas breves instrucciones y 3 ejemplos, que se detallan en el **Anexo G**.
- ✓ A sugerencia de nuestros directores de tesis, al final de la última página se añade también un par de preguntas sobre percepción de autoeficacia (escalas tipo Likert, desde ‘0=*Pésimo*’ hasta ‘10=*Excelente*’), que se detallan igualmente al final del **Anexo G**.

³⁶² Accesible en <https://goo.gl/FMKXzr>

Dada esta primera virtualización del TPC (*versión 2.0*), y cubriendo los objetivos específicos de investigación O_{3c} y O_{3d} , se procede a una prueba de validez aparente el día 25 de noviembre de 2014 en el Colegio Virgen de Europa³⁶³ (Boadilla del Monte, Madrid), en la que participan 22 estudiantes de 1º ESO (13 chicos y 9 chicas). De dicha prueba de validez aparente, a la que este doctorando asiste personalmente y toma notas de observación, se extraen las siguientes conclusiones y decisiones:

- La virtualización del TPC es accesible y navegable por los estudiantes a través de sus ordenadores portátiles.
- Las respuestas emitidas por los sujetos quedan correctamente recogidas en nuestra base de datos.
- El tiempo máximo fijado (45 minutos) es más que suficiente para la realización del test (el 100% de sujetos lo finaliza antes del tiempo límite)
- Las instrucciones dadas en esta primera virtualización del TPC (**Anexo G**) resultan algo insuficientes para los sujetos, que requieren al aplicador de varias aclaraciones orales. Atendiendo a las dudas expresadas por los sujetos, se decide incluir adicionalmente las siguientes instrucciones para la virtualización definitiva del instrumento:
 - ✓ En las instrucciones relativas al ‘EJEMPLO I’ se añade el siguiente párrafo intermedio: *“Es decir, llevar a ‘Pac-Man’ EXACTAMENTE a la casilla en la que se encuentra el fantasma (sin pasarse ni quedarse corto), y siguiendo estrictamente el camino señalado en amarillo (sin salirse y sin tocar las paredes, representadas por los cuadrados anaranjados)”*
 - ✓ En las instrucciones relativas al ‘EJEMPLO II’ se añade el siguiente párrafo intermedio: *“Te recordamos que la pregunta te pide llevar a ‘Pac-Man’ EXACTAMENTE a la casilla en la que se encuentra el fantasma (sin pasarse ni quedarse corto), y siguiendo estrictamente el camino señalado en amarillo (sin salirse y sin tocar las paredes, representadas por los cuadrados anaranjados)”*
 - ✓ En las instrucciones relativas al ‘EJEMPLO III’ se añade el siguiente párrafo intermedio: *“La orden MOVER empuja el lápiz dibujando, mientras que la orden SALTAR hace pegar un salto al artista sin dibujar. La flecha gris indica la dirección del primer movimiento del lápiz”*

Aplicando las decisiones anteriores, se procede a la virtualización y puesta en servicio definitivas del TPC³⁶⁴ (Román-González, 2014i), accesible desde cualquier ordenador o dispositivo móvil con conexión a Internet; que comienza a ser aplicado en amplias muestras de la población objetivo, tal y como se describe en el siguiente apartado.

³⁶³ <http://www.cve.edu.es/>

³⁶⁴ Accesible desde <http://goo.gl/IYEKMB>

6.5. Fase III: Puesta en servicio y aplicación

La Fase III consiste en la puesta en servicio definitiva del TPC (Román-González, 2014i), y su aplicación en amplias muestras de la población objetivo. A partir de dicha aplicación podrá estudiarse la psicometría general del instrumento, así como su fiabilidad y validez.

6.5.1. Método

6.5.1.1. Muestra

Para abordar el objetivo específico de investigación O_{3e} , el TPC (Román-González, 2014i) es aplicado sobre una muestra total de tamaño igual a 1.251 sujetos ($N = 1.251$), sin experiencia formal previa en tareas de programación. La distribución de la muestra según el sexo, el ciclo educativo y el curso académico de los sujetos, se presenta en las Tablas 6.13, 6.14, y 6.15.

Tabla 6.13. Distribución por sexo de la muestra total de aplicación del TPC ($N = 1.251$ sujetos)

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Chico	730	58,4	58,4	58,4
	Chica	521	41,6	41,6	100,0
	Total	1251	100,0	100,0	

Tabla 6.14. Distribución por ciclo educativo de la muestra total de aplicación del TPC ($N = 1.251$ sujetos)

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	3º Ciclo de Educación Primaria	176	14,1	14,1	14,1
	1º Ciclo de Educación Secundaria	735	58,8	58,8	72,8
	2º Ciclo de Educación Secundaria	340	27,2	27,2	100,0
	Total	1251	100,0	100,0	

Tabla 6.15. Distribución por curso académico de la muestra total de aplicación del TPC ($N = 1.251$ sujetos)

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	5º Primaria	103	8,2	8,2	8,2
	6º Primaria	73	5,8	5,8	14,1
	1º ESO	433	34,6	34,6	48,7
	2º ESO	302	24,1	24,1	72,8
	3º ESO	199	15,9	15,9	88,7
	4º ESO	141	11,3	11,3	100,0
	Total	1251	100,0	100,0	

Tal y como puede observarse en las tablas anteriores, el grueso de la muestra se corresponde con la población objetivo del TPC (1º-2º de la ESO: 58,8%); si bien se complementa con muestra proveniente de 3º Ciclo de Primaria (5º y 6º Primaria: 14,1%), y 2º ciclo de Secundaria (3º y 4º ESO: 27,2%), lo cual nos permitirá estudiar el suelo y el techo del instrumento.

También se observa en la composición de la muestra un desequilibrio de género a favor de los chicos (58,4%) frente a las chicas (41,6%). Este hecho se explica dado que, a diferencia del evento 'HoC' que fue aplicado en todo tipo de aulas, el TPC se aplica fundamentalmente en la clase optativa de Informática (en el caso de la ESO). Tómese, por tanto, este desequilibrio de género observado en

nuestra muestra como un indicador de la elección diferencial que cada sexo hace actualmente de la asignatura optativa de Informática.

Adicionalmente, se registra el tipo de dispositivo a través del cual el TPC es aplicado ‘*on-line*’ a los sujetos. Tal y como se detalla en la siguiente Tabla 6.16, el 80,0% de los sujetos realizaron el test a través de un ordenador PC’s (ya fuera un equipo ‘fijo’ o ‘portátil’), mientras que el 20,0% restante lo hizo a través de un dispositivo móvil tipo ‘*tablet*’.

Tabla 6.16. Distribución del total de la muestra según el dispositivo de aplicación

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Ordenador	1001	80,0	80,0	80,0
	Tableta	250	20,0	20,0	100,0
	Total	1251	100,0	100,0	

Los 1.251 sujetos que conforman el total de la muestra provienen de 24 centros educativos diferentes (Tabla 6.17, con los centros ordenados de mayor a menor aportación de sujetos). Es decir, la aportación media de sujetos por centro a la validación del instrumento es de 52,13 sujetos \approx 2 grupos de clase.

Tabla 6.17. Centro Educativo de procedencia de los sujetos de la muestra (ordenados de mayor a menor aportación)

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válidos	Colegio Mirasur	202	16,1	16,1	16,1
	IES Politécnico de Castellón	109	8,7	8,7	24,9
	IES Sevilla Este	94	7,5	7,5	32,4
	IES Vicente Aleixandre	85	6,8	6,8	39,2
	Colegio Virgen de Europa	72	5,8	5,8	44,9
	Escola La Masía	67	5,4	5,4	50,3
	CEIP Lope de Vega	61	4,9	4,9	55,2
	IES María Blasco	60	4,8	4,8	60,0
	IES Camp de Morvedre	57	4,6	4,6	64,5
	IES Fuente de San Luis	56	4,5	4,5	69,0
	IES Penyalgosa	49	3,9	3,9	72,9
	Yago School	48	3,8	3,8	76,7
	IES Dr. Lluís Simarro Lacabra	42	3,4	3,4	80,1
	IES Juan de Garay	35	2,8	2,8	82,9
	IES L’Elia	35	2,8	2,8	85,7
	IES Montserrat Roig	35	2,8	2,8	88,5
	Campus de Verano	32	2,6	2,6	91,0
	IES Andreu Sempere	28	2,2	2,2	93,3
	IES El Plá	23	1,8	1,8	95,1
	IES L’Almadrava	22	1,8	1,8	96,9
	IES Mestre Ramón Esteve	15	1,2	1,2	98,1
	IES Jordi Sant Jordi	14	1,1	1,1	99,2
	CRA Laguna de Pétrola	5	,4	,4	99,6
	Colegio San Vicente Ferrer	5	,4	,4	100,0
Total	1251	100,0	100,0		

En la Tabla 6.18 se muestran de nuevo los 24 centros educativos que aportan sujetos al proceso de validación del TPC, agrupados en función del procedimiento de muestreo utilizado en cada caso. Así, podemos diferenciar:

- Los centros del **procedimiento A** son muestreados intencionalmente en su calidad de centros participantes en el programa-curso trimestral ‘K-8 Intro to Computer Science’ (Code.org, 2015a) de la fundación Code.org. Como parte de la evaluación de este programa, que abordamos en el Capítulo 7, el TPC es aplicado en situación *pre-test* y *post-test*. Utilizaremos exclusivamente los resultados de la aplicación en situación *pre-test* a efectos de la baremación del instrumento.
- Análogamente, los centros del **procedimiento B** son muestreados intencionalmente en su calidad de centros participantes en el programa-curso trimestral ‘K-5 Computer Science Fundamentals (Course 2)’ (Code.org, 2015b), también de la fundación Code.org y que supone una versión simplificada del curso K-8 para ser aplicada en el último ciclo de Primaria. Como parte de la evaluación de este programa, que abordamos en el Capítulo 7, el TPC es aplicado en situación *pre-test* y *post-test*. Utilizaremos exclusivamente los resultados de la aplicación en situación *pre-test* a efectos de la baremación del instrumento.
- Los centros del **procedimiento C** son muestreados intencionalmente en su calidad de centros participantes en el estudio de validación convergente entre el TPC y la herramienta Dr. Scratch³⁶⁵ (Moreno-León & Robles, 2015; Moreno-León, Robles, & Román-González, 2015), una aplicación web que analiza automáticamente la calidad del código, en términos de pensamiento computacional, de proyectos hechos con Scratch³⁶⁶. Dicho estudio es detallado en el sub-epígrafe 6.5.2.3.3.2. Baste señalar por el momento que, en este grupo de centros: el TPC se aplica inicialmente en situación *pre-test*; posteriormente los sujetos inician un periodo de aprendizaje con Scratch (alrededor de 8 semanas); y finalmente los sujetos realizan una segunda aplicación del TPC en situación *post-test*, además de analizar sus proyectos finales de Scratch con la herramienta analítica Dr. Scratch. Sólo las aplicaciones *pre-test* del TPC son utilizadas a efectos de la baremación del instrumento.
- Finalmente, el **procedimiento D** incluye a centros que acudieron ‘*motu proprio*’ a este doctorando, solicitando aplicar el TPC sobre sus estudiantes para conocer su nivel de pensamiento computacional. De entre este grupo de centros, cabe destacar el Colegio Mirasur (Madrid), que se ofreció a aplicar masivamente el instrumento a sus estudiantes en todo el rango que va desde 5º Primaria a 4º ESO; por lo cual, aprovechamos para aplicar allí una selección de tareas Bebras (Dagiene & Stupuriene, 2014), agrupadas en dificultad creciente según el ciclo educativo, que será la base del estudio de validez convergente que se expondrá en el sub-epígrafe 6.5.2.3.3.1.

Además, en la Tabla 6.18 se refiere la ‘provincia’ y ‘titularidad’ de cada uno de los 24 centros; así como otros instrumentos de medición aplicados en los mismos, que se explicarán con más detalle en el siguiente sub-epígrafe 6.5.1.2.

³⁶⁵ <http://drscratch.org/>

³⁶⁶ <https://scratch.mit.edu/>

Tabla 6.18. Centros Educativos participantes en la validación del TPC, agrupados según el procedimiento de muestreo

	Sujetos	Provincia	Titularidad	Procedimiento de muestreo	Otros instrumentos aplicados	
Escola La Masía	67	Valencia	Privado	A	BFQ-NA ³⁶⁷	
IES María Blasco	60	Alicante	Público		RP30 ³⁶⁸	
IES Camp de Morvedre	57	Valencia	Público		Batería PMA ³⁶⁹	
IES Penyagolosa	49	Castellón	Público			
IES Dr. Lluís Simarro Lacabra	42	Valencia	Público			
IES Juan de Garay	35	Valencia	Público			
IES L'Elia	35	Valencia	Público			FI-R ³⁷⁰
IES Andreu Sempere	28	Alicante	Público			-
IES El Plá	23	Alicante	Público		Batería PMA	
IES L'Almadrava	22	Alicante	Público			
Subtotal Procedimiento A	N_A = 418					
Yago School	48	Sevilla	Privado	B	BFQ-NA	
Subtotal Procedimiento B	N_B = 48					
IES Sevilla Este	94	Sevilla	Público	C	Dr. Scratch	
IES Vicente Aleixandre	85	Sevilla	Público		Dr. Scratch y Bebras ³⁷¹	
IES Mestre Ramón Esteve	15	Valencia	Público		Dr. Scratch	
Subtotal Procedimiento C	N_C = 194					
Colegio Mirasur	202	Madrid	Privado	D	Bebras	
IES Politécnico de Castellón	109	Castellón	Público			
Colegio Virgen de Europa	72	Madrid	Privado		-	
CEIP Lope de Vega	61	Madrid	Público			
IES Fuente de San Luis	56	Valencia	Público			
IES Montserrat Roig	35	Alicante	Público			
Campus de Verano ³⁷²	32	Asturias	Privado			
IES Jordi Sant Jordi	14	Valencia	Público			
CRA Laguna de Pétrola	5	Albacete	Público			
Colegio San Vicente Ferrer	5	Valencia	Concertado			
Subtotal Procedimiento D	N_D = 591					
Total	N = 1251					

Así pues, del total de la muestra (N = 1.251): 825 sujetos (65,9%) estudian en centros públicos, y 426 (34,1%) en centros privados o concertados. Con respecto a la provincia de procedencia de los sujetos, de mayor a menor aportación a la muestra: Madrid (335 sujetos), Valencia (326), Sevilla (227), Alicante (168), Castellón (158), Asturias (32), y Albacete (5)

³⁶⁷ BFQ-NA: Cuestionario "Big Five" de Personalidad para Niños y Adolescentes [<http://web.teaediciones.com/bfq-na-cuestionario-%E2%80%9Cbig-five%E2%80%9D-de-personalidad-para-ni%C3%B1os-y-adolescentes.aspx>]

³⁶⁸ RP30: Resolución de Problemas [<http://web.teaediciones.com/rp30-resolucion-de-problemas.aspx>]

³⁶⁹ PMA: Batería de Aptitudes Mentales Primarias [<http://web.teaediciones.com/pma-aptitudes-mentales-primarias.aspx>]

³⁷⁰ FI-R: Formas Idénticas-R [<http://web.teaediciones.com/Formas-Identicas---R.aspx>]

³⁷¹ Se revisó en el sub-epígrafe 3.3.2.1 (Capítulo 3) de la parte teórica

³⁷² Campamento de Verano privado celebrado en el mes de julio de 2015

6.5.1.2. *Instrumentos*

Tal y como sabemos, el instrumento principal sobre el que se focaliza este capítulo es nuestro Test de Pensamiento Computacional (TPC) en su versión virtualizada-digitalizada final de 28 ítems de longitud (Figura 6.15) (Román-González, 2014i)



Figura 6.15 Código QR de enlace a la versión final del TPC (Román-González, 2014i)

Para la validación criterial (concurrente) del TPC, se utilizan adicionalmente los siguientes instrumentos estandarizados; todos ellos publicados por TEA Ediciones (Madrid), y con posibilidad de ser aplicados ‘on-line’:

- **Batería de Aptitudes Mentales Primarias (PMA):** está desarrollada por el departamento de I+D+i de TEA Ediciones. Su objetivo es la apreciación de factores cognitivos básicos que permiten una estimación de la inteligencia general. Es un instrumento de aplicación individual o colectiva, con un tiempo máximo de aplicación de 26 minutos, y que puede ser utilizado a partir de 10 años de edad (TEA Ediciones, 2015a, en línea)

La estructura de la batería PMA es reflejo de las investigaciones realizadas por L.L.Thurstone (1938a), aportación decisiva al campo psicométrico mediante la aplicación del análisis factorial. El resultado de esos estudios sigue aún vigente y se materializa en la medición precisa de los siguientes factores (cada factor se mide por un test diferente en la batería):

- **Factor V o verbal (PMA-V):** Capacidad para comprender y expresar ideas con palabras. El manual técnico de la batería (Departamento I+D; TEA Ediciones, 2007) informa de excelentes niveles de fiabilidad para el PMA-V ($r_{xx} = +0,91$; por el procedimiento de ‘dos mitades’), así como de su validez convergente respecto, por ejemplo, al test DAT³⁷³-Verbal ($r_{xy} = +0,61$)
- **Factor E o espacial (PMA-E):** Capacidad para imaginar y concebir objetos en dos y tres dimensiones. El manual informa niveles de fiabilidad aceptables del PMA-E ($r_{xx} = +0,73$), así como validez convergente con el Test de Rotación de Figuras ($r_{xy} = +0,57$)
- **Factor R o razonamiento (PMA-R):** Capacidad para resolver problemas lógicos, comprender y planear. El manual reporta una excelente fiabilidad del PMA-R ($r_{xx} = +0,92$), así como validez convergente con el Test de Dominós D-48³⁷⁴ ($r_{xy} = +0,84$)
- **Factor N o numérico (PMA-N):** Capacidad para manejar números y conceptos cuantitativos. El manual reporta una excelente fiabilidad del PMA-N ($r_{xx} = +0,99$), así como su validez convergente respecto, por ejemplo, al DAT-Numérico ($r_{xy} = +0,51$).

³⁷³ DAT: *Differential aptitude test* (Test de Aptitudes Diferenciales, diseñado por G.K. Bennett y colaboradores)

³⁷⁴ D-48: Test de Dominós, diseñado por E. Anstey y P. Picho, es una medida de inteligencia general fluida (‘factor g’)


En la siguiente Figura 6.16 se muestra un ejemplo de ítem relativo a cada factor. Así, en la parte superior de la figura, vemos un ítem relativo al Factor V o verbal; en la parte media de la figura, vemos un ítem relativo al Factor E o espacial; en la parte inferior izquierda, vemos un ítem relativo al Factor R o razonamiento; y en la parte inferior derecha, vemos un ítem relativo al Factor N o numérico.

Pregunta 5

¿Cuál de las cuatro palabras significa lo mismo que la propuesta, escrita en mayúsculas?


HABITUAL

- Nocturno
- Circular
- Principal
- Usual



Pregunta 10

Marque las letras de todas las figuras que sean iguales al modelo presentado a la izquierda.


A
B
C
D
E
F

- A
- B
- C
- D
- E
- F

Pregunta 5

¿Qué letra continúa la serie?

a b c a b d a b e a b f

- a
- b
- c
- f
- g
- h

Pregunta 4

Compruebe mentalmente si la suma está bien o mal.

73 29 56 33 ---- 211

- Bien
- Mal




Figura 6.16. Ejemplos de ítems de la Batería PMA: arriba, ítem del PMA-V; en el centro, ítem del PMA-E; abajo a la izquierda, ítem del PMA-R; y abajo a la derecha, ítem del PMA-N

- **Test de Resolución de Problemas (RP30):** está desarrollado igualmente por el departamento de I+D+i de TEA Ediciones. Su objetivo es evaluar la rapidez y flexibilidad para realizar operaciones lógicas. Es un instrumento de aplicación individual o colectiva, con un tiempo máximo de aplicación de 17 minutos, y que puede ser utilizado en población adolescente o adulta (TEA Ediciones, 2015b, en línea). El manual técnico (Seisdedos, 2002) informa de excelentes niveles de fiabilidad del test ($r_{xx} > +0,90$; por el procedimiento de ‘dos mitades’), y evidencias de validez criterial respecto, por ejemplo, al Test Cambios³⁷⁵ de flexibilidad cognitiva ($r_{xy} = +0,38$) o el DAT-Espacial ($r_{xy} = +0,34$)

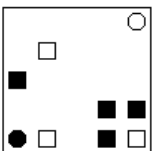
El RP30 aprecia una capacidad intelectual de tipo superior mediante la cual se captan una serie de relaciones lógicas propuestas en un problema con el fin de determinar si se cumplen dichas relaciones en otra estructura más simple. La tarea está muy relacionada con los aspectos no verbales de la inteligencia, y una buena dotación en ella parece ser un componente importante en muchas actividades (escolares o profesionales). Contiene 30 problemas con tres grados de dificultad, cada uno de los cuales contiene cinco estructuras más o menos complejas en las que hay que decidir si se cumplen las condiciones del problema. Exigen bastante concentración y se penalizan los errores. En la tarea subyacen tres componentes aptitudinales (Seisdedos, 2002):

- Razonamiento, por el hecho de captar la condición o condiciones que exige cada problema o grupo de elementos.
- Aptitud espacial, por tener que apreciar el contenido de los pequeños círculos y cuadrados de cada recuadro para decidir si se cumple o no la condición o condiciones exigidas.
- Memoria de trabajo, que permite retener la condición lógica aplicada sin necesidad de consultarla constantemente.

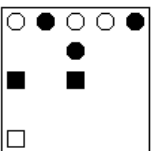
En resumen, el RP30 mide un conjunto de aptitudes que puede tener aplicación en muy diferentes comportamientos. En la Figura 6.17 se ilustra un ítem del RP30: el sujeto debe marcar qué recuadros (A, B, C, D y/o E) cumplen la condición dada en el encabezado.

Pregunta 11

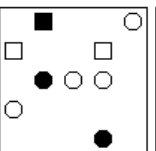
□ ≠ ● o ■ ≥ ○



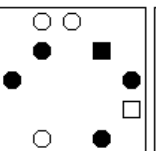
A



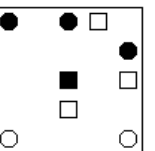
B



C



D



E

A

B

C

D

E

Figura 6.17. Ítem del Test de Resolución de Problemas RP30

³⁷⁵ CAMBIOS: Test de Flexibilidad Cognitiva, diseñado por N. Seisdedos (<http://web.teaediciones.com/cambios-test-de-flexibilidad-cognitiva.aspx>)

- **Test de Formas Idénticas – Revisadas (FI-R):** está desarrollado originalmente por L. L. Thurstone (1938b), y adaptado por el departamento de I+D+i de TEA Ediciones. Su objetivo es la evaluación de las aptitudes perceptivas y atencionales. Es un instrumento de aplicación individual o colectiva, con un tiempo máximo de aplicación de 4 minutos, y que puede ser utilizado en escolares de 9 a 12 años y adultos de 19 a 50 años (TEA Ediciones, 2015c, en línea)

El test FI-R permite evaluar las aptitudes perceptivas y atencionales de los escolares y los adultos. Está compuesto por 60 conjuntos de elementos gráficos; y se trata de una tarea muy bien aceptada por los evaluados consistente en identificar qué figura es igual a un modelo. En su edición revisada y actualizada por TEA Ediciones, se han incluido nuevas medidas de rendimiento que tienen en cuenta tanto los aciertos como los errores, lo que posibilita extraer información sobre la precisión perceptiva y el grado de impulsividad del evaluado. Así, la aplicación del test FI-R arroja dos puntuaciones: el ‘Índice de Percepción y Atención’ (A-E) y el ‘Índice de Control de la Impulsividad’ (ICI). En la Figura 6.18 se muestra un ítem del Test FI-R.

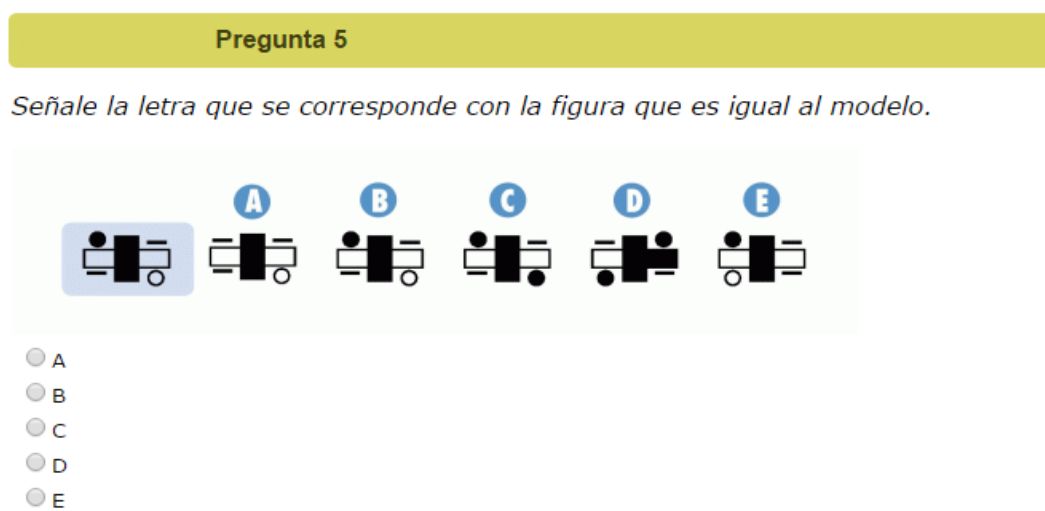


Figura 6.18. Ítem del Test de Formas Idénticas-Revisadas (FI-R)

- **Cuestionario “Big Five” de Personalidad para Niños y Adolescentes (BFQ-NA):** desarrollado originalmente por Barbaranelli, Caprara, Rabasca, & Pastorelli (2003), y adaptado posteriormente a población española por el equipo de M^a Victoria del Barrio. Su objetivo es la evaluación de las 5 dimensiones del modelo de los ‘Cinco grandes’ en la estructura de la personalidad: Conciencia, Apertura, Extraversión, Amabilidad e Inestabilidad Emocional. Es un instrumento de aplicación individual o colectiva, con un tiempo aproximado de aplicación de 30 minutos, y que puede ser utilizado en escolares de 8 a 15 años (TEA Ediciones 2015d, en línea). El BFQ-NA representa la adaptación del modelo de los ‘Cinco Grandes’ de la personalidad para niños y adolescentes. Las cinco dimensiones evaluadas son:
 - **Conciencia (Co):** incluye elementos relacionados con la autonomía, orden, precisión, perseverancia y el cumplimiento de normas y compromisos. El manual técnico (Barbaranelli, Caprara, & Rabasca, 2013) informa de una fiabilidad buena para esta

dimensión cuando la aplicación es autoinformada y la muestra está formada por sujetos entre 11 y 13 años ($\alpha = 0,83$); y una leve correlación positiva de la misma con rendimiento académico ($r = +0,13$)

- **Apertura (Ap):** incluye elementos relacionados con aspectos intelectuales, creatividad e intereses culturales. Tiene fiabilidad $\alpha = 0,83$ (aplicación autoinformada) y moderada correlación positiva con rendimiento académico ($r = +0,51$)
- **Extraversión (Ex):** relativa a la sociabilidad, actividad, entusiasmo, asertividad, autoconfianza. Tiene fiabilidad $\alpha = 0,80$ (aplicación autoinformada) y leve correlación negativa con rendimiento académico ($r = -0,13$)
- **Amabilidad (Am):** mide tendencia al apoyo y a la prosociabilidad, el grado de cooperación y la sensibilidad hacia los otros y sus necesidades. Tiene fiabilidad $\alpha = 0,83$ (aplicación autoinformada) y no correlaciona con rendimiento académico.
- **Inestabilidad Emocional (Ie):** tendencia al malestar y al neuroticismo, manifestada en los cambios de humor, tendencia a la ansiedad, la depresión, el descontento y la irritabilidad. Tiene fiabilidad $\alpha = 0,85$ (aplicación autoinformada) y no correlaciona con rendimiento académico.

El BFQ-NA está compuesto por 65 ítems que se valoran mediante una escala tipo Likert de 5 alternativas; y puede ser cumplimentado por el niño (aplicación autoinformada) o por los padres, tutores o profesores, ya que el instrumento permite que los ítems puedan ser formulados en tercera persona refiriéndose al niño. En la Figura 6.19 se muestra un ítem del cuestionario BFQ-NA.

Pregunta 3

Señala la respuesta que mejor describa con qué frecuencia eres o te comportas de ese modo.

Hago las cosas con atención y sin distraerme.

- Casi siempre
- Muchas veces
- Algunas veces
- Pocas veces
- Casi nunca

Figura 6.19. Ítem del Cuestionario “*Big Five*” de Personalidad para Niños y Adolescentes (BFQ-NA)

Para la validación convergente del TPC, se utilizan las siguientes 2 herramientas de medida del pensamiento computacional, que ya se describieron con detalle en la parte teórica (Capítulo 3):

- La herramienta **Dr. Scratch**, de análisis automático de proyectos Scratch, expuesta con detalle en el [sub-epígrafe 3.3.2.2](#).
- Tareas extraídas del Concurso Internacional Bebras (descrito en detalle en el [sub-epígrafe 3.3.2.1](#)). Concretamente, se seleccionaron (y tradujeron al español) un total de **9 Tareas Bebras** (*‘Bebras Tasks’*), todas ellas con formato de ítem de elección múltiple con 4 alternativas de respuesta (sólo 1 correcta). Las 9 Tareas Bebras seleccionadas fueron divididas en 3 bloques (3 tareas por bloque) en función del grupo de edad para el que fueron

diseñadas: 3 tareas B1, para 5º-6º de Primaria; 3 tareas B2, para 1º-2º de la ESO; y 3 tareas B3 para 3º y 4º de la ESO. En la Tabla 6.19 se muestran las especificaciones de las 9 Tareas Bebras, que se adjuntan al completo en el [Anexo H](#).

Tabla 6.19. Especificaciones de las 9 Tareas Bebras seleccionadas para la validación convergente del TPC

ID	Nombre de la Tarea	Fuente	Categoría	Descriptor	Grupo de Edad
B1-1	'Ice Cream Machine' (La Máquina de Helados)	Lithuanian Bebras Contest 2013 (Dagiene & Stupuriene, 2014)	?	Detectar un algoritmo; funcionamiento de máquinas; bucles	
B1-2	'Dress Code for Beavers' (La Vestimenta de los Castores)	Lit. Bebras Web – Task examples ³⁷⁶	ALG ³⁷⁷ , STRUC ³⁷⁸	?	5º y 6º de Primaria
B1-3	'Tic-tac-toe' (Las Tres en Raya)	Bebras UK-2014 ³⁷⁹	?	Abstracción y pensamiento algorítmico	
B2-1	'Water Supply' (Suministro de agua)	Lit. Bebras Web – Task examples	STRUC	?	
B2-2	'Fast Laundry' (Lavandería rápida)	Lit. Bebras Web – Task examples	ALG	?	1º y 2º de la ESO
B2-3	'Abacus' (Ábaco chino)	Bebras UK-2014	?	Abstracción, descomposición y pensamiento algorítmico	
B3-1	'Friends' (La red social)	Lit. Bebras Web – Task examples	STRUC, SOC ³⁸⁰	?	
B3-2	'Visiting Friends' (Visitando amigos)	Lithuanian Bebras Contest 2013 (Dagiene & Stupuriene, 2014)	ALG, STRUC	Conteo. Detección de patrones y periodicidad	3º y 4º de la ESO
B3-3	'Mobile Phones' (Teléfonos móviles)	Bebras UK-2014	?	Abstracción, descomposición y evaluación	

Para aplicar en una misma sesión nuestro Test de Pensamiento Computacional (TPC) junto con las Tareas Bebras seleccionadas, se generaron en *Google Drive Forms* las siguientes versiones del TPC (Figura 6.20): el TPC-B1³⁸¹ (para 5º y 6º Primaria, incluye los 28 ítems del TPC más las 3 tareas B1); el TPC-B2³⁸² (para 1º y 2º de la ESO, incluye los 28 ítems del TPC más las 3 tareas B2); y el TPC-B3³⁸³ (para 3º y 4º de la ESO, incluye los 28 ítems del TPC más las 3 tareas B3). En estas versiones se modifican ligeramente las instrucciones respecto a la original: se indica que hay una página final con las 3 tareas Bebras adicionales; y se eleva el tiempo de aplicación a 50 minutos.

³⁷⁶ <http://www.bebas.org/?q=examples>

³⁷⁷ ALG: 'Algorithmic thinking' ('Pensamiento algorítmico')

³⁷⁸ STRUC: 'Structures, patterns and arrangements' ('Estructuras, patrones y disposiciones')

³⁷⁹ Disponible en <http://www.bever-comp.org.uk/uploads/2/1/8/6/21861082/juniorukbebras2014-answers.pdf>

³⁸⁰ SOC: 'ICT and Society' ('TIC y Sociedad')

³⁸¹ <https://goo.gl/NBSweA>

³⁸² <https://goo.gl/VU0gcu>

³⁸³ <https://goo.gl/ijmuAu>



Figura 6.20. Códigos QR de enlace a las versiones TPC-B1 (izquierda), TPC-B2 (centro), y TPC-B3 (derecha)

6.5.1.3. Procedimiento

El procedimiento de aplicación difiere ligeramente de unos centros a otros, en función de cuál fue el modo en que fueron muestreados (procedimiento A, B, C o D; según ya vimos en el [sub-epígrafe 6.5.1.1](#), [Tabla 6.18](#)). Así pues:

- Los centros muestreados por el **procedimiento A y B**: son los que realizaron los cursos trimestrales *K-8* y *K-5* respectivamente de Code.org ([2015a](#), [2015b](#)). Estos centros típicamente aplicaron el TPC, y adicionalmente alguno de los instrumentos estandarizados de TEA Ediciones; y lo hicieron tanto en condición *pre* (antes de comenzar el curso *K-8* o *K-5*), como en condición *post* (tras finalizar dicho curso). A efectos de la baremación del TPC (objetivo O_{3f}) y de su validación criterial concurrente (objetivos $O_{3i} \dots O_{3n}$) sólo tomaremos en consideración las aplicaciones en condición *pre*.

Los grupos participantes relativos a estos centros son típicamente de 1º y 2º ESO (procedimiento A), o de 5º y 6º Primaria (procedimiento B), y cursan la asignatura optativa de Informática, a la que acuden 2 veces por semana (1 hora cada vez). Habitualmente, se aplicó el TPC durante la primera de las dos clases semanales, y el instrumento de TEA Ediciones en la segunda.

Para la aplicación del TPC, de carácter colectivo, el profesor de Informática siguió las instrucciones que le fueron remitidas por correo electrónico en la semana previa ([Anexo I](#)); y las respuestas directas de los estudiantes a cada uno de los ítems del TPC quedaron guardadas en la base de datos de *Google Drive* asociada al instrumento, que fue posteriormente descargada como archivo *.xls* de Excel.

Para la aplicación del correspondiente instrumento de TEA Ediciones, aplicación de carácter colectivo, los estudiantes fueron previamente dados de alta como usuarios en la plataforma 'on-line' de la editorial³⁸⁴. Llegado el día de la aplicación, los sujetos accedieron con su identificación de usuario a dicha plataforma y realizaron la prueba, que autocontiene instrucciones. Desde nuestro perfil de administrador, posteriormente pudimos descargar, a un archivo *.xls* de Excel, los resultados de los sujetos: no las respuestas directas de cada sujeto a cada elemento de la prueba, sino la puntuación total en la misma (tanto una puntuación total directa o P_d ; como una puntuación total típica o P_t , relativa al baremo de referencia). Para todos los análisis utilizaremos las puntuaciones totales directas o P_d .

³⁸⁴ <http://www.e-teaediciones.com/>

Adicionalmente, a estos centros se les solicitaron datos de rendimiento académico (asignaturas de Lengua, Matemáticas e Informática) al finalizar el trimestre de realización de los cursos *K-8* y *K-5*; así como se descargaron las estadísticas de desempeño ('número de niveles superados' y 'líneas de código escritas') de los estudiantes en dichos cursos desde la plataforma de Code.org. Estos datos servirán para el estudio de la validez criterial predictiva del TPC (objetivos O_{3o} y O_{3p}), tomado éste lógicamente en condición *pre*.

- Los centros muestreados por el **procedimiento C** son los que participaron en el estudio de validación convergente entre el TPC y la herramienta Dr. Scratch (Moreno-León & Robles, 2015; Moreno-León, Robles, & Román-González, 2015). En este grupo de centros: el TPC fue aplicado inicialmente en situación *pre*; posteriormente los sujetos iniciaron un periodo de aprendizaje con Scratch (alrededor de 8 semanas)³⁸⁵; y finalmente los sujetos realizaron una segunda aplicación del TPC en situación *post*, además de analizar sus proyectos finales de Scratch con la herramienta analítica Dr. Scratch. A efectos de la baremación del TPC (objetivo O_{3f}) sólo tomaremos en consideración la aplicación *pre*. A efectos de la validación convergente del TPC con Dr. Scratch (objetivo O_{3s}), calcularemos tanto un coeficiente de validez convergente predictivo (TPC_{pre} * Dr. Scratch) como un coeficiente de validez convergente concurrente (TPC_{post} * Dr. Scratch).

Con respecto a la aplicación del TPC, ésta fue análoga a la explicada para los centros de los procedimientos A y B. Con respecto a la aplicación de Dr. Scratch, una vez finalizado el periodo de aprendizaje de 8 semanas con Scratch, el profesor seleccionó y envió el mejor proyecto de cada estudiante (si trabajaron con metodología '*single programming*') o de cada pareja de estudiantes (si trabajaron con metodología '*pair programming*'). Una vez recibidos los proyectos, los sometimos a análisis con Dr. Scratch, volcando sus puntuaciones sobre un archivo *.xls* de Excel.

- Finalmente, los centros muestreados por el **procedimiento D** sólo realizaron una aplicación del TPC, en condición *pre*, que es la utilizada a efectos de baremación del instrumento (objetivo O_{3f}). La aplicación del TPC fue igualmente análoga a la explicada en los procedimientos A, B y C. Con la salvedad del Colegio Mirasur y el IES Politécnico de Castellón, que aplicaron las versiones TPC-B1, TPC-B2 o TPC-B3 (según el curso académico del sujeto) que servirán para el estudio de validez convergente de nuestro instrumento con las Tareas Bebras (objetivo O_{3f})

Finalmente, todos los archivos *.xls* generados durante la recogida de datos fueron exportados a un único archivo *.sav*, que constituye la matriz de datos sometida a análisis con el software SPSS versión 22. De dicho análisis surgen los resultados que exponemos a continuación.

6.5.2. Resultados

Los resultados se presentan en 3 grandes bloques: descriptivos (incluye estadísticos descriptivos del TPC, baremos y análisis de ítems), estudios de fiabilidad, y estudios de validez.

³⁸⁵ Para el periodo de aprendizaje con Scratch (MIT Media Lab, 2015), se recomendó a los centros seguir la guía didáctica 'Creative Computing' (Brennan, Balch, & Chung, 2014)

6.5.2.1. *Descriptivos*

Dando respuesta al objetivo específico **O_{3f}**, en la Tabla 6.20 se detallan los descriptivos principales de la puntuación total en el TPC (calculada como suma de respuestas correctas) del total de sujetos de la muestra. Y en la Figura 6.21 se ilustra el histograma correspondiente.

Tabla 6.20. Descriptivos de la puntuación total en el TPC para toda la muestra

N	Válido	1251
	Perdidos	0
Media		16,38
Error estándar de la media		,136
Mediana		16,00
Moda		17
Desviación estándar		4,824
Varianza		23,271
Asimetría		,058
Curtosis		-,446
Mínimo		3
Máximo		28
Percentiles	10	10,00
	20	12,00
	25	13,00
	30	14,00
	40	15,00
	50	16,00
	60	17,00
	70	19,00
	75	20,00
	80	21,00
90	23,00	

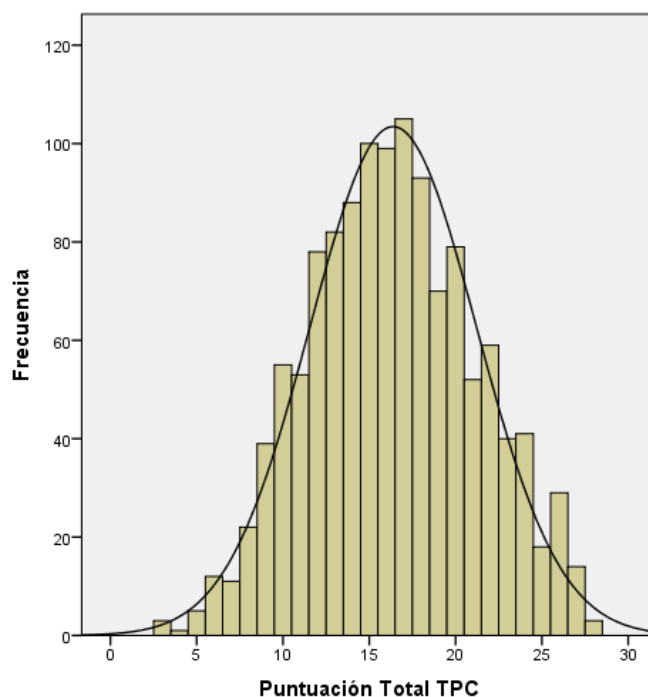


Figura 6.21. Histograma de las puntuaciones totales en el TPC para toda la muestra

Tal y como puede observarse en la anterior figura, la distribución de las puntuaciones totales en el TPC se ajusta notablemente a la curva normal; si bien, dado el elevadísimo tamaño de la muestra, la prueba de Kolgomorov-Smirnov penaliza los pequeños desajustes existentes y permite rechazar la hipótesis nula de normalidad ($Z_{k-s} = 0,052$; $p(Z_{k-s}) = 0,000 < 0,01$).

Segmentando la muestra por ciclo educativo, en la siguiente Tabla 6.21 se muestran los descriptivos principales para la puntuación total en el TPC, diferenciados para 3º Ciclo de Primaria (5º y 6º de Primaria), 1º Ciclo de Secundaria (1º y 2º de la ESO), y 2º Ciclo de Secundaria (3º y 4º de la ESO). Tal y como puede apreciarse en la tabla, existe un incremento consistente en la media y la mediana a medida que avanzamos en el ciclo educativo; y la variabilidad de las puntuaciones se incrementa ligeramente según avanzamos en el ciclo educativo. Con respecto al grado de asimetría: la distribución de las puntuaciones totales en el TPC es prácticamente simétrica en el grupo de edad central, 1º Ciclo de Secundaria ($A_s = 0,018$); en el grupo de edad inferior, 3º Ciclo de Primaria, la distribución es asimétrica positiva ($A_s = 0,125$), indicativo de que el TPC se va acercando a su suelo como instrumento en esas edades; y en el grupo de edad superior, 2º Ciclo de Secundaria, la distribución es asimétrica negativa ($A_s = -0,097$), indicativo de que el TPC se va acercando a su techo al finalizar la ESO. Puede observarse igualmente en las Figuras 6.22 y 6.23 de las páginas siguientes.

Tabla 6.21. Descriptivos de la puntuación total en el TPC por ciclo educativo

		3º Ciclo de Primaria (5º y 6º Primaria)	1º Ciclo de Secundaria (1º y 2º ESO)	2º Ciclo de Secundaria (3º y 4º ESO)
N	Válido	176	735	340
	Perdidos	0	0	0
Media		13,76	16,24	18,05
Error estándar de la media		,326	,167	,274
Mediana		14,00	16,00	18,00
Moda		15	18	17
Desviación estándar		4,330	4,519	5,049
Varianza		18,746	20,419	25,496
Asimetría		,125	,018	-,097
Curtosis		-,148	-,453	-,577
Mínimo		3	3	3
Máximo		26	27	28
Percentiles	10	8,00	10,00	12,00
	20	10,00	12,00	13,20
	25	11,00	13,00	14,00
	30	11,00	14,00	15,00
	40	13,00	15,00	17,00
	50	14,00	16,00	18,00
	60	15,00	17,00	19,00
	70	16,00	19,00	21,00
	75	16,75	20,00	22,00
	80	17,00	20,00	23,00
	90	20,00	22,00	25,00

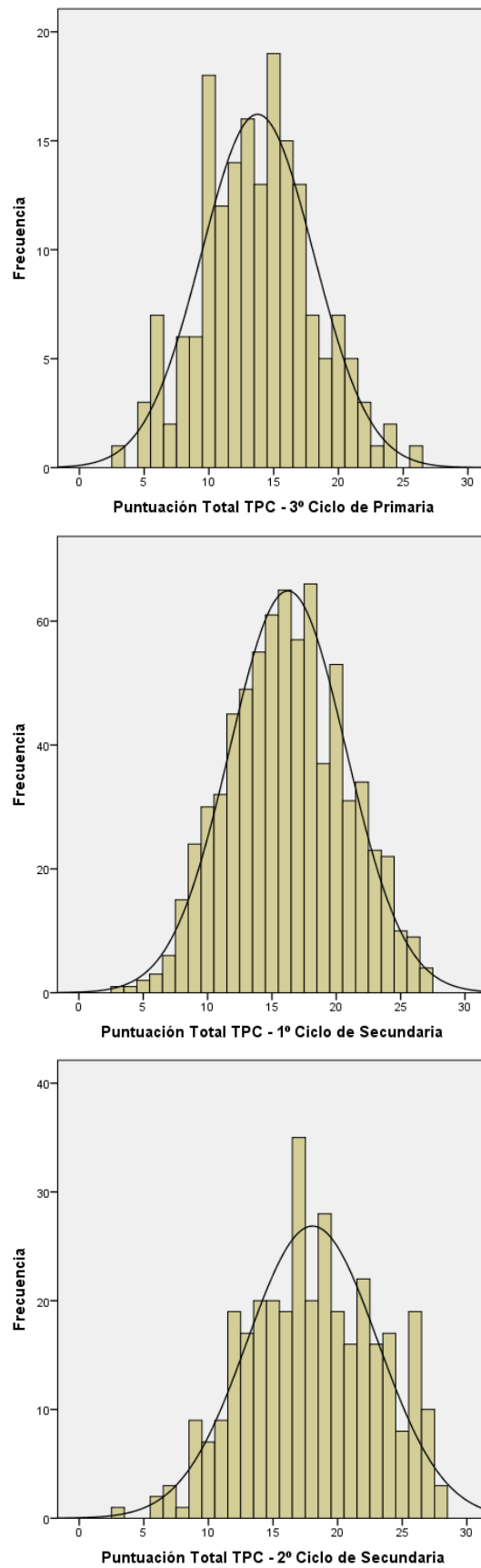


Figura 6.22. Comparativa de las distribuciones de la puntuación total en el TPC por ciclo educativo: arriba, la distribución de 3º Ciclo de Primaria; en el centro, la de 1º Ciclo de Secundaria; y abajo, la de 2º Ciclo de Secundaria

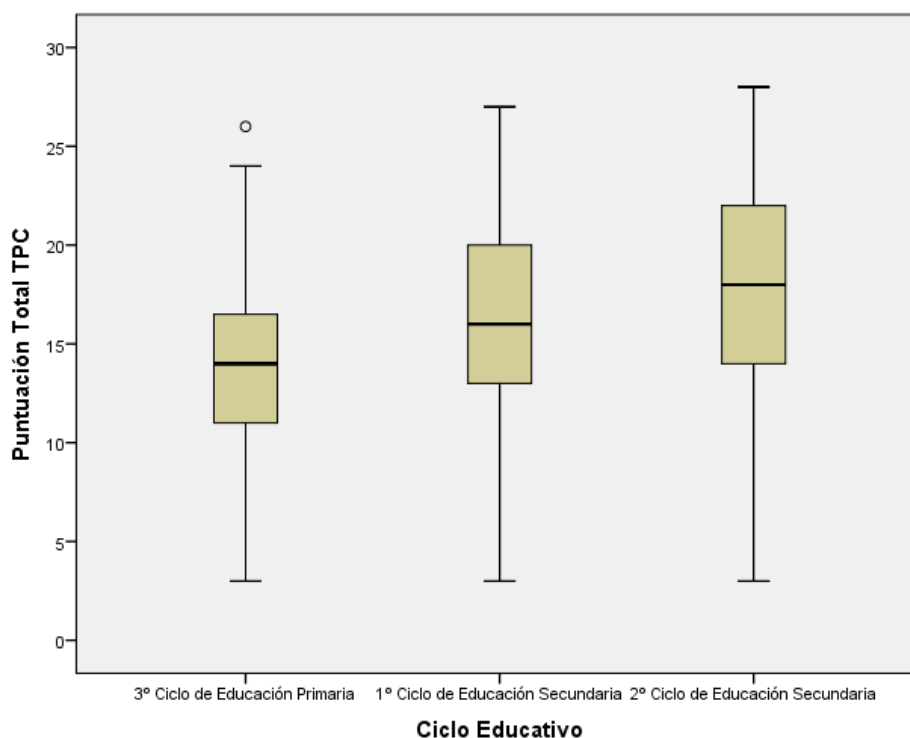


Figura 6.23. Diagrama de cajas de la puntuación total en el TPC por ciclo educativo. El ‘outlier’ es un caso de 3º Ciclo de Primaria (6º de Primaria), que obtuvo una puntuación de 26 (≈ 3 desviaciones típicas por encima de la media de su ciclo)

El ANOVA arroja diferencias estadísticamente muy significativas en la puntuación total del TPC según ciclo educativo ($F = 50,514$; $p_{(F)} = 0,000 < 0,01$). La prueba *post-hoc* de Tukey, de diferencias entre pares de medias, muestra diferencias muy significativas ($p_{(Tukey)} = 0,000 < 0,01$) entre todas las parejas posibles (3º Ciclo de Primaria vs. 1º Ciclo de Secundaria; 3º Ciclo de Primaria vs. 2º Ciclo de Secundaria; y 1º Ciclo de Secundaria vs. 2º Ciclo de Secundaria). En la Figura 6.24 se ilustran las medias en la puntuación total en el TPC por ciclo educativo, apreciándose un incremento constante.

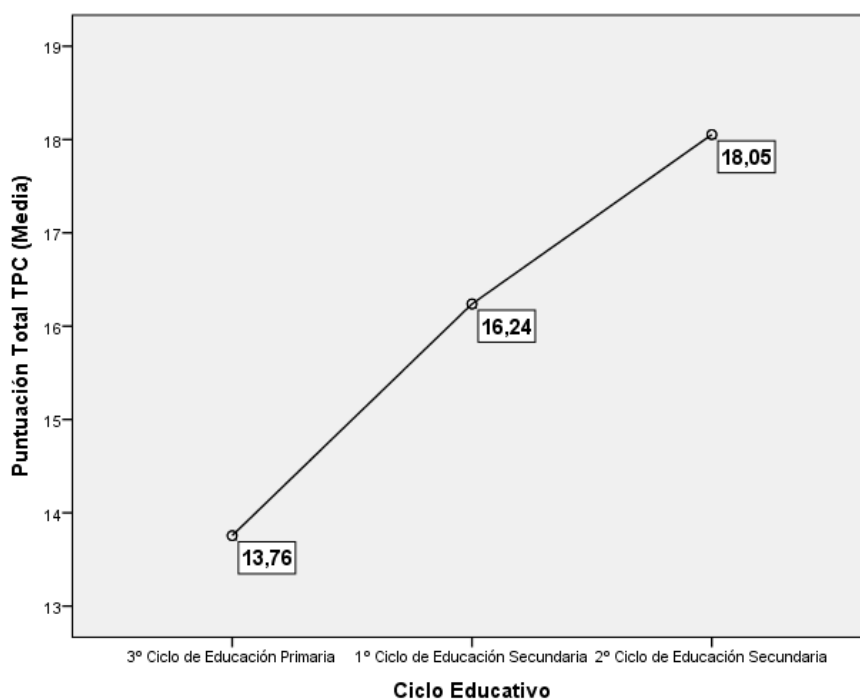


Figura 6.24. Media en la puntuación total del TPC por ciclo educativo

Haciendo un análisis análogo, ahora por curso académico, encontramos los resultados que se muestran en la siguiente Tabla 6.22 y Figura 6.25.

Tabla 6.22. Descriptivos de la puntuación total en el TPC por curso académico

		5º Primaria	6º Primaria	1º ESO	2º ESO	3º ESO	4º ESO
N	Válido	103	73	433	302	199	141
	Perdidos	0	0	0	0	0	0
Media		13,09	14,70	15,74	16,95	17,16	19,31
Error estándar de la media		,398	,534	,213	,262	,329	,449
Mediana		13,00	15,00	16,00	17,00	17,00	20,00
Moda		13	15	16	18	19	17
Desviación estándar		4,044	4,566	4,429	4,557	4,647	5,336
Varianza		16,355	20,852	19,619	20,762	21,590	28,473
Mínimo		3	5	5	3	6	3
Máximo		22	26	27	27	27	28
Percentiles	10	8,00	10,00	10,00	11,00	11,00	12,00
	20	9,80	10,80	12,00	13,00	13,00	15,00
	25	10,00	11,00	12,50	14,00	14,00	16,00
	30	11,00	12,00	13,00	15,00	14,00	17,00
	40	12,00	13,00	14,00	16,00	16,00	17,00
	50	13,00	15,00	16,00	17,00	17,00	20,00
	60	14,00	15,00	17,00	18,00	19,00	22,00
	70	15,80	16,80	18,00	20,00	19,00	23,00
	75	16,00	18,00	19,00	20,00	20,00	24,00
	80	17,00	19,00	20,00	21,00	21,00	24,00
90	18,00	21,00	22,00	23,00	23,00	26,00	

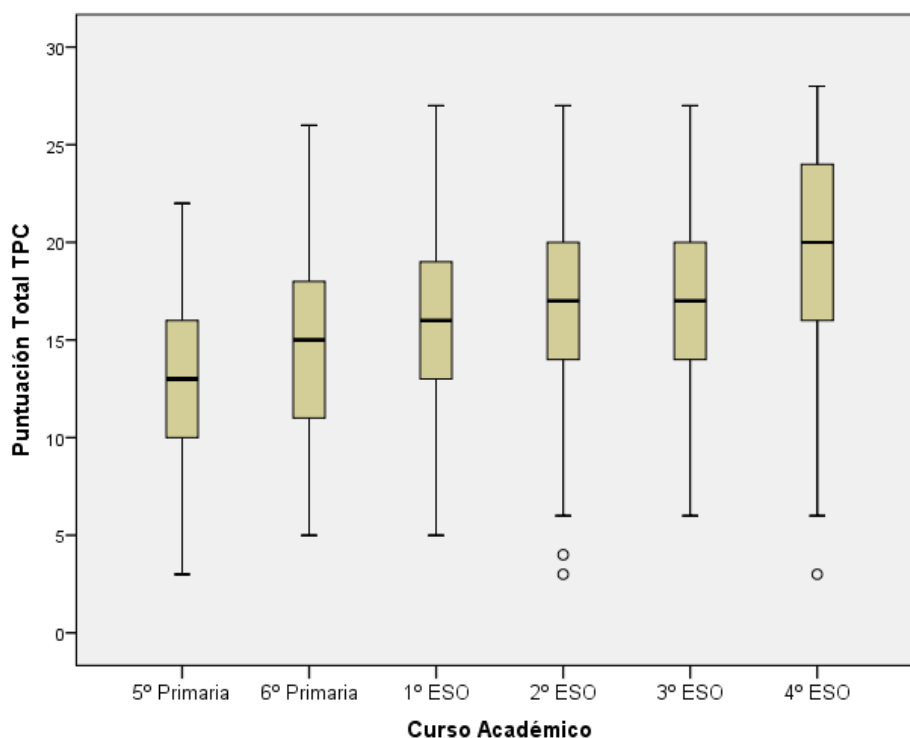


Figura 6.25. Diagrama de cajas de la puntuación total en el TPC por curso académico

Se observa, pues, un rendimiento en el TPC consistentemente creciente a medida que avanzamos de curso académico; algo deseable para un instrumento que supuestamente mide una aptitud de tipo cognitivo que, como tal, está vinculada a aspectos madurativos.

El ANOVA arroja diferencias estadísticamente muy significativas en la puntuación total del TPC según curso académico ($F = 27,930$; $p_{(F)} = 0,000 < 0,01$). Las medias en la puntuación total del TPC por curso académico se muestran en la Figura 6.26; y la significación de la prueba *post-hoc* de Tukey para cada diferencia de medias, se presenta en la Tabla 6.23.

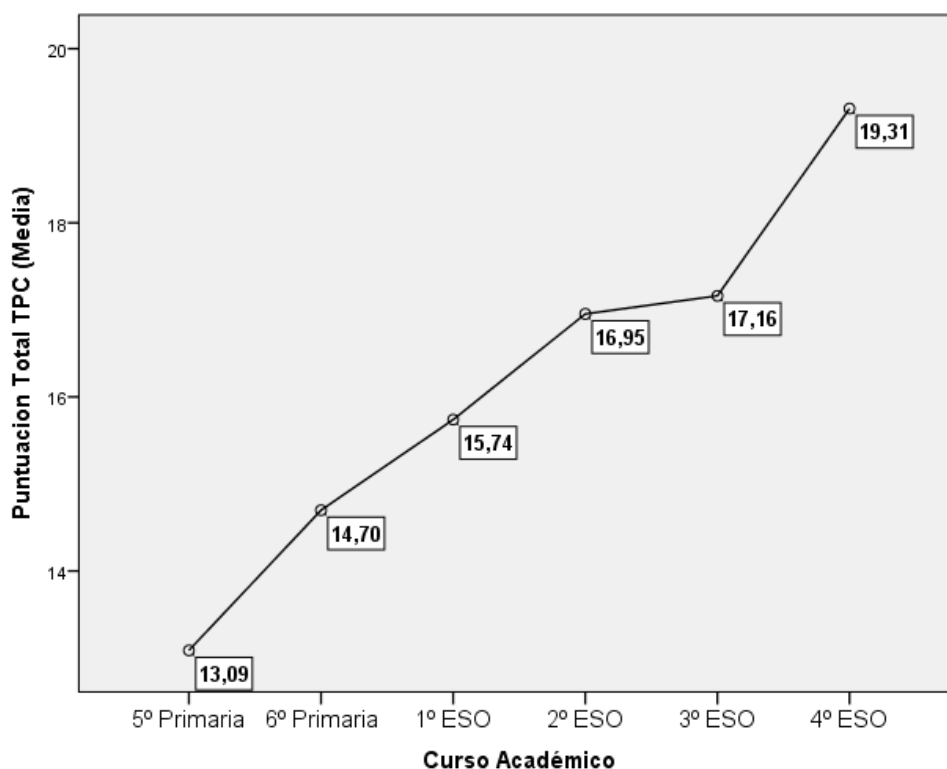


Figura 6.26. Media en la puntuación total del TPC por curso académico

Tabla 6.23. Significatividad de la prueba *post-hoc* de Tukey para cada diferencia de medias

	5º Primaria (N=103)	6º Primaria (N=73)	1º ESO (N=433)	2º ESO (N=302)	3º ESO (N=199)	4º ESO (N=141)
5º Primaria	-	,196	,000**	,000**	,000**	,000**
6º Primaria		-	,470	,002**	,001**	,000**
1º ESO			-	,006**	,004**	,000**
2º ESO				-	,996	,000**
3º ESO					-	,000**
4º ESO						-

** Significatividad al nivel $< 0,01$

Tal y como puede comprobarse en la anterior Tabla 6.23, todas las diferencias de medias entre pares de cursos académicos son estadísticamente muy significativas ($p_{(Tukey)} < 0,01$), a excepción del par 2º ESO – 3º ESO (cuyas medias son muy próximas según se aprecia en la anterior Figura 6.26); y los pares adyacentes a 6º Primaria (5º Primaria – 6º Primaria, y 6º Primaria – 1º ESO), cuya significatividad se ve afectada por la baja muestra relativa de 6º Primaria (sólo $N = 73$).

Con respecto a posibles diferencias de rendimiento en el TPC según sexo, encontramos una diferencia estadísticamente muy significativa en la puntuación total del TPC a favor del colectivo masculino ($t = 5,374$; $p_{(t)} = 0,000 < 0,01$), con un tamaño del efecto asociado $d = 0,31$ (Tabla 6.24)

Tabla 6.24. Diferencias de rendimiento en el TPC por sexo

	Sexo	N	Media	Desviación estándar	t	p _(t)	d de Cohen
Puntuación Total TPC	Chico	730	16,99	4,802	5,374**	,000	0,31
	Chica	521	15,52	4,727			

** Significativo al nivel $p_{(t)} < 0,01$

Si analizamos las diferencias en el TPC por sexo, a lo largo de los ciclos educativos (Tabla 6.25), encontramos siempre valores superiores en el colectivo masculino, pero dichas diferencias son sólo estadísticamente muy significativas a partir de 1º Ciclo de la ESO ($t = 2,928$; $p_{(t)} = 0,004 < 0,01$), intensificándose en el 2º Ciclo de la ESO ($t = 3,451$; $p_{(t)} = 0,001 < 0,01$). Parece, pues, que la brecha de rendimiento en el TPC por género es progresiva según avanzamos en el ciclo educativo (Figura 6.27).

Tabla 6.25. Diferencias de rendimiento en el TPC por sexo, a lo largo de los ciclos educativos

	Sexo	N	Media	Desviación estándar	t	P _(t)	d de Cohen	
Puntuación Total TPC	3º Ciclo de Primaria	Chico	78	14,40	4,185	1,765	,079	0,27
		Chica	98	13,24	4,396			
	1º Ciclo de ESO	Chico	450	16,62	4,463	2,928**	,004	0,22
		Chica	285	15,63	4,547			
	2º Ciclo de ESO	Chico	202	18,82	5,115	3,451**	,001	0,38
		Chica	138	16,93	4,749			

** Significativo al nivel $p_{(t)} < 0,01$

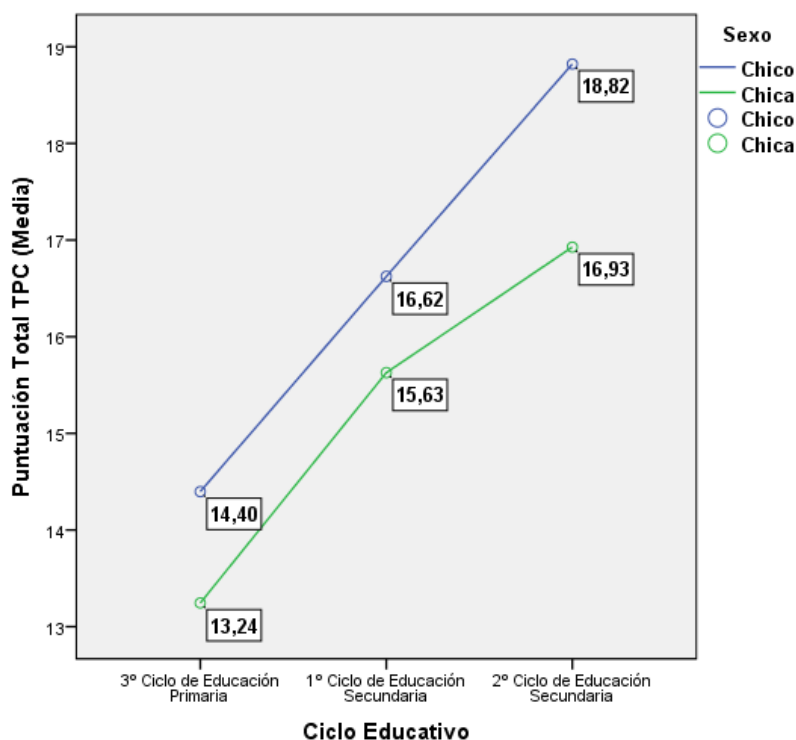


Figura 6.27. Diferencias por sexo en la puntuación total en el TPC, a lo largo de los ciclos educativos

Con respecto a las 2 preguntas de autoevaluación que se formulan al final del TPC: “De 0 a 10, ¿cómo consideras que te ha salido el Test?” (la denominaremos ‘autoeficacia específica’), y “De 0 a 10, ¿cómo consideras que se te dan los ordenadores y la informática?” (la denominaremos ‘autoeficacia general’); sus descriptivos se presentan en la Tabla 6.26, y sus correlaciones con la puntuación total en el TPC en la Tabla 6.27.

Tabla 6.26. Descriptivos relativos a las preguntas de autoevaluación

	De 0 a 10, ¿cómo consideras que te ha salido el Test? (‘Autoeficacia específica’)	De 0 a 10, ¿cómo consideras que se te dan los ordenadores y la informática? (‘Autoeficacia general’)
N	Válido	1237
	Perdidos	14
Media	7,07	7,38
Mediana	7,00	8,00
Moda	7	8
Desviación estándar	1,812	1,955
Varianza	3,283	3,824
Asimetría	-1,034	-1,091
Curtosis	2,097	1,639
Mínimo	0	0
Máximo	10	10
Percentiles	10	5,00
	20	6,00
	25	6,00
	30	6,00
	40	7,00
	50	7,00
	60	8,00
	70	8,00
	75	8,00
	80	8,00
90	9,00	

Tabla 6.27. Correlaciones entre autoeficacia específica, autoeficacia general y puntuación total TPC

	Puntuación Total TPC	Autoeficacia específica	Autoeficacia general
Puntuación Total TPC	-	,381**	,188**
Autoeficacia específica		-	,513**
Autoeficacia general			-

** Correlación significativa al nivel $p(r) < 0,01$

Tal y como puede observarse, todas las correlaciones son estadísticamente muy significativas ($p(r) < 0,01$). Destaca el hecho de que la puntuación total en el TPC correlaciona con mayor intensidad con la ‘autoeficacia específica’ ($r = +0,381$) que con la ‘autoeficacia general’ ($r = +0,188$); tal y como era de esperar.

En ambas preguntas de autoeficacia aparecen diferencias estadísticamente muy significativas en función del ciclo educativo ($F = 8,743$ y $F = 16,825$, respectivamente para ‘autoeficacia específica’ y ‘autoeficacia general’, en ambas $p(F) = 0,000 < 0,01$). Tal y como puede verse en las Figuras 6.28 y 6.29, la percepción de autoeficacia desciende según avanzamos en ciclo educativo, justo a la inversa que el comportamiento de la puntuación total en el TPC (que aumenta según avanzamos en ciclo educativo).

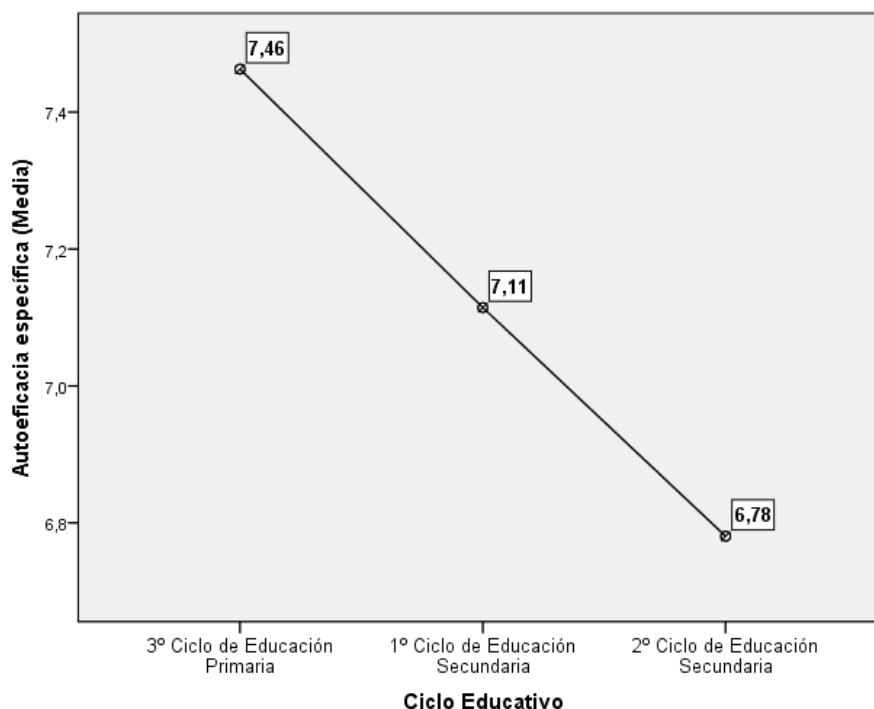


Figura 6.28. Evolución de la ‘autoeficacia específica’ por ciclo educativo

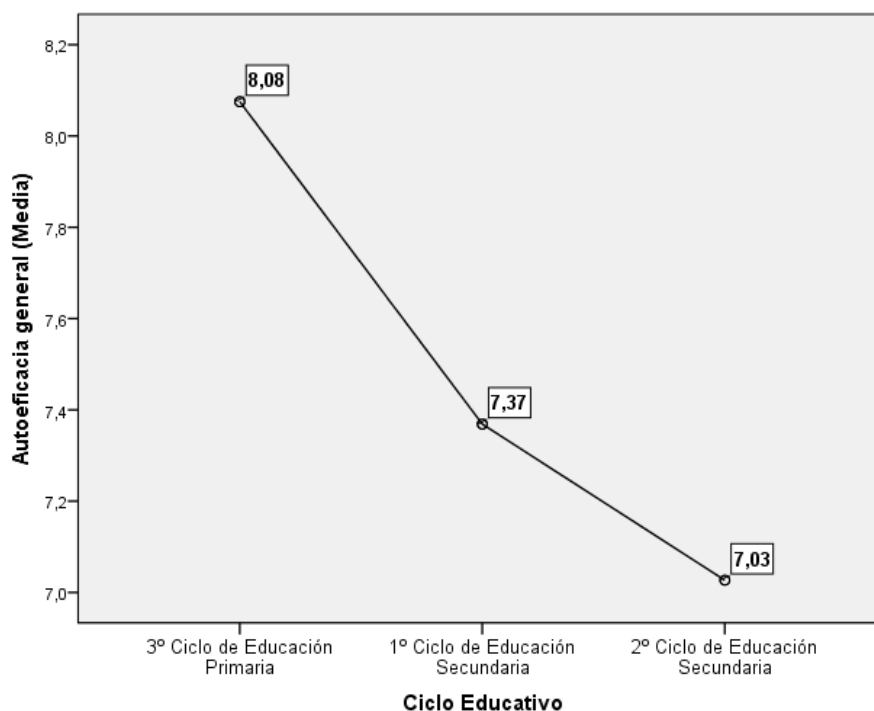


Figura 6.29. Evolución de la ‘autoeficacia general’ por ciclo educativo

Esta aparente paradoja se explica porque, a medida que avanzamos en ciclo educativo, los estudiantes se muestran más autocríticos en ambas cuestiones, pero también más consistentes en sus apreciaciones. Así, en la Tabla 6.28 observamos como la correlación de la puntuación total en el TPC con la ‘autoeficacia específica’ se eleva desde $r = +0,361$ en 3º Ciclo de Primaria a $r = +0,527$ en 2º Ciclo de la ESO. Dicho en otras palabras, a medida que los estudiantes son más mayores, son más precisos en la apreciación de cómo ha sido su desempeño en el test.

Tabla 6.28. Correlaciones entre autoeficacia específica, autoeficacia general y puntuación total TPC, por ciclo educativo

3º Ciclo de Educación Primaria			
	Puntuación Total TPC	Autoeficacia específica	Autoeficacia general
Puntuación Total TPC	-	,361**	,250**
Autoeficacia específica		-	,433**
Autoeficacia general			-
1º Ciclo de Educación Secundaria			
	Puntuación Total TPC	Autoeficacia específica	Autoeficacia general
Puntuación Total TPC	-	,387**	,180**
Autoeficacia específica		-	,451**
Autoeficacia general			-
2º Ciclo de Educación Secundaria			
	Puntuación Total TPC	Autoeficacia específica	Autoeficacia general
Puntuación Total TPC	-	,527**	,350**
Autoeficacia específica		-	,613**
Autoeficacia general			-

** Correlación significativa al nivel $p_{(t)} < 0,01$

Para concluir con las cuestiones de autoeficacia, en la Tabla 6.29 se muestra la respuesta diferencial a ambas por sexo. Tal y como puede observarse, tanto en ‘autoeficacia específica’ ($t = 7,261$; $p_{(t)} = 0,000 < 0,01$) como en ‘autoeficacia general’ ($t = 4,136$; $p_{(t)} = 0,000 < 0,01$), encontramos diferencias estadísticamente muy significativas a favor de los chicos; tal y como ocurría con la puntuación total en el TPC. Sin embargo, es de destacar que, si el tamaño del efecto del sexo sobre la puntuación en el TPC fue $d = 0,31$; el tamaño sobre la ‘autoeficacia específica’ asciende a $d = 0,42$. Dicho en otras palabras, en la percepción de autoeficacia se abre una brecha de género adicional a la ya existente en el desempeño objetivo.

Tabla 6.29. Diferencias en ‘autoeficacia específica’ y ‘autoeficacia general’ por sexo

	Sexo	N	Media	Desviación estándar	t	$p_{(t)}$	d de Cohen
Autoeficacia específica	Chico	723	7,38	1,787	7,261**	,000	0,42
	Chica	514	6,64	1,759			
Autoeficacia general	Chico	723	7,57	1,968	4,136**	,000	0,24
	Chica	509	7,10	1,906			

** Significativo al nivel $p_{(t)} < 0,01$

6.5.2.1.1. Análisis de ítems

Siguiendo las recomendaciones de Buffum *et al.* (2015), calculamos los siguientes índices para cada uno de los ítems del TPC: índice de dificultad, correlación ítem-total, e índice de discriminación.

En cuanto al primero de los índices, en la Tabla 6.30 se presenta el índice de dificultad (p_i) para cada uno de los ítems del TPC, expresado como porcentaje de sujetos que acertaron el ítem en tanto por uno; y también el índice de dificultad corregido ($p_{i\text{-corregido}}$), que toma en cuenta el efecto del acierto al azar en los ítems de elección múltiple. Tal y como puede observarse en la última fila de la tabla, el TPC tiene un índice de dificultad promedio de 0,59 (sin corregir) o 0,45 (corregido): es decir, en ambos casos, se sitúa en valores de dificultad media (en el rango 0,40 a 0,60) tal y como se recomienda para pruebas de aptitud (Navas Ara, 2001). Igualmente, debemos prestar atención a los ítems excesivamente fáciles, con $p_{i\text{-corregido}} \geq 0,95$ (sólo el ítem 1 se aproxima a ese nivel con 0,94); o excesivamente difíciles, con $p_{i\text{-corregido}} \leq 0,05$ (encontramos los ítems 12, 15, y 23; en rosa). Estos ítems de facilidad o dificultad extrema serán presumiblemente los menos fiables y discriminatorios del instrumento, y candidatos a ser revisados o eliminados de futuras versiones del test.

Tabla 6.30. Índice de dificultad (sin corregir y corregido) para cada uno de los ítems del TPC

Ítem	Índice de Dificultad sin corregir (p_i)	Índice de Dificultad corregido ($p_{i\text{-corregido}}$)
1	0,96	0,94
2	0,91	0,87
3	0,79	0,73
4	0,51	0,34
5	0,89	0,86
6	0,88	0,84
7	0,66	0,54
8	0,40	0,20
9	0,93	0,90
10	0,70	0,60
11	0,75	0,67
12	0,23	-0,03
13	0,72	0,63
14	0,49	0,32
15	0,26	0,01
16	0,37	0,16
17	0,49	0,32
18	0,58	0,44
19	0,49	0,32
20	0,55	0,40
21	0,54	0,39
22	0,32	0,09
23	0,16	-0,12
24	0,66	0,55
25	0,34	0,12
26	0,59	0,45
27	0,55	0,41
28	0,66	0,55
Media	0,59	0,45

En la Figura 6.30 se ilustran los anteriores índices de dificultad. Como puede verse, el test tiene una dificultad progresiva (tendencia desde valores p_i altos al inicio del test hacia valores p_i bajos), tal y como se recomienda en pruebas de aptitud (Navas Ara, 2001)

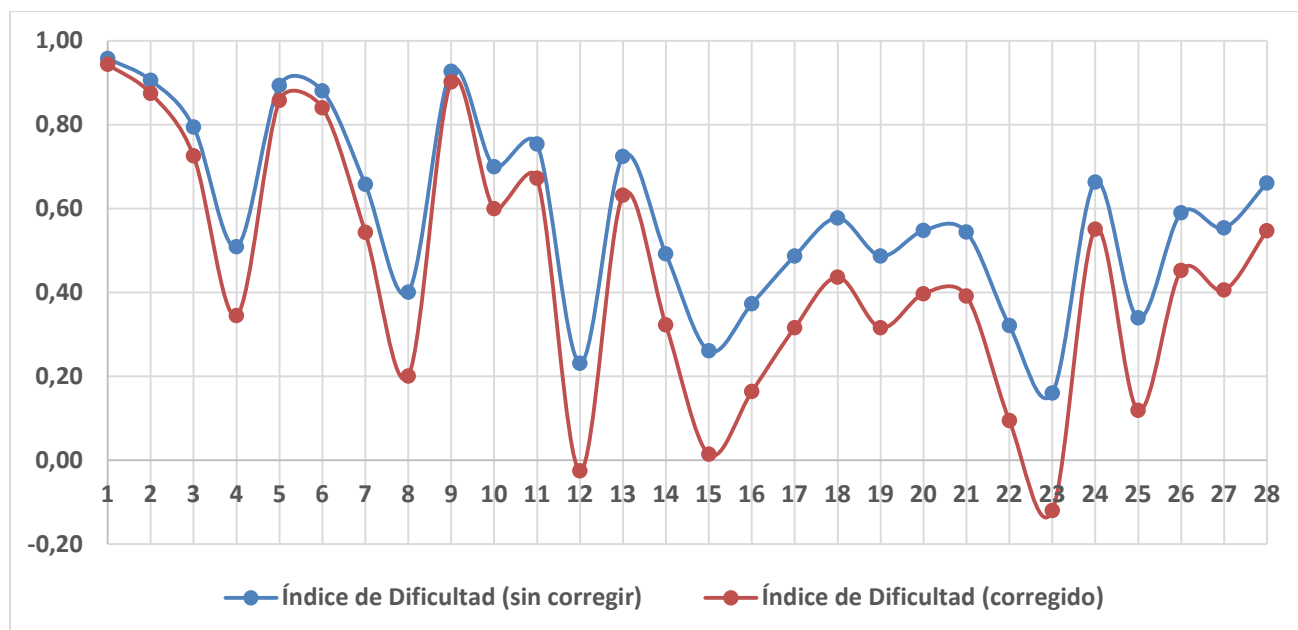


Figura 6.30. Índice de dificultad (sin corregir y corregido) para cada uno de los ítems del TPC

Con respecto a la correlación ítem-total (*Item-Total Correlation* o ITC), se calcula como coeficiente de correlación de cada uno de los ítems del TPC con la puntuación total en el test. Se muestra en la Tabla 6.31.

Tabla 6.31. Correlación ítem-total para cada uno de los ítems del TPC

Ítem	Correlación Ítem-Total (ITC)	Ítem	Correlación Ítem-Total (ITC)
1	0,20	15	0,16
2	0,36	16	0,43
3	0,41	17	0,43
4	0,49	18	0,43
5	0,32	19	0,34
6	0,36	20	0,50
7	0,50	21	0,41
8	0,38	22	0,47
9	0,31	23	0,22
10	0,39	24	0,44
11	0,43	25	0,35
12	0,25	26	0,41
13	0,41	27	0,45
14	0,52	28	0,42
Correlación Ítem-Total promedio = 0,39			

Tal y como nos recuerdan Buffum *et al.* (2015), el test está bien construido si su correlación ítem-total promedio se sitúa por encima de 0,30. Así es el caso de nuestro TPC, en que el valor promedio de la correlación ítem-total es $ITC = 0,39 > 0,30$. Igualmente, debemos prestar atención a los ítems con valores $ITC < 0,30$ (es el caso, como era de esperar, de los ítems 1, 12, 15, y 23; en rosa en la tabla anterior), que son candidatos a ser revisados o eliminados de futuras versiones del test por sus dudosas propiedades psicométricas. Se ilustra en la siguiente Figura 6.31.

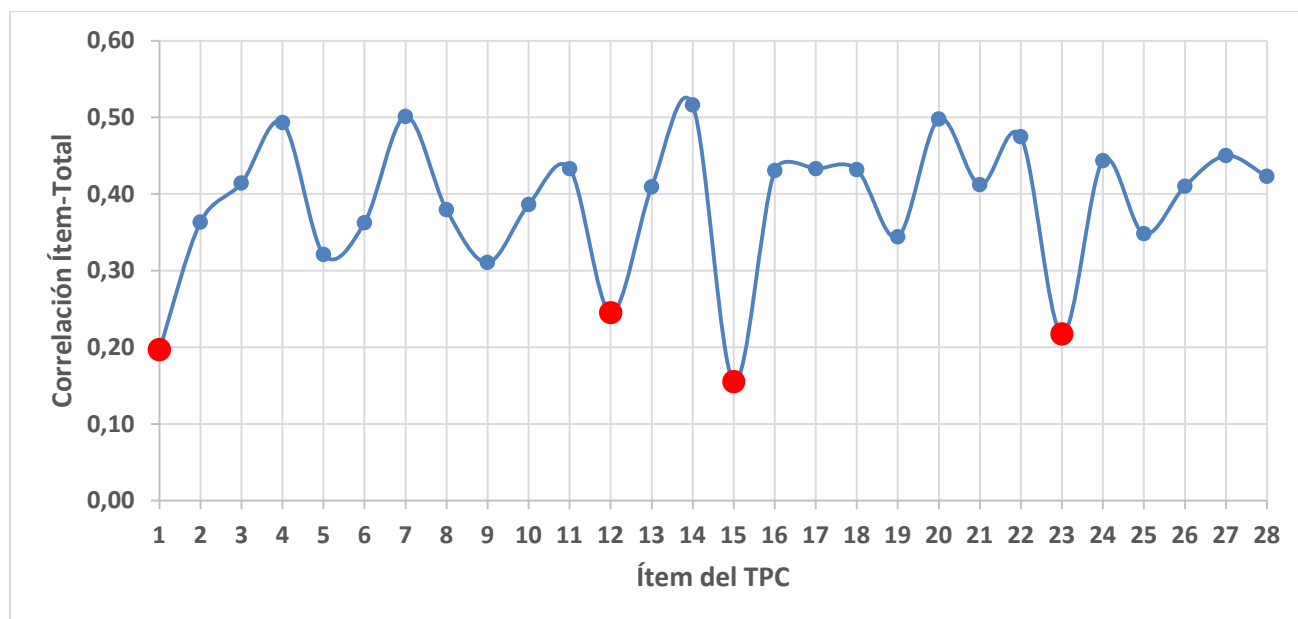


Figura 6.31. Correlación ítem-total para cada uno de los ítems del TPC

Con respecto al índice de discriminación (D) de cada uno de los ítems del TPC, calculado como diferencia entre número de aciertos en el ítem del grupo de rendimiento superior en el test (27% superior) y del grupo de rendimiento inferior (27% inferior) dividido por el número de sujetos que conforman el grupo más numeroso, se presenta en la Tabla 6.32.

Tabla 6.32. Índice de Discriminación (D) para cada uno de los ítems del TPC

Ítem	Índice de Discriminación (D)	Ítem	Índice de Discriminación (D)
1	0,02	15	0,11
2	0,18	16	0,47
3	0,36	17	0,49
4	0,59	18	0,49
5	0,16	19	0,38
6	0,20	20	0,59
7	0,54	21	0,49
8	0,43	22	0,53
9	0,11	23	0,14
10	0,38	24	0,47
11	0,41	25	0,38
12	0,21	26	0,47
13	0,39	27	0,54
14	0,64	28	0,44
Índice de Discriminación promedio = 0,38			

Tal y como nos recuerdan Buffum *et al.* (2015), el test está bien construido si su índice de discriminación promedio se sitúa por encima de 0,30. Así es el caso de nuestro TPC, en que el valor promedio es $D = 0,38 > 0,30$. Igualmente, debemos prestar atención a los ítems con valores $D < 0,30$ (Navas Ara, 2011), que, por su bajo poder discriminatorio, son candidatos a ser revisados o eliminados en futuras versiones del test: es el caso de los ítems 1, 12, 15 y 23 (de los que ya

conocíamos sus debilidades psicométricas), a los que se unen ahora los ítems 2, 5, 6 y 9 (que son ítems muy fáciles, con $p_i \approx 0,90$, lo que incide en su baja capacidad de discriminación). Se ilustra en la Figura 6.32.

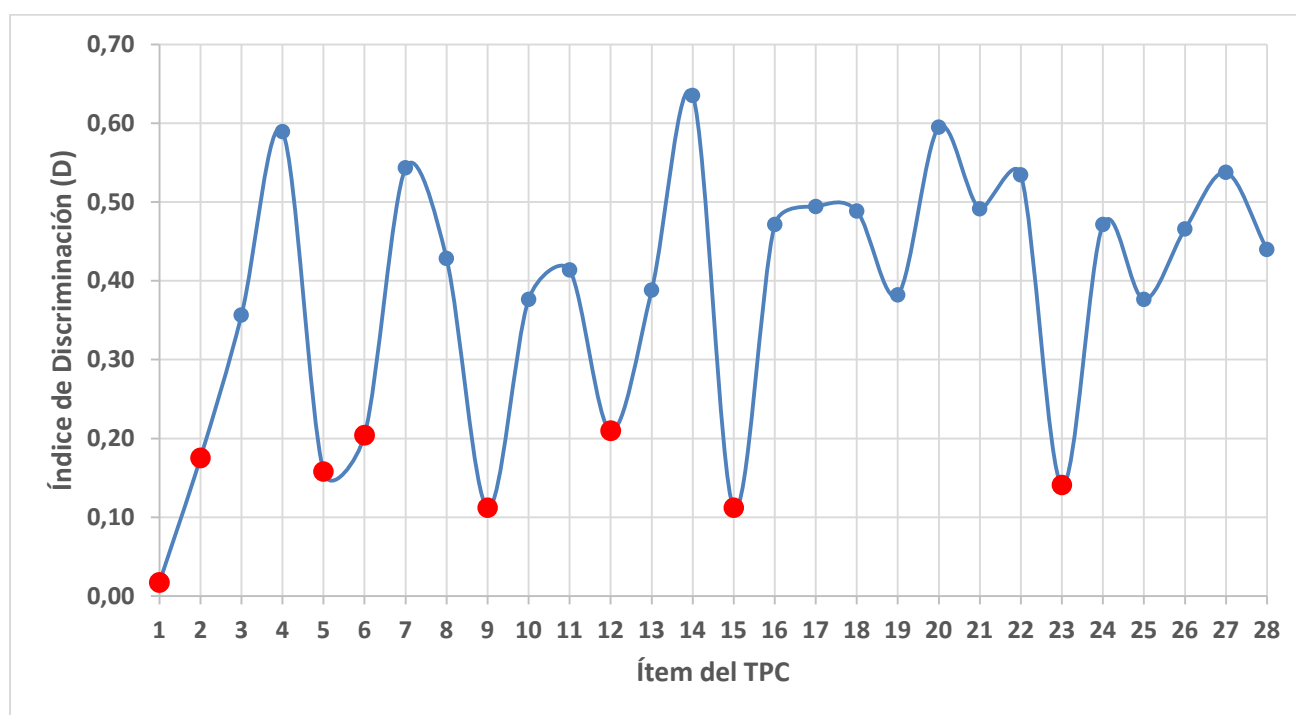


Figura 6.32. Índice de Discriminación (D) para cada uno de los ítems del TPC

6.5.2.2. *Fiabilidad*

Estudiamos la fiabilidad del TPC desde dos puntos de vista: fiabilidad como consistencia interna (objetivo específico O_{3g}), y fiabilidad como estabilidad (objetivo específico O_{3h})

6.5.2.2.1. *Consistencia interna*

Abordando el objetivo específico O_{3g} , la consistencia interna del TPC medida a través del estadístico Alfa de Cronbach es $\alpha = 0,793 \approx 0,80$ (Tabla 6.33), que puede considerarse una fiabilidad ‘buena’ (Navas Ara, 2001).

Tabla 6.33. Fiabilidad del TPC como consistencia interna

Muestra	Estadísticas de fiabilidad			Estadísticas de escala			
N de sujetos	Alfa de Cronbach	N de ítems	Media	Varianza	Desviación estándar	N de ítems	
1251	,793	28	16,38	23,271	4,824	28	

En la Tabla 6.34, se muestran las estadísticas de la escala en caso de eliminar cada uno de los ítems de la misma; así como el Alfa de Cronbach corregido en caso igualmente de prescindir de cada uno de los ítems. Tal y como puede verse, sólo conseguimos incrementar la consistencia interna del TPC eliminando el ítem 15 (nuevo $\alpha = 0,798$; lo que supone un incremento, prácticamente inapreciable, de +0,005). Eliminando el ítem 12 o el ítem 23, la fiabilidad no se resiente (se mantiene $\alpha = 0,793$).

Tabla 6.34. Fiabilidad del TPC en caso de eliminar cada uno de los ítems

	Media de escala si el ítem se ha suprimido	Varianza de escala si el ítem se ha suprimido	Alfa de Cronbach si el ítem se ha suprimido	Incremento de Alfa de Cronbach si el ítem se ha suprimido
Ítem 1	15,42	22,929	,792	-0,001
Ítem 2	15,48	22,333	,788	-0,005
Ítem 3	15,59	21,818	,785	-0,008
Ítem 4	15,87	21,142	,782	-0,011
Ítem 5	15,49	22,408	,789	-0,004
Ítem 6	15,50	22,241	,787	-0,006
Ítem 7	15,73	21,201	,781	-0,012
Ítem 8	15,98	21,717	,788	-0,005
Ítem 9	15,46	22,557	,789	-0,004
Ítem 10	15,68	21,774	,787	-0,006
Ítem 11	15,63	21,656	,785	-0,008
Ítem 12	16,15	22,452	,793	=0,000
Ítem 13	15,66	21,705	,786	-0,007
Ítem 14	15,89	21,032	,780	-0,013
Ítem 15	16,12	22,806	,798	+0,005
Ítem 16	16,01	21,498	,785	-0,008
Ítem 17	15,90	21,433	,785	-0,008
Ítem 18	15,80	21,456	,785	-0,008
Ítem 19	15,90	21,862	,790	-0,003
Ítem 20	15,83	21,129	,781	-0,012
Ítem 21	15,84	21,539	,786	-0,007
Ítem 22	16,06	21,351	,782	-0,011
Ítem 23	16,22	22,637	,793	=0,000
Ítem 24	15,72	21,471	,784	-0,009
Ítem 25	16,04	21,905	,789	-0,004
Ítem 26	15,79	21,566	,786	-0,007
Ítem 27	15,83	21,360	,784	-0,009
Ítem 28	15,72	21,563	,785	-0,008

En la Tabla 6.35, se muestran los valores de Alfa de Cronbach en función del ciclo educativo y del curso académico. Tal y como puede observarse, la consistencia interna del TPC aumenta según avanzamos de ciclo educativo y curso académico.

Tabla 6.35. Fiabilidad del TPC como consistencia interna, en función de ciclo educativo y curso académico

Ciclo Educativo	Alfa de Cronbach	N de sujetos	Curso Académico	Alfa de Cronbach	N de sujetos
3º Ciclo de Primaria	,721	176	5º de Primaria	,668	103
			6º de Primaria	,766	73
1º Ciclo de ESO	,762	735	1º ESO	,749	433
			2º ESO	,772	302
2º Ciclo de ESO	,824	340	3º ESO	,784	199
			4º ESO	,854	141

Finalmente, en la Tabla 6.36 se detallan los valores Alfa de Cronbach en función del dispositivo de aplicación (ordenador o tableta). Tal y como puede observarse, la consistencia interna del TPC es mayor cuando se aplica en dispositivos tipo 'tablet' (la consistencia interna sube a $\alpha = 0,817$); efecto especialmente notable en el 1º Ciclo de Educación Secundaria, foco central del instrumento, en donde el valor Alfa de Cronbach asciende hasta un notable $\alpha = 0,836 \approx 0,84$.

Tabla 6.36. Fiabilidad del TPC como consistencia interna, en función del dispositivo de aplicación

Dispositivo de aplicación	Alfa de Cronbach	N de sujetos	Ciclo Educativo	Alfa de Cronbach	N de sujetos
Ordenador	,786	1001	3º Ciclo de Primaria	,719	66
			1º Ciclo de ESO	,744	659
			2º Ciclo de ESO	,824	276
Tableta	,817	250	3º Ciclo de Primaria	,712	110
			1º Ciclo de ESO	,836	76
			2º Ciclo de ESO	,825	64

6.5.2.2.2. Estabilidad

Abordando ahora el objetivo específico O_{3h} , la fiabilidad del TPC como estabilidad temporal (ρ_{xx}), calculada como correlación entre las puntuaciones totales en el mismo, derivadas de dos aplicaciones sucesivas a los mismos sujetos, es $\rho_{xx} = 0,704 \approx 0,70$; que puede considerarse un valor aceptable (Navas Ara, 2001). Para calcular dicho valor, sólo se tomaron en cuenta las aplicaciones *pre* y *post* en el TPC de los sujetos *testigo* de los centros muestreados según el **procedimiento A** (ver Tabla 6.18); es decir, sujetos que forman parte de los grupos *de control* que configuramos para abordar la evaluación del curso 'K-8 Intro to Computer Science' (Code.org, 2015a) que se aborda en el Capítulo 7 de esta tesis doctoral. Entre ambas aplicaciones *pre* y *post* transcurrió un periodo de 10-12 semanas. Se muestra en las siguientes Tabla 6.37 y Figura 6.33.

Tabla 6.37. Fiabilidad del TPC como estabilidad temporal

	Correlación de Pearson	Puntuación Total TPC - <i>post</i>
Puntuación Total TPC - <i>pre</i>		,704**
	Sig. (bilateral)	,000
	N	54

** Correlación significativa al nivel $< 0,01$

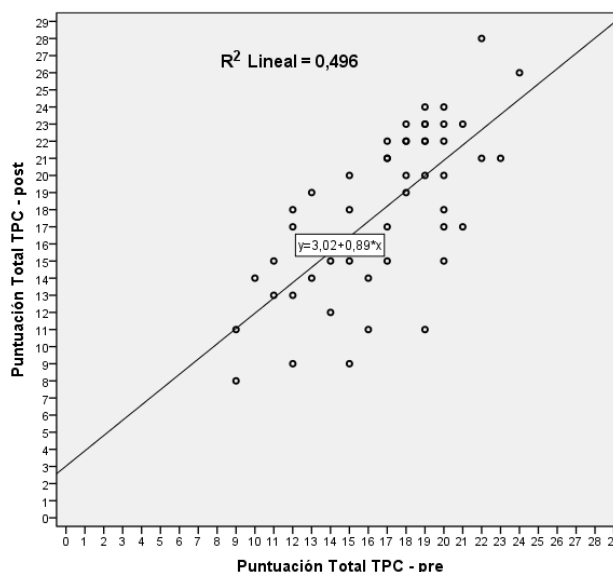


Figura 6.33. Fiabilidad del TPC como estabilidad temporal

6.5.2.3. *Validez*

Estudiaremos la validez del TPC desde cuatro puntos de vista: validez criterial (concurrente y predictiva), validez discriminante, validez convergente y validez factorial.

6.5.2.3.1. *Validez criterial*

6.5.2.3.1.1. *Concurrente*

Comenzamos abordando los objetivos específicos de investigación $O_{3i} \dots O_{3i}$, es decir, estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada, respectivamente, de aptitud ‘razonamiento lógico’ (PMA-R), de aptitud ‘espacial’ (PMA-E), de aptitud ‘numérica’ (PMA-N), y de aptitud ‘verbal’ (PMA-V). En la Tabla 6.38 se muestran las correlaciones entre las puntuaciones totales en nuestro TPC y las puntuaciones totales (directas) en los distintos tests de la batería PMA (TEA Ediciones, 2015a, en línea)

Tabla 6.38. Matriz de correlaciones del TPC con el PMA-R, PMA-E, PMA-N, y PMA-V

	PMA-R	PMA-E	PMA-N	PMA-V
TPC	,442**	,439**	-,157	,273**
PMA-R		,356**	-,030	,334**
PMA-E			-,164*	,225**
PMA-N				-,020

141 ≤ N ≤ 166

** Correlación significativa al nivel $p_{(r)} < 0,01$

* Correlación significativa al nivel $p_{(r)} < 0,05$

Tal y como puede observarse en la tabla anterior, el TPC presenta una correlación estadísticamente muy significativa ($p_{(r)} < 0,01$), positiva y de intensidad moderada con el PMA-R (factor ‘razonamiento’) y con el PMA-E (factor ‘espacial’). Por otro lado, el TPC no correlaciona de manera estadísticamente significativa ($p_{(r)} > 0,05$) con el PMA-N (factor ‘numérico’). Finalmente, el TPC presenta una correlación estadísticamente muy significativa ($p_{(r)} < 0,01$), positiva pero de intensidad sólo baja con el PMA-V (factor ‘verbal’).

En las siguientes Figuras 6.34, 6.35, 6.36, y 6.37 se ilustran los diagramas de dispersión correspondientes, que incluyen los respectivos coeficientes de determinación R^2 . Así, por ejemplo, el coeficiente de determinación del TPC sobre el PMA-R es $R^2 = 0,195$; lo que indica que el 19,5% de la varianza de las puntuaciones en el PMA-R puede ser explicado por la varianza de las puntuaciones en el TPC, y viceversa (varianza conjunta).

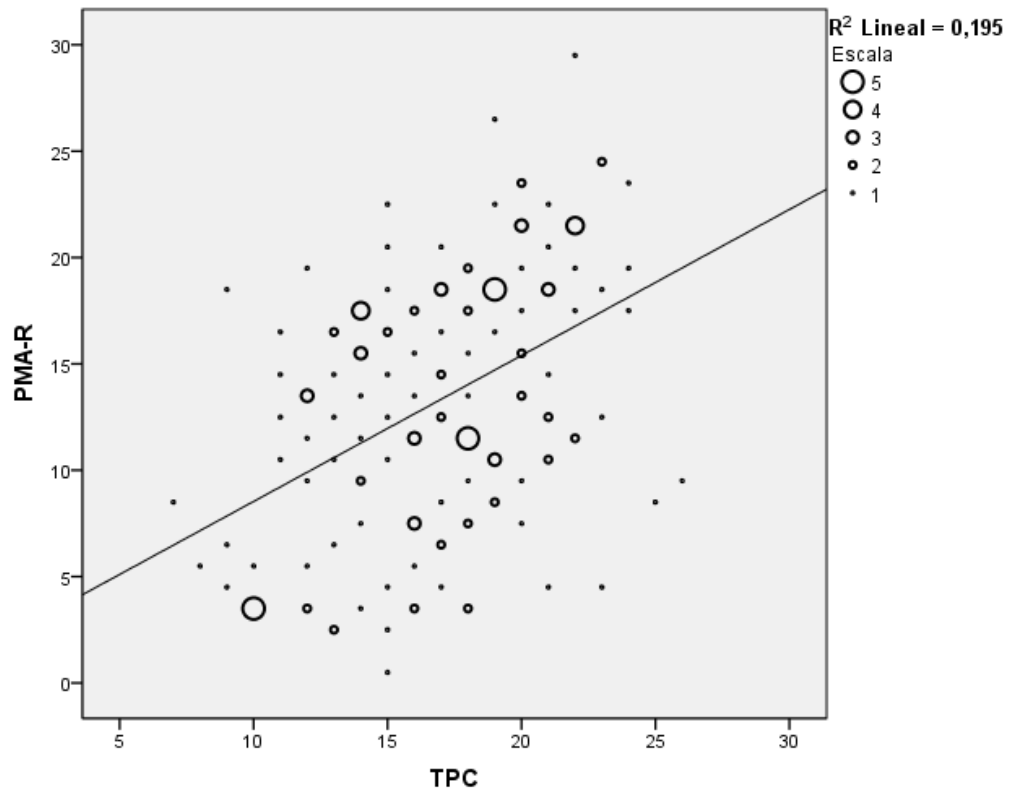


Figura 6.34. Diagrama de dispersión TPC * PMA-R

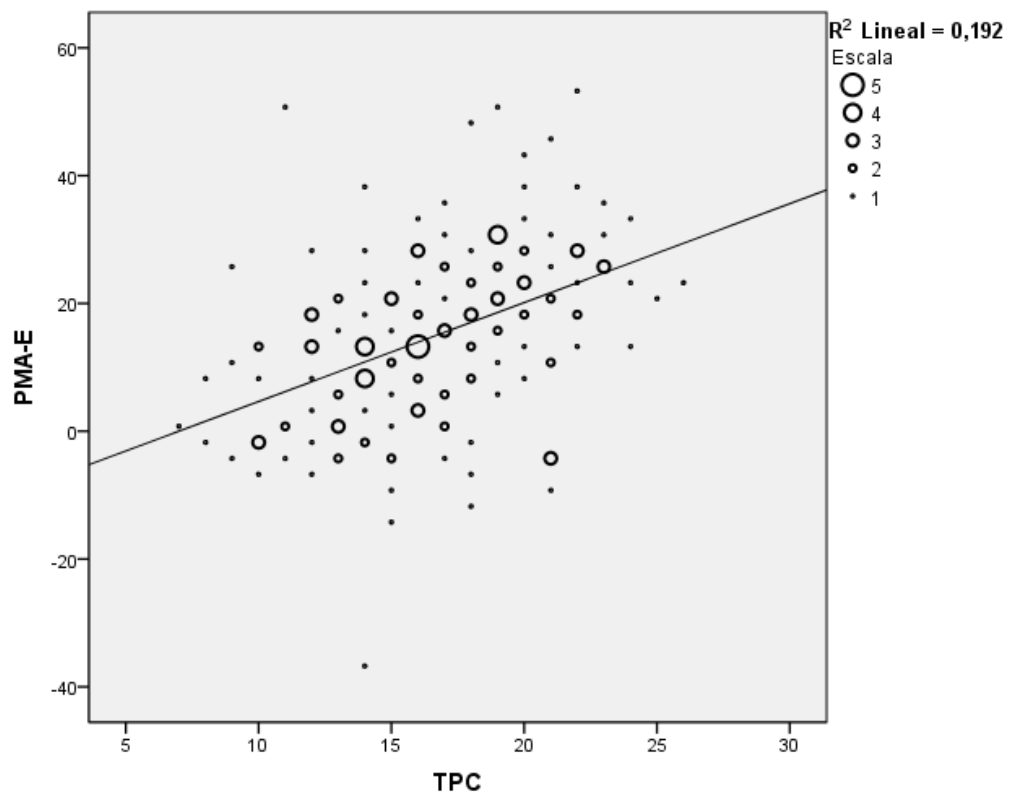


Figura 6.35. Diagrama de dispersión TPC * PMA-E

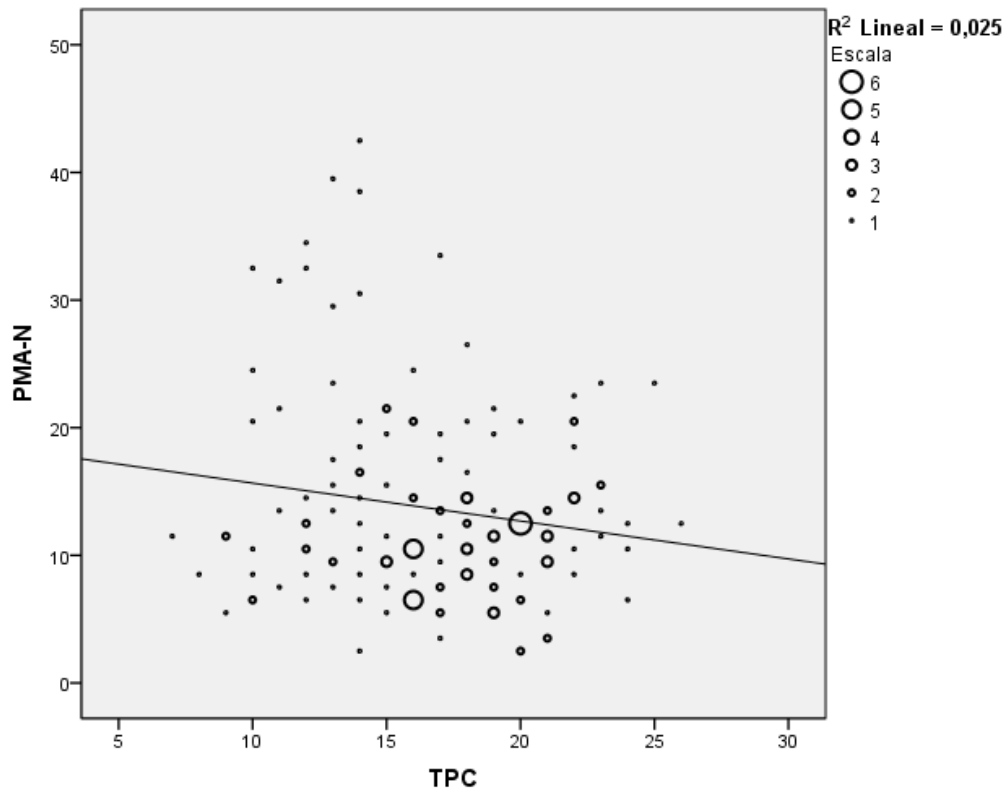


Figura 6.36. Diagrama de dispersión TPC * PMA-N

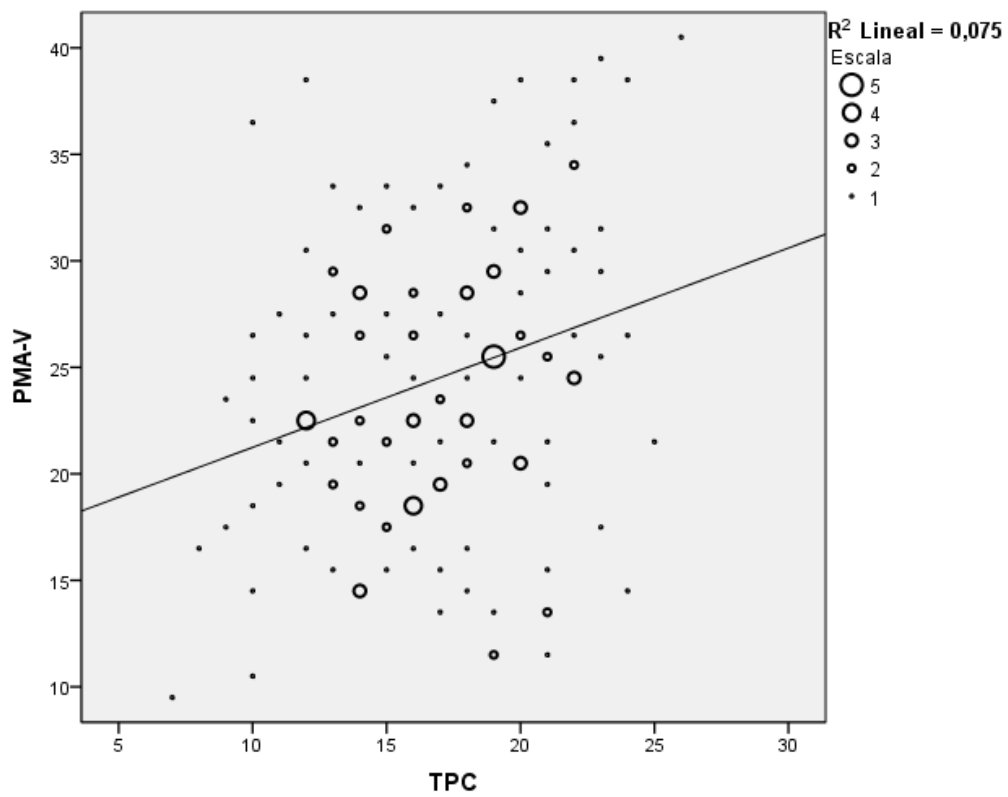


Figura 6.37. Diagrama de dispersión TPC * PMA-V

En este punto, consideramos de máximo interés realizar una regresión múltiple sobre el TPC (tomada como variable dependiente) a partir de las puntuaciones en el PMA-R, PMA-E, PMA-N y PMA-V (tomadas como variables predictoras; en ese sentido se utilizará el término ‘predecir’). En la siguiente Tabla 6.39 se presenta el resumen del modelo de regresión, calculado según el método ‘introducir’. Tal y como puede comprobarse, el modelo construido a partir de las puntuaciones en la batería PMA correlaciona $r = +0,540$ con el TPC; lo que supone un R^2_{ajustado} de 0,27 (el 27,0% de la varianza del TPC es explicado a partir de una combinación lineal de las aptitudes primarias medidas a través del PMA). Se comprobó la normalidad de los residuos del modelo de regresión.

Tabla 6.39. Resumen del modelo^b de regresión sobre el TPC a partir de las puntuaciones en la batería PMA

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	,540 ^a	,291	,270	3,391

a. Variables predictoras: (Constante), PMA-V, PMA-N, PMA-E, PMA-R

b. Variable dependiente: TPC

Tal y como puede verse en la siguiente Tabla 6.40, el modelo de regresión es capaz de explicar-predecir, de manera estadísticamente muy significativa, las diferencias en las puntuaciones de la variable dependiente (TPC), dado que $F = 13,457$ ($p_{(F)} = 0,000 < 0,01$).

Tabla 6.40. ANOVA del TPC según el modelo de regresión construido a partir de la batería PMA

Modelo	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Regresión	619,013	4	154,753	13,457**	,000
1 Residual	1506,517	131	11,500		
Total	2125,529	135			

** Significativo al nivel $p_{(F)} < 0,01$

Ahora bien, según vemos en la siguiente Tabla 6.41 de coeficientes del modelo de regresión, sólo el PMA-R (factor ‘razonamiento’) y el PMA-E (‘factor espacial’) son capaces, de manera específica y estadísticamente muy significativa ($p_{(t)} < 0,01$), de explicar-predecir cambios en la variable dependiente (TPC). Los coeficientes tipificados del modelo son, de mayor a menor peso relativo, $\beta_{(PMA-E)} = 0,308$, $\beta_{(PMA-R)} = 0,265$; $\beta_{(PMA-V)} = 0,134$; $\beta_{(PMA-N)} = -0,051$.

Tabla 6.41. Coeficientes^a del modelo de regresión sobre el TPC a partir de la batería PMA

Modelo	Coeficientes no estandarizados		Coeficientes tipificados	t	Sig.
	B	Error típ.	Beta		
(Constante)	11,793	1,309		9,006**	,000
PMA-R	,169	,052	,265	3,253**	,001
1 PMA-E	,089	,023	,308	3,865**	,000
PMA-N	-,027	,039	-,051	-0,688	,493
PMA-V	,079	,046	,134	1,715	,089

a. Variable dependiente: TPC

** Significativo al nivel $p_{(t)} < 0,01$

En la siguiente Figura 6.38 se muestran los gráficos de regresión parcial sobre el TPC de los 2 predictores que se han mostrado estadísticamente significativos (PMA-R y PMA-E). Y en la Figura 6.39, como síntesis, se muestra el diagrama de dispersión entre las puntuaciones totales observadas en el TPC y las puntuaciones predichas por el modelo de regresión calculado.

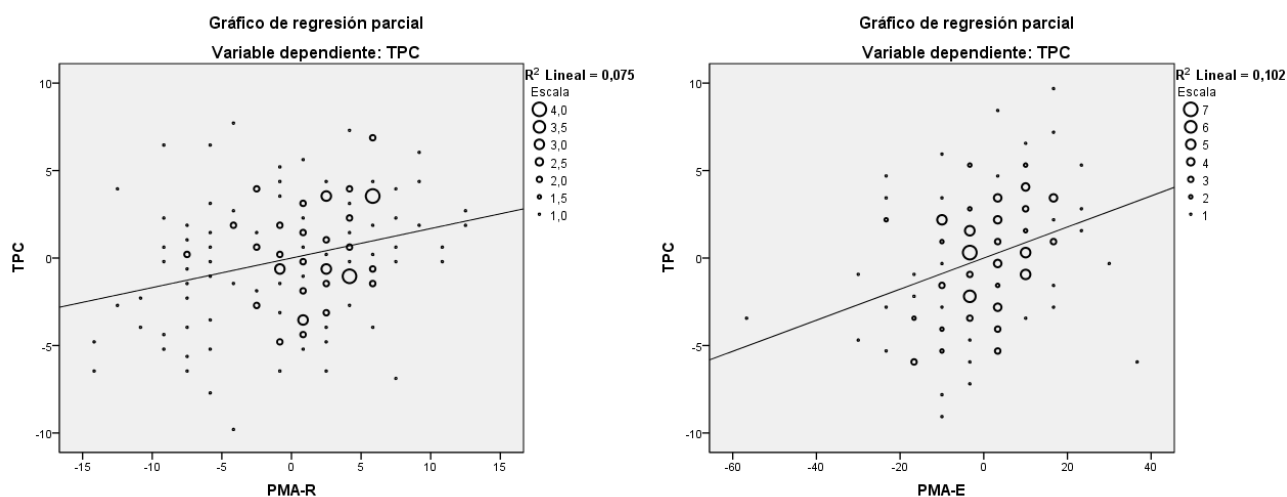


Figura 6.38. Gráficos de regresión parcial de PMA-R sobre TPC (izquierda), y PMA-E sobre TPC (derecha)

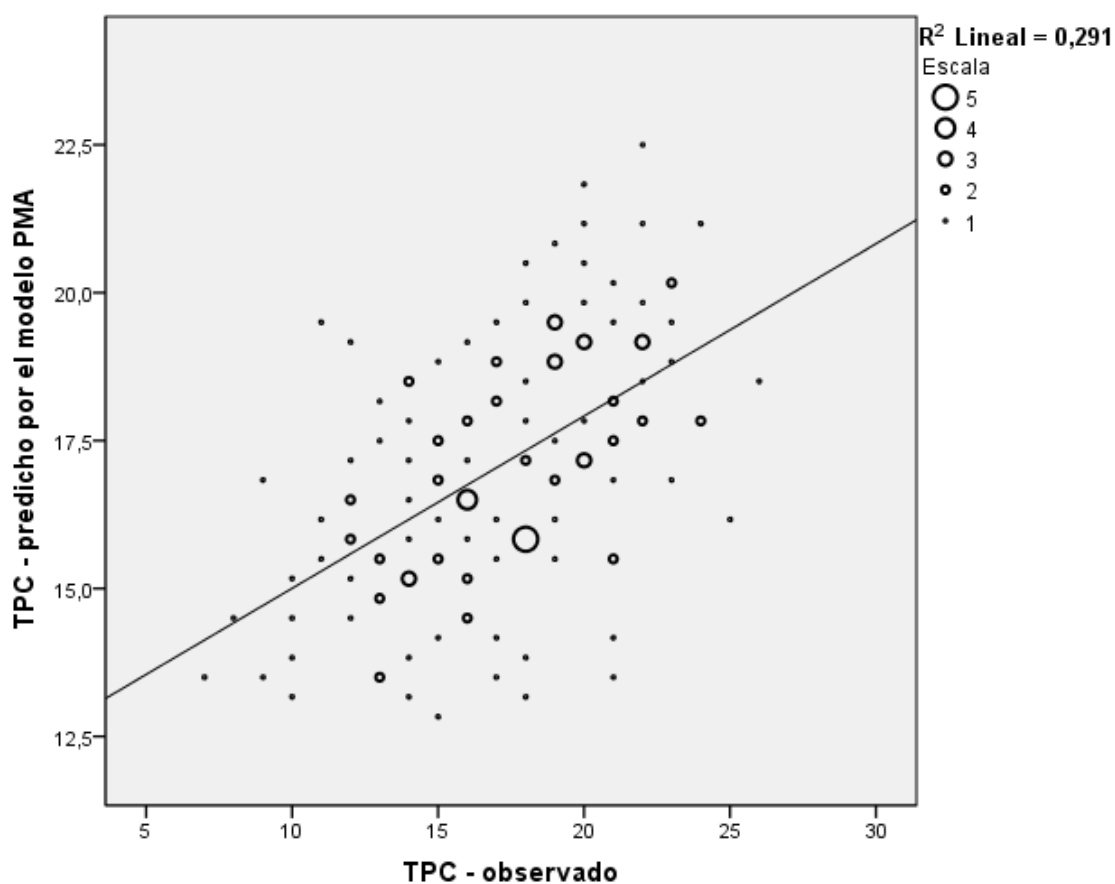


Figura 6.39. Diagrama de dispersión entre las puntuaciones observadas TPC y las predichas por el modelo PMA

Pasamos ahora a abordar el objetivo específico O_{3m} , es decir, a estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘resolución de problemas’, el Test de Resolución de Problemas RP30 (TEA Ediciones, 2015b, en línea). Así, en la Tabla 6.42 se muestra la correlación entre las puntuaciones totales en el TPC y las puntuaciones totales (directas) en el RP30.

Tabla 6.42. Correlación del TPC con el RP30

		RP30
TPC	Correlación de Pearson	,669**
	Sig. (bilateral)	,000
	N	56

** Correlación significativa al nivel $p(r) < 0,01$

Tal y como puede observarse, existe una correlación estadísticamente muy significativa, de signo positivo y de intensidad moderada-alta ($r = +0,669$) entre el TPC y el RP30. En la Figura 6.40 se ilustra el correspondiente diagrama de dispersión, y el coeficiente de determinación $R^2 = 0,447$ (44,7% de varianza compartida entre ambas puntuaciones)

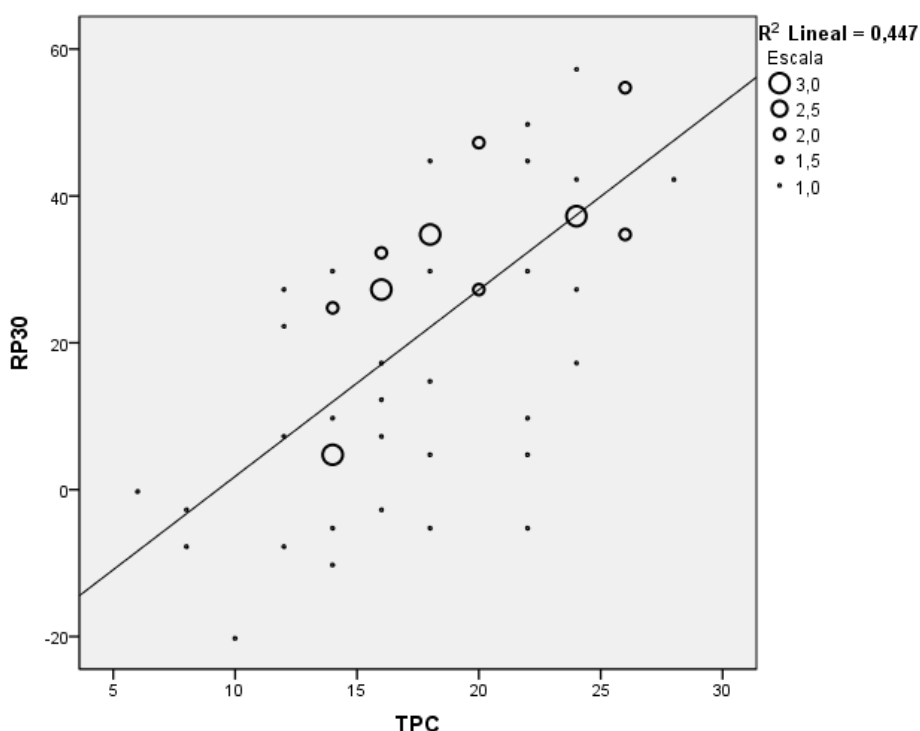


Figura 6.40. Diagrama de dispersión TPC * RP30

Con respecto al objetivo específico O_{3n} , es decir, estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de aptitud ‘perceptivo-atencional’, el Test Formas Idénticas-Revisadas (FI-R) (TEA Ediciones, 2015c, en línea), en la siguiente Tabla 6.43 se muestran las correlaciones entre las puntuaciones totales en el TPC, y las dos puntuaciones totales (directas) que arroja la aplicación del test FI-R: el ‘Índice de Percepción y Atención’ (A-E) y el ‘Índice de Control de la Impulsividad’ (ICI). Tal y como puede observarse en la tabla, aunque se encuentran con ambos índices correlaciones positivas de intensidad baja, ninguna de dichas correlaciones llega a ser estadísticamente significativa ($p(r) > 0,05$).

Tabla 6.43. Correlación del TPC con el FI-R (índices A-E e ICI)

		FI-R (A-E)	FI-R (ICI)
TPC	Correlación de Pearson	,229	,204
	Sig. (bilateral)	,076	,115
	N	61	61

Finalmente, para cerrar este sub-epígrafe, abordamos el objetivo específico **O_{3n}**, es decir, estudiar la validez criterial (concurrente) del TPC con respecto a una medida estandarizada de personalidad, el cuestionario BFQ-NA (TEA Ediciones, 2015d, en línea). En la Tabla 6.44 se muestra la matriz de correlaciones entre las puntuaciones totales en el TPC y las puntuaciones totales (directas) en cada una de las cinco dimensiones de personalidad que mide el BFQ-NA: Conciencia (Co), Apertura (Ap), Extraversión (Ex), Amabilidad (Am), e Inestabilidad Emocional (Ie).

Tabla 6.44. Matriz de correlaciones (N=99) entre el TPC y las dimensiones de personalidad del BFQ-NA

	BFQ-NA (Co)	BFQ-NA (Ap)	BFQ-NA (Ex)	BFQ-NA (Am)	BFQ-NA (Ie)
TPC	,267**	,407**	,304**	,133	,092
BFQ-NA (Co)		,834**	,605**	,753**	-,374**
BFQ-NA (Ap)			,444**	,611**	-,202*
BFQ-NA (Ex)				,645**	-,234*
BFQ-NA (Am)					-,363**

N=99

** Correlación significativa al nivel $p(r) < 0,01$

* Correlación significativa al nivel $p(r) < 0,05$

Tal y como puede observarse en la tabla anterior, el TPC correlaciona de manera estadísticamente muy significativa ($p(r) < 0,01$), en sentido positivo, con intensidad moderada con la dimensión ‘Apertura’ ($r = +0,407$) y con intensidad baja con las dimensiones ‘Extraversión’ ($r = +0,304$) y ‘Conciencia’ ($r = +0,267$). No se encuentran correlaciones estadísticamente significativas ni con la dimensión ‘Amabilidad’, ni con la dimensión ‘Inestabilidad Emocional’. Los correspondientes diagramas de dispersión, de menor a mayor R^2 , se muestran en las siguientes Figuras 6.41 a 6.43.

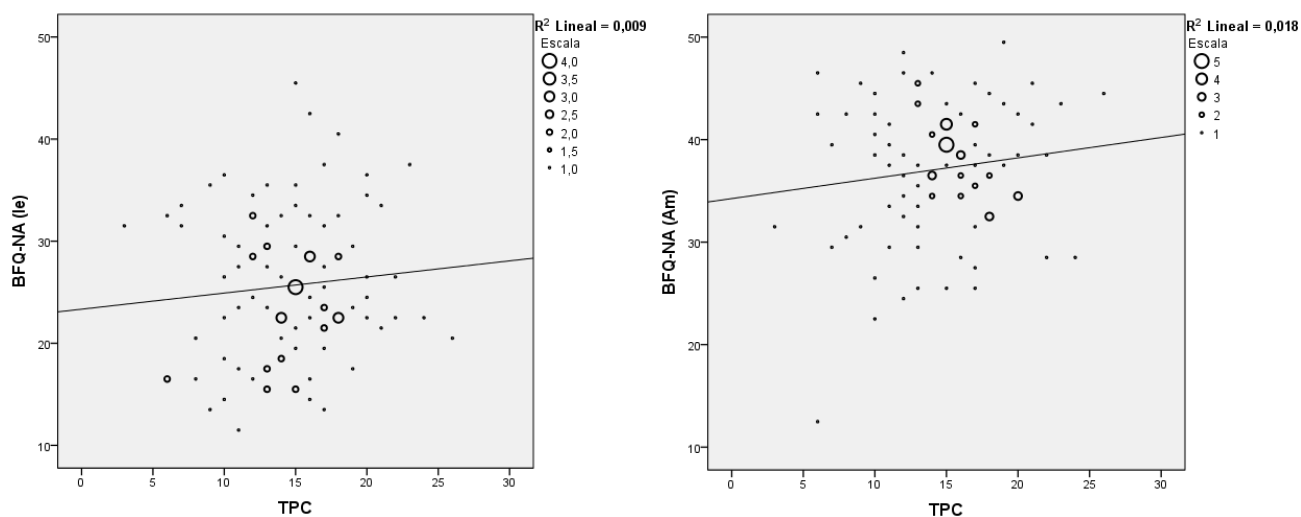


Figura 6.41. Diagrama de dispersión TPC*Inestabilidad Emocional (izquierda) y TPC*Amabilidad (derecha)

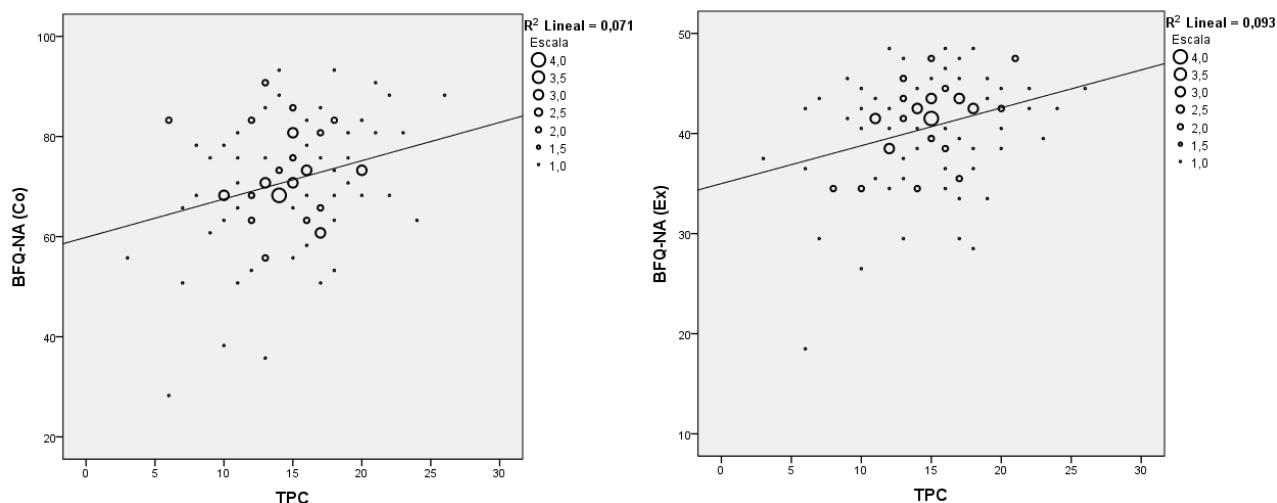


Figura 6.42. Diagrama de dispersión TPC*Conciencia (izquierda) y TPC*Extraversión (derecha)

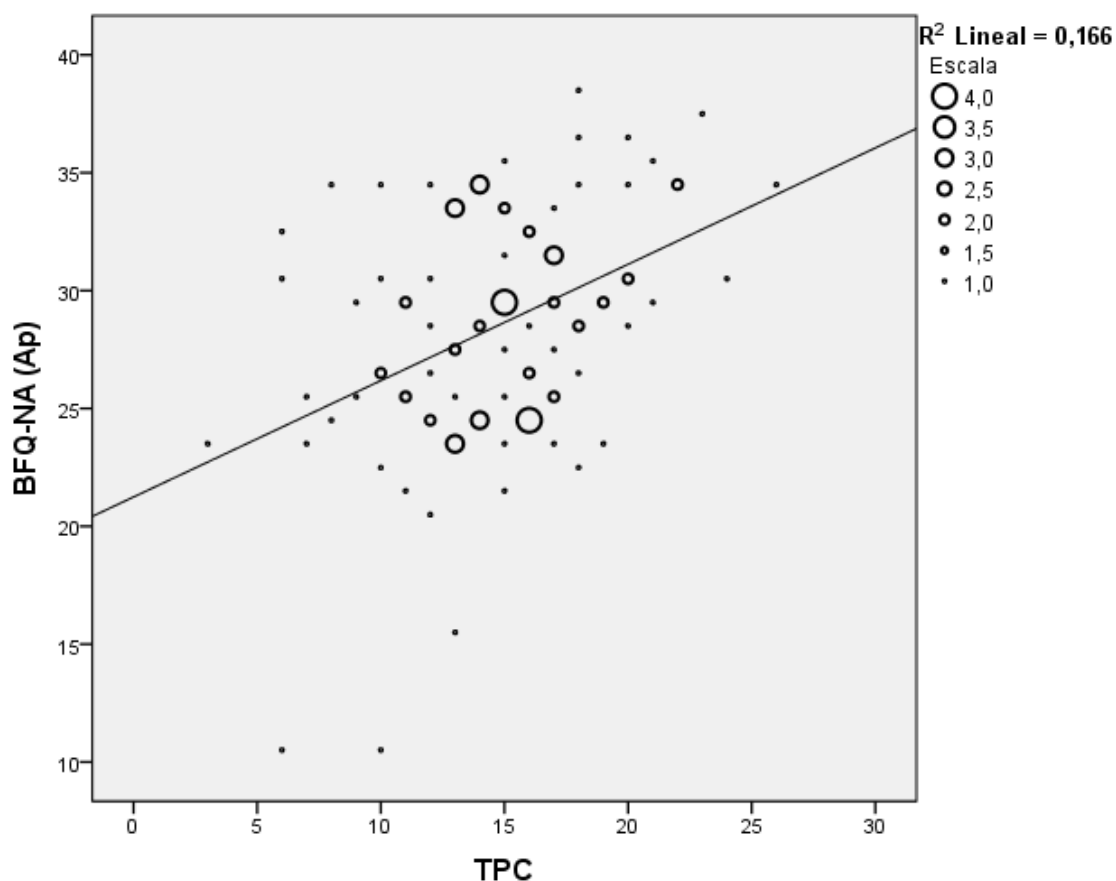


Figura 6.43. Diagrama de dispersión TPC*Apertura

Análogamente a lo que hicimos con la batería PMA, nos resulta de gran interés realizar una regresión múltiple sobre el TPC (tomada como variable dependiente) a partir de las puntuaciones en cada una de las dimensiones de personalidad medidas por el BFQ-NA (tomadas como predictores). En la siguiente Tabla 6.45 se presenta el resumen del modelo de regresión, que ha sido calculado según el método ‘introducir’. Tal y como puede comprobarse, el modelo construido a partir de las puntuaciones en las dimensiones del BFQ-NA correlaciona $r = +0,529$ con el TPC; lo que supone un $R^2_{ajustado} = 0,241$ (el 24,1% de la varianza del TPC es explicado a partir de una combinación lineal de

las dimensiones de personalidad medidas a través del BFQ-NA). Se comprobó la normalidad de los residuos del modelo de regresión.

Tabla 6.45. Resumen del modelo^b de regresión sobre el TPC a partir de las puntuaciones en el cuestionario BFQ-NA

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	,529 ^a	,280	,241	3,663

a. Variables predictoras: (Constante), BFQ-NA (Ie), BFQ-NA (Ap), BFQ-NA (Ex), BFQ-NA (Am), BFQ-NA (Co)

b. Variable dependiente: TPC

Tal y como puede verse en la siguiente Tabla 6.46, el modelo de regresión es capaz de explicar-predecir, de manera estadísticamente muy significativa, las diferencias en las puntuaciones de la variable dependiente (TPC), dado que $F = 7,216$ ($p_{(F)} = 0,000 < 0,01$).

Tabla 6.46. ANOVA del TPC según el modelo de regresión construido a partir del cuestionario BFQ-NA

Modelo	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Regresión	484,038	5	96,808	7,216**	,000
1 Residual	1247,598	93	13,415		
Total	1731,636	98			

** Significativo al nivel $p_{(F)} < 0,01$

Ahora bien, según vemos en la siguiente Tabla 6.47 de coeficientes del modelo de regresión, sólo la dimensión ‘Apertura’ (Ap) y la dimensión ‘Extraversión’ (Ex) son capaces, de manera específica y estadísticamente muy significativa ($p_{(t)} < 0,01$), de explicar-predecir cambios en la variable dependiente (TPC). Los coeficientes tipificados del modelo son, de mayor a menor peso relativo, $\beta_{(Ap)} = 0,618$, $\beta_{(Ex)} = 0,365$; $\beta_{(Am)} = -0,270$; $\beta_{(Co)} = -0,220$, y $\beta_{(Ie)} = 0,122$.

Tabla 6.47. Coeficientes^a del modelo de regresión sobre el TPC a partir del cuestionario BFQ-NA

Modelo	Coeficientes no estandarizados		Coeficientes tipificados	t	Sig.
	B	Error típ.	Beta		
(Constante)	-1,472	3,785		-,389	,698
1 BFQ-NA (Co)	-,077	,072	-,220	-1,066	,289
BFQ-NA (Ap)	,510	,136	,618	3,745**	,000
BFQ-NA (Ex)	,294	,096	,365	3,050**	,003
BFQ-NA (Am)	-,180	,097	-,270	-1,861	,066
BFQ-NA (Ie)	,071	,057	,122	1,243	,217

a. Variable dependiente: TPC

** Significativo al nivel $p_{(t)} < 0,01$

En la siguiente Figura 6.44 se muestran los gráficos de regresión parcial sobre el TPC de los 2 predictores que se han mostrado estadísticamente significativos: BFQ-NA (Ap), y BFQ-NA (Ex). Y en la Figura 6.45, como síntesis, se muestra el diagrama de dispersión entre las puntuaciones totales observadas en el TPC y las puntuaciones predichas por el modelo de regresión calculado.

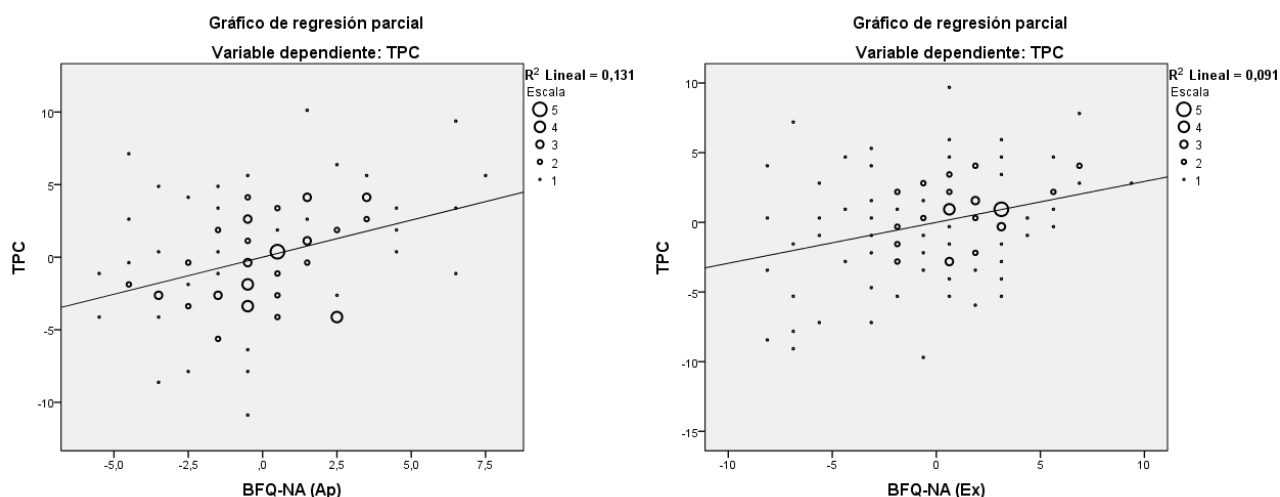


Figura 6.44. Gráficos de regresión parcial de BFQ-NA (Ap) sobre TPC (izquierda), y BFQ-NA (Ex) sobre TPC (derecha)

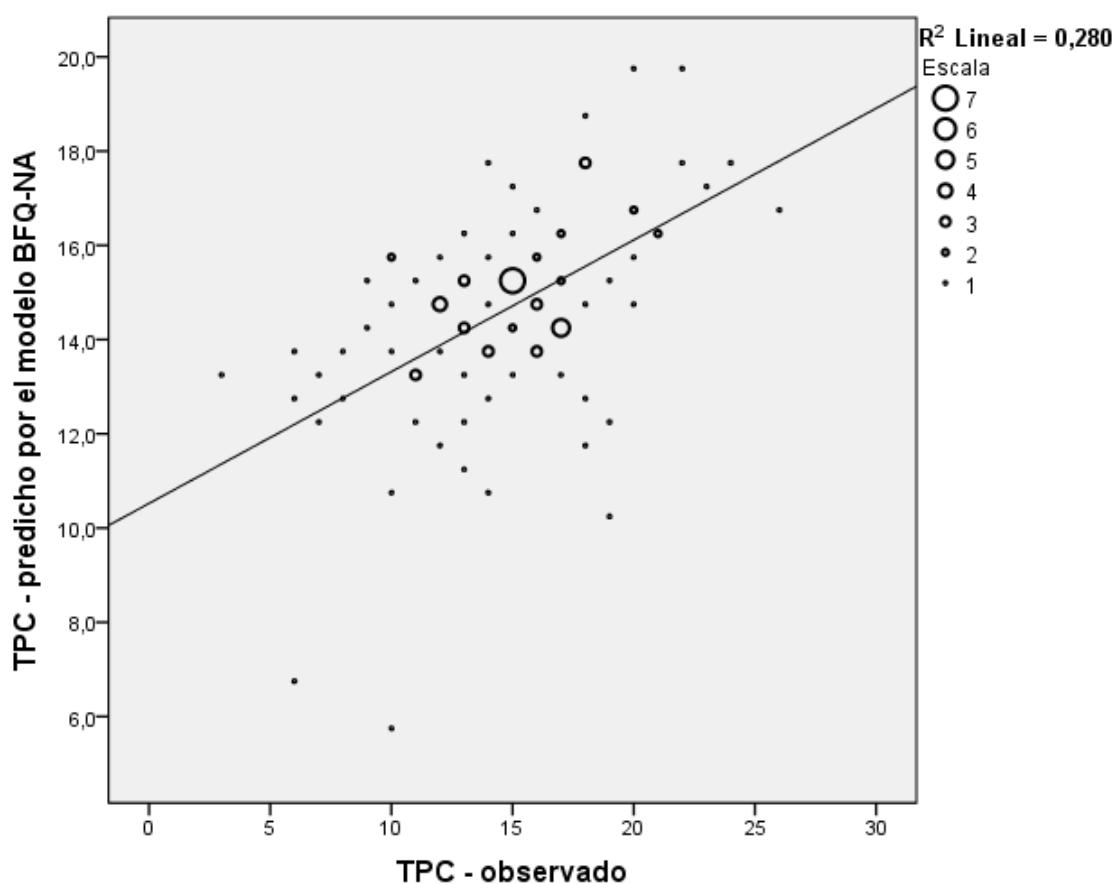


Figura 6.45. Diagrama de dispersión entre las puntuaciones observadas TPC y las predichas por el modelo BFQ-NA

6.5.2.3.1.2. Predictiva

Con respecto al objetivo específico O_{30} , es decir, estudiar la validez criterial (predictiva) del TPC con respecto al rendimiento académico (RA) en varias asignaturas; en la Tabla 6.48 se presentan las correlaciones entre las puntuaciones totales en el TPC (tomadas, en situación *pre*, a comienzo del

trimestre escolar) y las calificaciones académicas en las asignaturas de Informática, Matemáticas y Lengua (emitidas al final de dicho trimestre escolar, es decir, unas 12 semanas más tarde).

Tabla 6.48. Matriz de correlaciones (N=158) entre el TPC (al comienzo del trimestre) y las calificaciones en Informática, Matemáticas y Lengua (al final del trimestre)

	Nota (Informática)	Nota (Matemáticas)	Nota (Lengua)
TPC	,412**	,333**	,361**
Nota (Informática)		,597**	,523**
Nota (Matemáticas)			,689**

N=158

** Correlación significativa al nivel $p(r) < 0,01$

Tal y como puede observarse en la tabla anterior, el TPC correlaciona en sentido positivo y de manera estadísticamente muy significativa ($p(r) < 0,01$) con las calificaciones académicas de las tres asignaturas: con intensidad moderada en la asignatura de Informática ($r = +0,412$) y con intensidad baja en las asignaturas de Lengua ($r = +0,361$) y Matemáticas ($r = +0,333$). Los correspondientes diagramas de dispersión, de menor a mayor R^2 , se muestran en las siguientes Figuras 6.46 a 6.48.

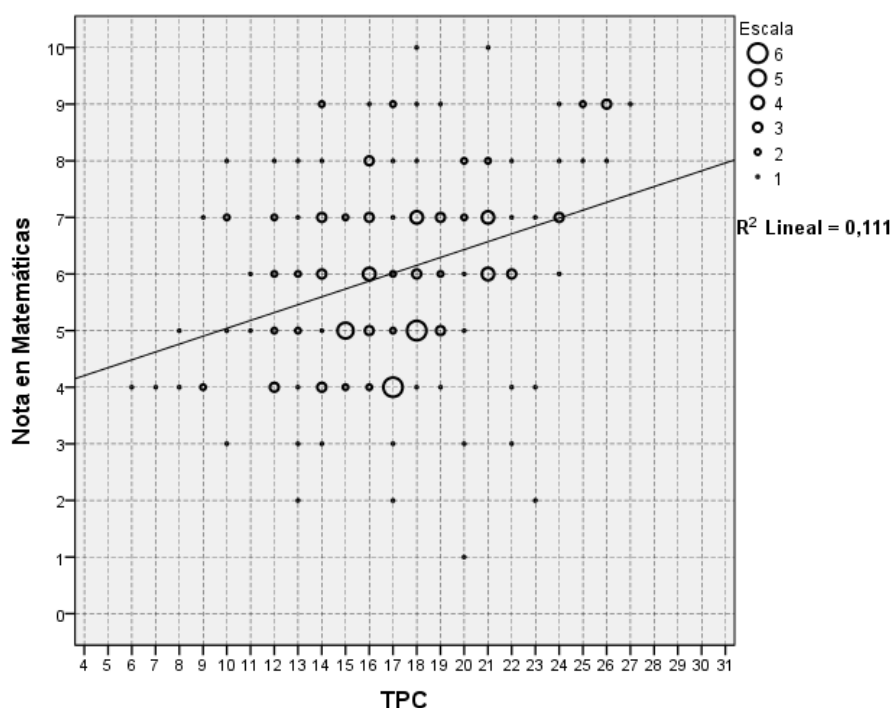


Figura 6.46. Diagrama de dispersión TPC*Nota en Matemáticas

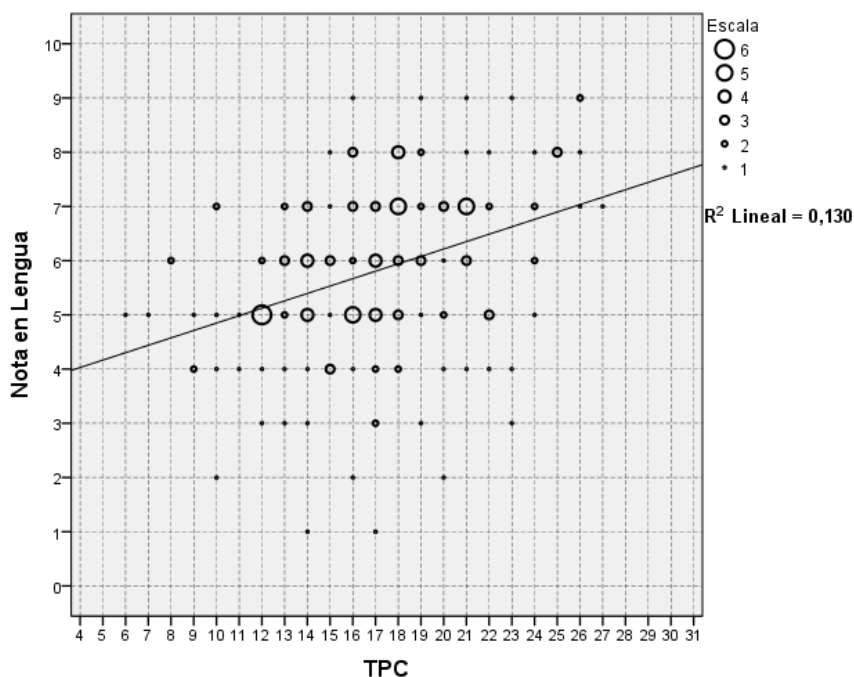


Figura 6.47. Diagrama de dispersión TPC*Nota en Lengua

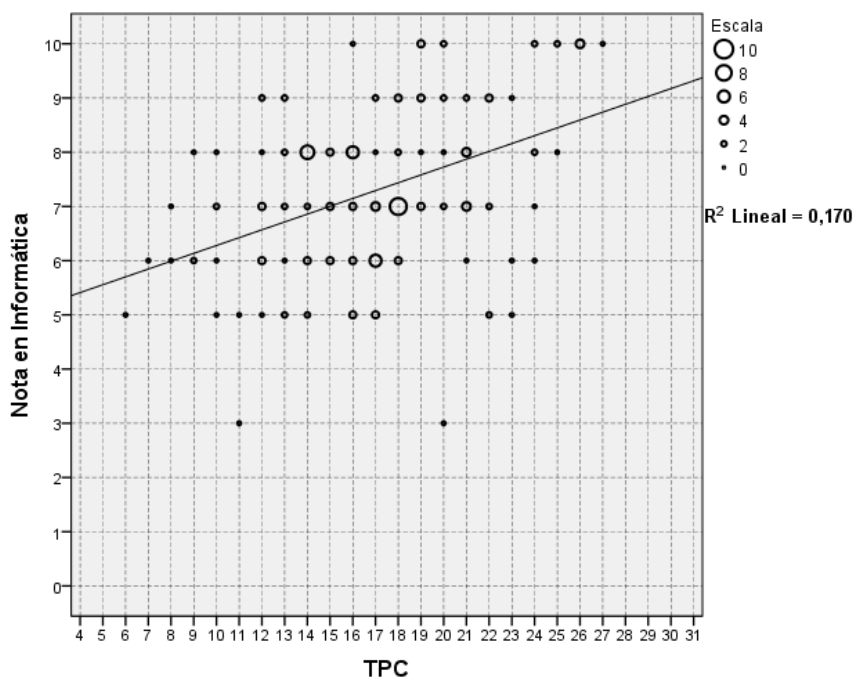


Figura 6.48. Diagrama de dispersión TPC*Nota en Informática

Podemos construir un modelo de regresión múltiple sobre el TPC a partir de las calificaciones en las tres asignaturas, que arroja un $R^2_{ajustado} = 0,184$. Dicho modelo es capaz de explicar-predecir, de manera retrospectiva y estadísticamente muy significativa, las puntuaciones totales en el TPC a partir de las calificaciones en las tres asignaturas, emitidas 12 semanas más tarde ($F = 12,773$; $p_{(F)} < 0,01$); aunque sólo la nota de Informática ($\beta_t = 0,300$) es la que, de manera específica y estadísticamente muy significativa ($p_{(t)} < 0,01$) explica-predice diferencias en el TPC. Se detalla en las siguientes Tablas 6.49 a 6.51.

Tabla 6.49. Resumen del modelo^b de regresión sobre el TPC (al comienzo del trimestre) a partir de las calificaciones en Informática, Matemáticas y Lengua (al final del trimestre)

Modelo	R	R cuadrado	R cuadrado corregida	Error típ. de la estimación
1	,446 ^a	,199	,184	3,924

a. Variables predictoras: (Constante), Nota Lengua, Nota Informática, Nota Matemáticas

b. Variable dependiente: TPC

Tabla 6.50. ANOVA del TPC según el modelo de regresión construido a partir de las calificaciones académicas

Modelo	Suma de cuadrados	gl	Media cuadrática	F	Sig.
Regresión	590,151	3	196,717	12,773**	,000
1 Residual	2371,824	154	15,401		
Total	2961,975	157			

** Significativo al nivel $p_{(F)} < 0,01$

Tabla 6.51. Coeficientes^a del modelo de regresión sobre el TPC a partir de las calificaciones académicas

Modelo	Coeficientes no estandarizados		Coeficientes tipificados	t	Sig.
	B	Error típ.	Beta		
(Constante)	7,539	1,577		4,781**	,000
1 Nota Informática	,855	,261	,300	3,273**	,001
Nota Matemáticas	,060	,258	,025	,234	,815
Nota Lengua	,492	,268	,186	1,837	,068

a. Variable dependiente: TPC

** Significativo al nivel $p_{(t)} < 0,01$

En la siguiente Figura 6.49 se muestra el diagrama de dispersión de las puntuaciones totales observadas en el TPC (al comienzo del trimestre) y las predichas por el modelo de regresión a partir de las calificaciones académicas en Informática, Matemáticas y Lengua (al finalizar dicho trimestre).

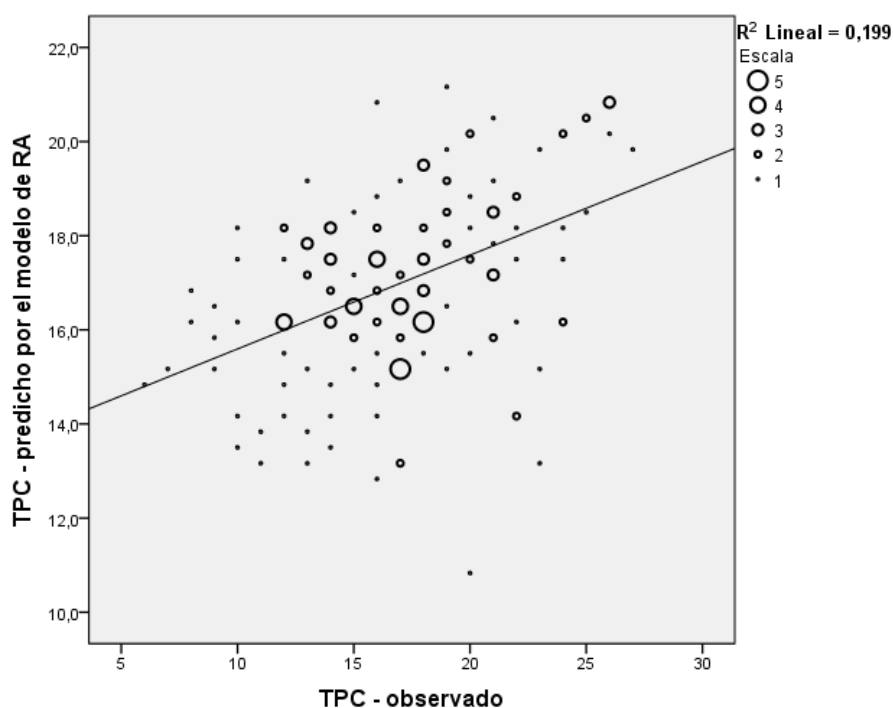


Figura 6.49. Diagrama de dispersión entre las puntuaciones observadas TPC y las predichas por el modelo RA

Finalmente es de destacar que, dentro de los 158 sujetos de los cuales poseemos datos en el TPC y en RA, con los que hemos calculado las anteriores correlaciones y modelo de regresión: hay 138 sujetos *experimentales* que posteriormente al TPC realizaron el curso trimestral ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) cuyo contenido gira alrededor del pensamiento computacional y que se evalúa en el Capítulo 7; y 20 sujetos *control* que posteriormente al TPC siguieron con el currículum habitual en la asignatura de Informática, basado en el uso de herramientas ofimáticas y navegación web (currículum TIC tradicional). Por tanto, la calificación en Informática al final del trimestre tiene un significado distinto para ambos grupos de sujetos, y parece conveniente calcular la correlación predictiva TPC * Nota de Informática de manera segmentada para cada uno de los grupos. Tal y como puede observarse en la Tabla 6.52, sólo aparece una correlación estadísticamente muy significativa entre el TPC (aplicado a comienzos del trimestre) y la nota en Informática (emitida al final del trimestre), cuando el currículum cursado durante dicho periodo tiene contenidos relativos al desarrollo del pensamiento computacional ($r = +0,483$, $p_{(r)} < 0,01$); y no cuando se cursa el currículum TIC tradicional ($r = -0,003$; $p_{(r)} > 0,05$).

Tabla 6.52. Correlación TPC*Nota en Informática, en función del currículum seguido

		Nota (Informática)
TPC (sujetos de control – currículum TIC tradicional)	Correlación de Pearson	-,003
	N	20
TPC (sujetos experimentales – currículum K8 de Code.org)	Correlación de Pearson	,483**
	N	138

** Correlación significativa al nivel $p_{(r)} < 0,01$

Con respecto al objetivo específico **O_{3p}**, es decir, estudiar la validez criterial (predictiva) del TPC con respecto a las estadísticas de desempeño en el curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a); en la Tabla 6.53 se muestran las correlaciones entre las puntuaciones totales en el TPC (tomadas en situación *pre* al inicio del trimestre) y las dos estadísticas de desempeño que proporciona la plataforma Code.org sobre sus usuarios-aprendices (descargadas al final del curso trimestral). Estas dos estadísticas de desempeño son: número de ‘niveles-pantallas’ correctamente completadas por el sujeto, y número de ‘líneas de código escritas’,³⁸⁶

Tabla 6.53. Correlaciones entre las puntuaciones totales en el TPC y las estadísticas de desempeño en Code.org (N=324)

	Niveles Completados [Code.org]	Líneas de Código escritas [Code.org]
TPC	,331 **	,254 **
Niveles Completados [Code.org]		,765 **

N=324

** Correlación significativa al nivel $p_{(r)} < 0,01$

³⁸⁶ Nótese que, para superar un mismo nivel-pantalla, se pueden escribir diferentes programas con un número variable de líneas de código. Por eso, la correlación entre ‘Niveles completados’ y ‘Líneas de Código escritas’ no es perfecta. Por ejemplo, un programa que haga avanzar un objeto hacia adelante cuatro casillas puede escribirse como “[Repetir *4] [avanzar]” (que son 2 líneas de código); o como “[avanzar] [avanzar] [avanzar] [avanzar]” (que son 4 líneas de código). Volveremos sobre este detalle en el Capítulo 7.

Tal y como puede observarse, encontramos sendas correlaciones positivas y estadísticamente muy significativas ($p(r) < 0,01$) entre el TPC, tanto con el número de ‘niveles-pantallas completados’ ($r = +0,331$) como con el número de ‘líneas de código escritas’ ($r = +0,254$) (Figuras 6.50 y 6.51)

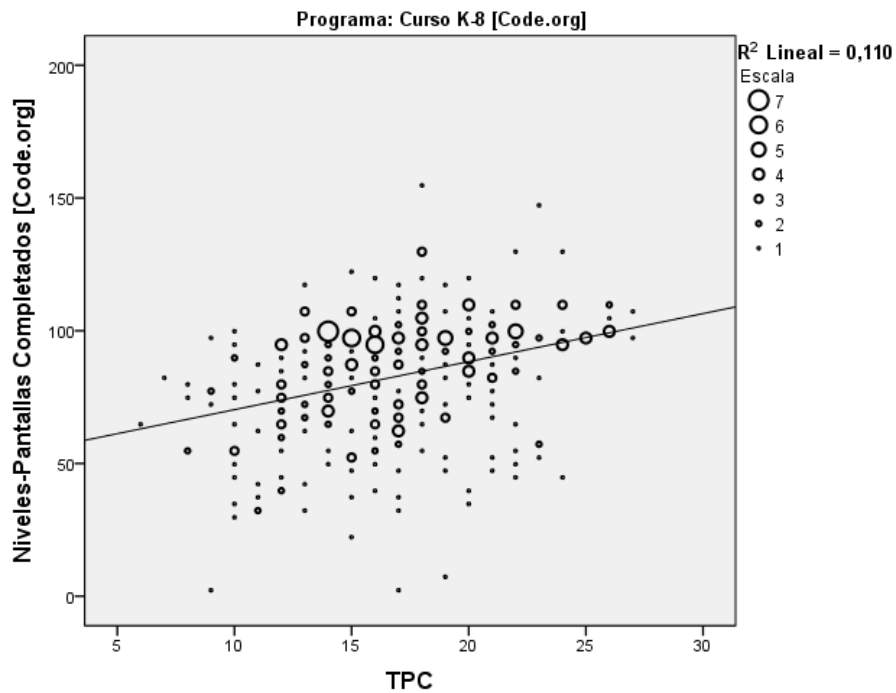


Figura 6.50. Diagrama de dispersión TPC* Niveles-Pantallas completados en Code.org

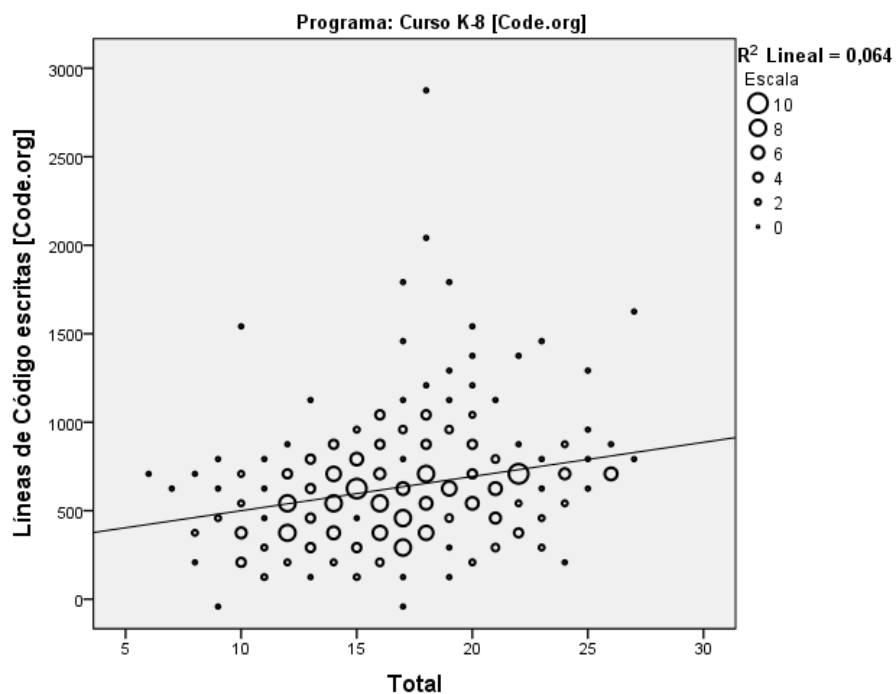


Figura 6.51. Diagrama de dispersión TPC* Líneas de código escritas en Code.org

Sin embargo, si segmentamos las correlaciones anteriores entre las aulas que siguieron el curso ‘K-8 Intro to Computer Science’ (Code.org, 2015a) según un sistema metodológico ‘teacher-paced’ (‘guiado por el profesor’, de manera que todo el grupo de clase va avanzando por las etapas del curso al ritmo homogéneo marcado por el docente), y las que lo hicieron según un sistema ‘self-paced’ (‘autoguiado’, de manera que cada sujeto va avanzando, incluso en su casa, por las etapas del curso a su propio ritmo); entonces los valores difieren entre ambos sistemas. Tal y como puede observarse en la siguiente Tabla 6.54 y Figura 6.52, la correlación es de mayor intensidad en el caso ‘self-paced’ (se eleva hasta $r = +0,458$)

Tabla 6.54. Correlación TPC * Niveles completados en Code.org, en función de la metodología de aula

		Niveles Completados [Code.org]
TPC – <i>Teacher-paced</i>	Correlación de Pearson	,330**
	N	279
TPC – <i>Self-paced</i>	Correlación de Pearson	,458**
	N	45

** Correlación significativa al nivel $p(r) < 0,01$

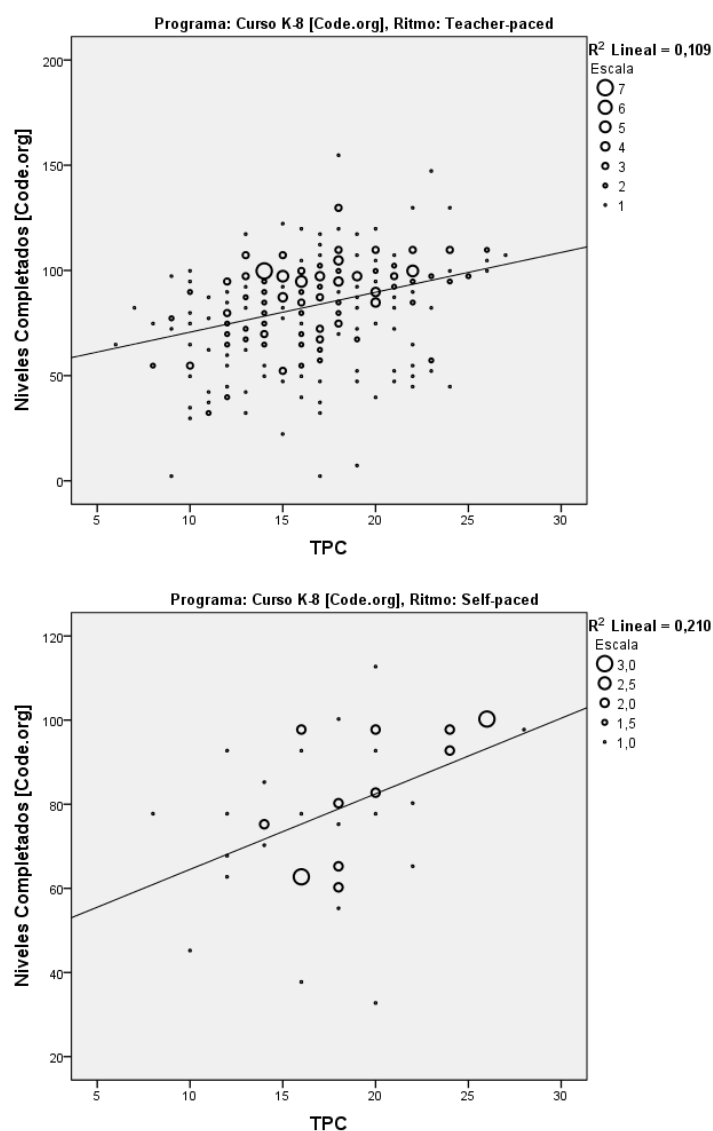


Figura 6.52. Diagrama de dispersión TPC*Niveles completados en Code.org, según sistema metodológico

6.5.2.3.2. Validez discriminante

Con respecto al objetivo específico O_{3q} , es decir, estudiar la validez discriminante del TPC para detectar sujetos de ‘alta capacidad computacional’; trabajaremos exclusivamente con la sub-muestra de sujetos que realizaron la aplicación *pre* del TPC, y posteriormente realizaron el curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) mediante una metodología ‘*self-paced*’ (‘autoguiada’ por el ritmo de cada sujeto); metodología que permitió emerger y manifestarse a la ‘alta capacidad computacional’, que es el criterio discriminante que utilizaremos y que definimos operativamente un poco más adelante.

Esta sub-muestra está compuesta por 45 estudiantes (N=45), 29 chicos (64,4%) y 16 chicas (35,6%). Todos los sujetos son de 1º Ciclo de la ESO, y provienen de: un aula de 1º ESO del IES María Blasco (17 sujetos), un aula de 1º ESO del IES Andreu Sempere (15 sujetos), y un aula de 2º ESO también del IES Andreu Sempere (13 sujetos).

Los sujetos categorizados como de ‘alta capacidad computacional’ son aquéllos que completaron el curso ‘*K-8 Intro to Computer Science*’³⁸⁷ (Code.org, 2015a) a un ritmo notablemente más rápido que sus compañeros de aula, y solicitaron acelerar a un curso de nivel superior: el denominado ‘*Computer Programming*’³⁸⁸ de la plataforma Khan Academy (2015). Atendiendo a este criterio, de los 45 sujetos de la sub-muestra: 7 sujetos (15,6%) sí fueron categorizados como de ‘alta capacidad computacional’ (2 del IES María Blasco y 5 del IES Andreu Sempere; todos ellos varones), mientras que los 38 sujetos restantes (84,4%) no lo fueron (grupo ‘normal’)

Al comparar las medias en la puntuación total en el TPC entre ambos grupos, obtenemos una diferencia estadísticamente significativa ($t = -2,300$; $p_{(t)} = 0,026 < 0,05$), y con un tamaño del efecto ‘grande’ ($d = 0,95$), a favor del grupo categorizado como de ‘alta capacidad computacional’. Es decir, la media en el TPC del grupo de ‘alta capacidad computacional’ se aparta casi una desviación típica de la media del grupo ‘normal’ Se detalla en la Tabla 6.55.

Tabla 6.55. Prueba *t* de diferencia de medias en el TPC entre el grupo ‘normal’ y el de ‘alta capacidad computacional’

	N	Media	Desviación estándar	<i>t</i>	$P_{(t)}$	<i>d</i> de Cohen	
TPC	‘ <i>Self-paced</i> ’- Grupo ‘normal’	38	17,45	4,360	-2,300*	,026	0,95
	‘ <i>Self-paced</i> ’- Grupo ‘alta capacidad computacional’	7	21,57	4,353			

*Significativo al nivel $p_{(t)} < 0,05$

Tal y como puede observarse, la media en el TPC para el grupo de ‘alta capacidad computacional’ fue de $\bar{X} = 21,57$; puntuación equivalente a casi un percentil 90 en el baremo de 1º Ciclo de la ESO según vimos en el subepígrafe de descriptivos.

Otro análisis de interés para estudiar la potencia discriminante del TPC para detectar-predecir los sujetos de ‘alta capacidad computacional’, tal y como los hemos definido, es la curva COR (acrónimo español del inglés *ROC: Receiver Operating Characteristic*) de rendimiento diagnóstico.

³⁸⁷ Recordemos que el curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) trabaja con un lenguaje de programación visual ‘por bloques’ (concretamente con Blockly)

³⁸⁸ Recordemos que el curso ‘*Computer Programming*’ (Khan Academy, 2015) trabaja con un lenguaje de programación textual (concretamente con Processing JavaScript); de complejidad superior.

La curva COR de rendimiento diagnóstico del TPC para discriminar sujetos de ‘alta capacidad computacional’ se muestra en la Figura 6.53; que se acompaña de las tablas asociadas de resultados, Tablas 6.56 y 6.57.

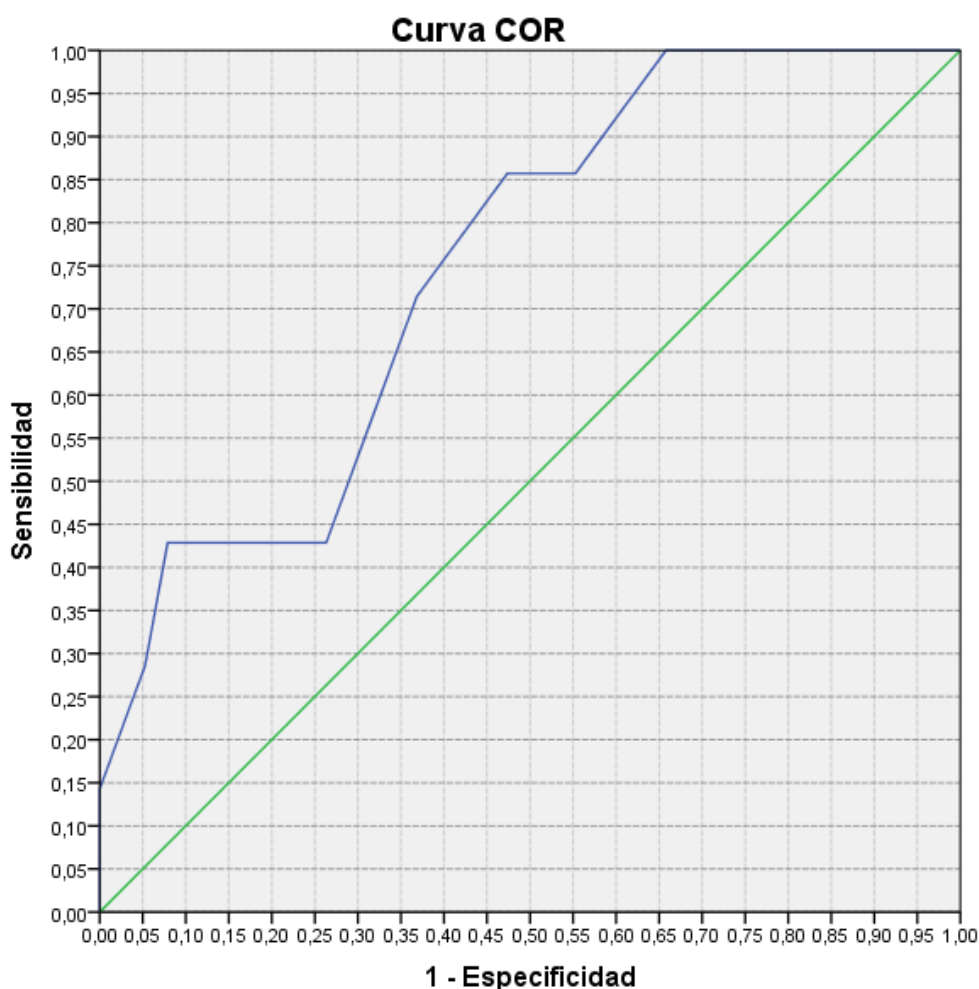


Figura 6.53. Curva COR de rendimiento diagnóstico del TPC sobre la ‘alta capacidad computacional’

Tabla 6.56. Área bajo la Curva COR de rendimiento diagnóstico del TPC sobre la ‘alta capacidad computacional’

Área bajo la curva				
Variable(s) de resultado de prueba: TPC				
Área	Error estándar ^a	Significación asintótica ^b	95% de intervalo de confianza asintótico	
			Límite inferior	Límite superior
,750*	,092	,037	,569	,931

a. Bajo el supuesto no paramétrico

b. Hipótesis nula: área verdadera = 0,5

* Significativa a un nivel $p < 0,05$

Tal y como puede observarse en la tabla anterior, el área bajo la curva COR resultante es $A_{COR} = 0,75$. Este resultado es estadísticamente significativo ($p = 0,037 < 0,5$); y puede interpretarse como una potencia discriminante ‘buena’ (Navas Ara, 2001) del TPC sobre la ‘alta capacidad computacional’.

En la Tabla 6.57 señalamos en fondo verde los puntos de corte de mayor potencia discriminante de nuestro TPC para detectar-predecir sujetos de ‘alta capacidad computacional’: son los puntos de corte que maximizan la ‘sensibilidad’ del TPC (% de sujetos de ‘alta capacidad computacional’ detectados correctamente o ‘verdaderos positivos’), minimizando su ‘especificidad’ (% de sujetos ‘normales’ clasificados incorrectamente como de ‘alta capacidad computacional’, o ‘falsos positivos’). Estos importantes puntos de corte, que se visualizan en la anterior curva COR como los vértices izquierdos en la misma, son:

- ✓ Puntuación $\geq 17,5$ en el TPC: detecta un 85,7% de ‘verdaderos positivos’ y un 47,4% de ‘falsos positivos’ en lo relativo a la ‘alta capacidad computacional’
- ✓ Puntuación $\geq 18,5$ en el TPC: detecta un 71,4% de ‘verdaderos positivos’ y un 36,8% de ‘falsos positivos’ en lo relativo a la ‘alta capacidad computacional’
- ✓ Puntuación $\geq 24,5$ en el TPC: detecta un 42,9% de ‘verdaderos positivos’ y un 7,9% de ‘falsos positivos’ en lo relativo a la ‘alta capacidad computacional’

Tabla 6.57. Coordenadas de la curva COR y puntos de corte de mayor potencia discriminante

Coordenadas de la curva		
Variable(s) de resultado de prueba: TPC		
Positivo si es mayor o igual que ^a	Sensibilidad	1 - Especificidad
7,00	1,000	1,000
9,00	1,000	,974
11,00	1,000	,947
13,00	1,000	,842
14,50	1,000	,763
15,50	1,000	,658
16,50	,857	,553
17,50	,857	,474
18,50	,714	,368
19,50	,571	,316
20,50	,429	,263
22,50	,429	,132
24,50	,429	,079
25,50	,286	,053
26,50	,143	,000
28,00	,000	,000

a. El valor de corte más pequeño es el valor mínimo de prueba observado menos 1 y el valor de corte más grande es el valor máximo de prueba observado más 1. Todos los demás valores de corte son los promedios de los dos valores de prueba observados solicitados consecutivos.

6.5.2.3.3. Validez convergente

Para estudiar la validez convergente de nuestro TPC, hemos utilizado otras dos medidas alternativas del pensamiento computacional: una selección de tareas extraídas del ‘Concurso Internacional Bebras’ (se revisó en el [sub-epígrafe 3.3.2.1](#) de la parte teórica); y la herramienta automática de análisis de proyectos Scratch, denominada ‘Dr. Scratch’ (se revisó en el [sub-epígrafe 3.3.2.2](#))

6.5.2.3.3.1. Con tareas Bebras

Abordando pues el objetivo específico de investigación O_{3r} , es decir, estudiar la validez convergente del TPC con respecto a una selección de tareas Bebras; en la Tabla 6.58 se muestran las correlaciones del TPC con las tareas B-1 (aplicadas concurrentemente al TPC en 3° Ciclo de Primaria), con las tareas B-2 (ídem en 1° Ciclo de Secundaria), y con las tareas B-3 (ídem en el 2° Ciclo de Secundaria). Ya nos referimos a esta selección de tareas Bebras en el sub-epígrafe relativo a ‘Instrumentos’ (6.5.1.2), y se muestran con detalle en el [Anexo H](#).

Tabla 6.58. Correlaciones del TPC con las Tareas Bebras

		Tarea B1-1	Tarea B1-2	Tarea B1-3	Total Tareas B1
TPC	Correlación de Pearson	,394**	,260*	,283*	,511**
	N	61	61	61	61
	% de sujetos que contestaron correctamente a la tarea	31,1%	34,4%	75,4%	
		Tarea B2-1	Tarea B2-2	Tarea B2-3	Total Tareas B2
TPC	Correlación de Pearson	,427**	,055	,483**	,523**
	N	179	179	179	179
	% de sujetos que contestaron correctamente a la tarea	44,7%	19,6%	46,9%	
		Tarea B3-1	Tarea B3-2	Tarea B3-3	Total Tareas B3
TPC	Correlación de Pearson	,283**	,028	,014	,175
	N	95	95	95	95
	% de sujetos que contestaron correctamente a la tarea	74,7%	50,5%	23,2%	

* Correlación significativa al nivel $p_{(r)} < 0,05$

** Correlación significativa al nivel $p_{(r)} < 0,01$

Tal y como puede observarse en la tabla anterior, las puntuaciones totales en el TPC correlacionan de manera estadísticamente muy significativa ($p_{(r)} < 0,01$), en sentido positivo y con intensidad moderada, con las puntuaciones totales en las tareas Bebras B-1 ($r = +0,511$), y tareas Bebras B-2 ($r = +0,523$). No se encuentra correlación estadísticamente significativa entre el TPC y el total de las tareas Bebras B-3 ($r = +0,175$).

Haciendo un análisis más pormenorizado por tareas individuales, y centrándonos en las tareas Bebras B-2 (que es donde acumulamos mayor muestra, que además se corresponde con el foco central de nuestra validación, es decir, 1° Ciclo de la ESO); vemos que el TPC correlaciona de manera estadísticamente muy significativa ($p_{(r)} < 0,01$), en sentido positivo y con intensidad moderada, con la tarea B2-1 ($r = +0,427$), denominada ‘*Water Supply*’, y con la tarea B2-3 ($r = +0,483$), denominada ‘*Abacus*’. No así con la tarea B2-2 ($r = +0,055$) denominada ‘*Fast Laundry*’. En el [apartado 6.6](#) de discusión y conclusiones parciales se aporta una posible interpretación a estos resultados.

En la siguiente Figura 6.54 se aportan los diagramas de dispersión correspondientes.

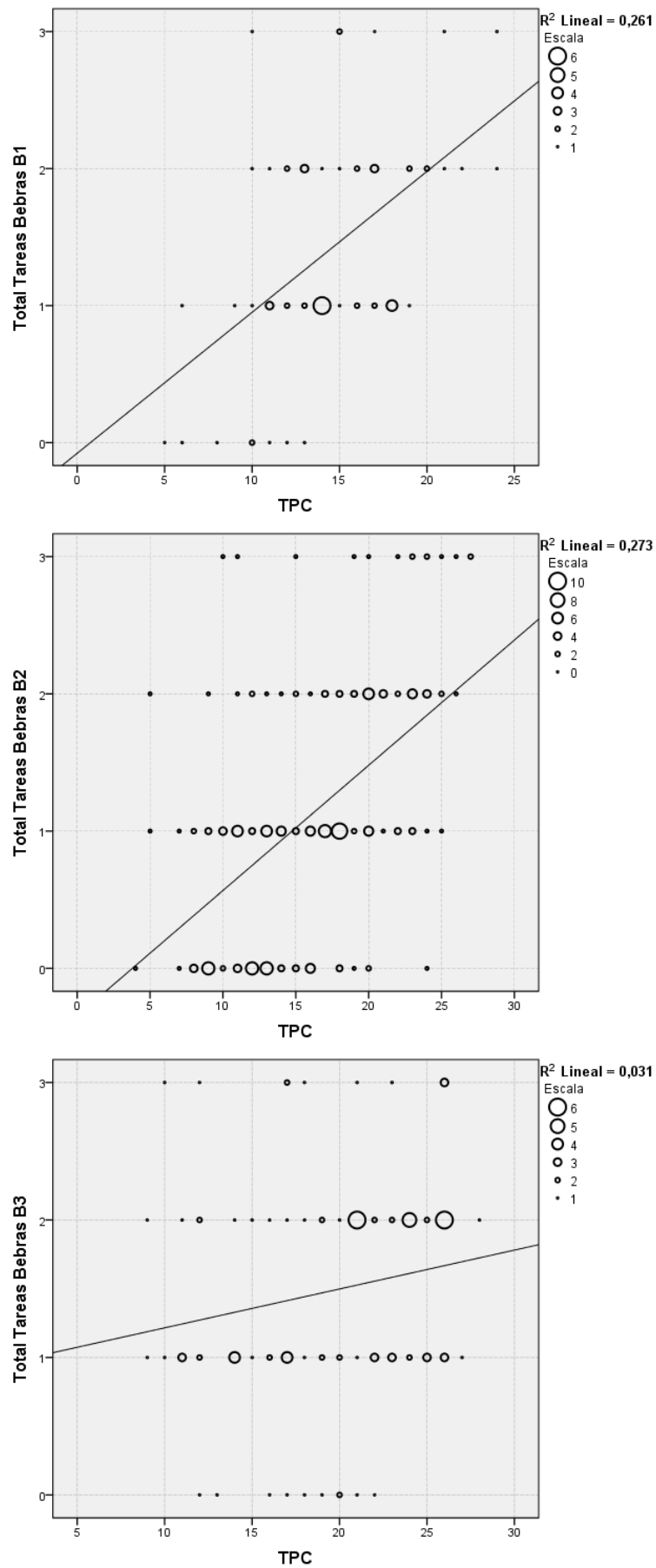


Figura 6.54. Diagramas de dispersión TPC * Tareas Bebras (B-1 arriba, B-2 en el centro, y B-3 abajo)

6.5.2.3.3.2. Con Dr. Scratch

Abordando el objetivo específico de investigación O_{3s} , es decir, estudiar la validez convergente del TPC con respecto a la herramienta Dr. Scratch; recordemos que la sub-muestra para este estudio se compone de sujetos que provienen de 3 centros educativos: IES Vicente Aleixandre, IES Mestre Ramón Esteve, e IES Sevilla Este. En estos centros, el TPC fue aplicado inicialmente en situación *pre*; posteriormente los sujetos iniciaron un periodo de aprendizaje con Scratch (alrededor de 8 semanas); y finalmente los sujetos realizaron una segunda aplicación del TPC en situación *post*, además de analizar sus proyectos finales de Scratch con la herramienta analítica Dr. Scratch. Por tanto, a efectos de la validación convergente del TPC con Dr. Scratch, calcularemos tanto un coeficiente de validez convergente predictivo (TPC_{pre} * Dr. Scratch) como un coeficiente de validez convergente concurrente (TPC_{post} * Dr. Scratch). La muestra finalmente válida está compuesta por los 71 sujetos, todos ellos de 2º ESO, que completaron las 3 medidas (TPC_{pre}, TPC_{post} y Dr. Scratch).

Es importante reseñar que, durante el periodo de aprendizaje con Scratch, en el IES Vicente Aleixandre (30 sujetos) trabajaron con metodología ‘*pair programming*’ (2 estudiantes por cada ordenador); mientras que en el IES Ramón Esteve (13 sujetos) y en el IES Sevilla Este (28 sujetos) lo hicieron con metodología ‘*single programming*’ (1 estudiante por cada ordenador). Una vez finalizado el periodo de aprendizaje de 8 semanas con Scratch, el profesor seleccionó y envió el mejor proyecto de cada estudiante (si trabajaron con metodología ‘*single programming*’) o de cada pareja de estudiantes (si trabajaron con metodología ‘*pair programming*’), para ser analizados con Dr. Scratch.

En la Tabla 6.59 se muestran las estadísticas de muestras emparejadas entre el TPC *pre* y *post*. Tal y como puede observarse, la puntuación media en el TPC en condición *post* ($\bar{X} = 18,58$) es mayor que en la condición *pre* ($\bar{X} = 17,07$). Esta diferencia de medias emparejadas es estadísticamente muy significativa ($t = -3,334$; $p_{(t)} = 0,001 < 0,01$), con un tamaño del efecto asociado $d = 0,37$, que puede considerarse un efecto moderado (Ellis, 2010). Además, la correlación TPC_{pre}*TPC_{post} es estadísticamente muy significativa, de sentido positivo e intensidad moderada ($r = +0,580$).

Tabla 6.59. Estadísticas de muestras emparejadas TPC *pre* y *post*

	Media	N	Desviación estándar	t	p _(t)	d de cohen	r	p _(r)	
Par 1	TPC (Pretest)	17,07	71	3,571	-3,334**	,001	0,37	0,580**	,000
	TPC (Postest)	18,58	71	4,531					

** Significativo al nivel $p < 0,01$

En la Tabla 6.60 se muestra la matriz de correlaciones entre las 3 medidas. Toda ellas son estadísticamente muy significativas ($p_{(r)} < 0,01$), positivas y de intensidad moderada. Y en las Figuras 6.55 y 6.56 se ilustran los diagramas de dispersión del TPC con Dr. Scratch.

Tabla 6.60. Matriz de correlaciones TPC_{pre}, TPC_{post}, y Dr. Scratch

	TPC (Postest)	Total Dr. Scratch
TPC (Pretest)	,580**	,439**
TPC (Postest)		,535**

N=71

** Correlación significativa al nivel $p_{(r)} < 0,01$

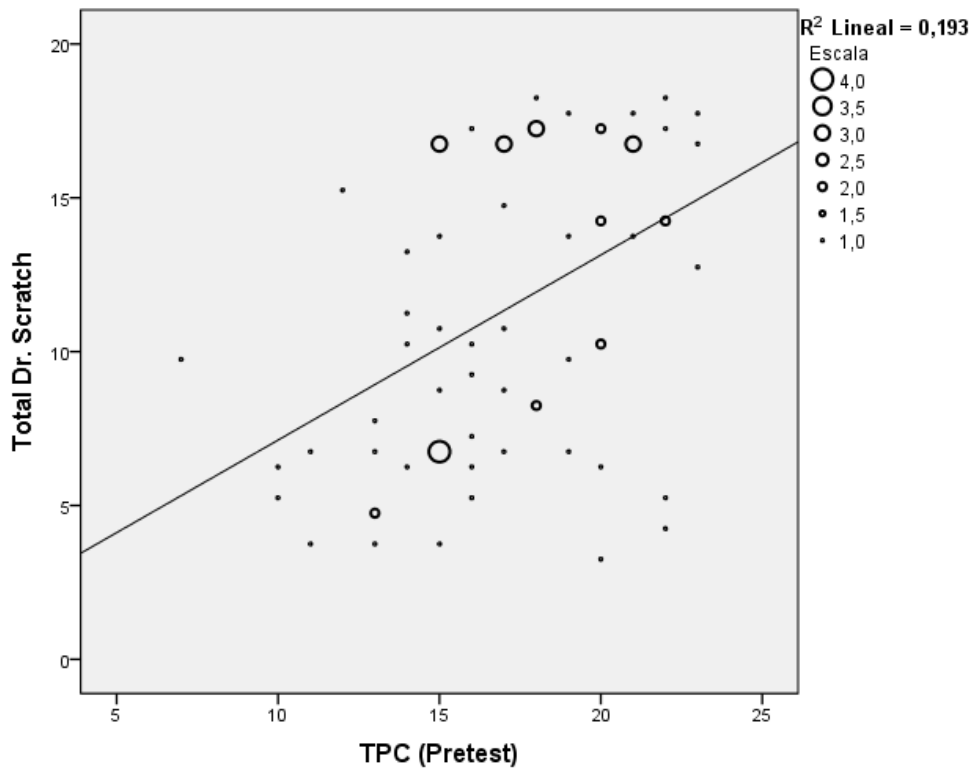


Figura 6.55. Diagrama de dispersión $TPC_{pre} * Dr. Scratch$

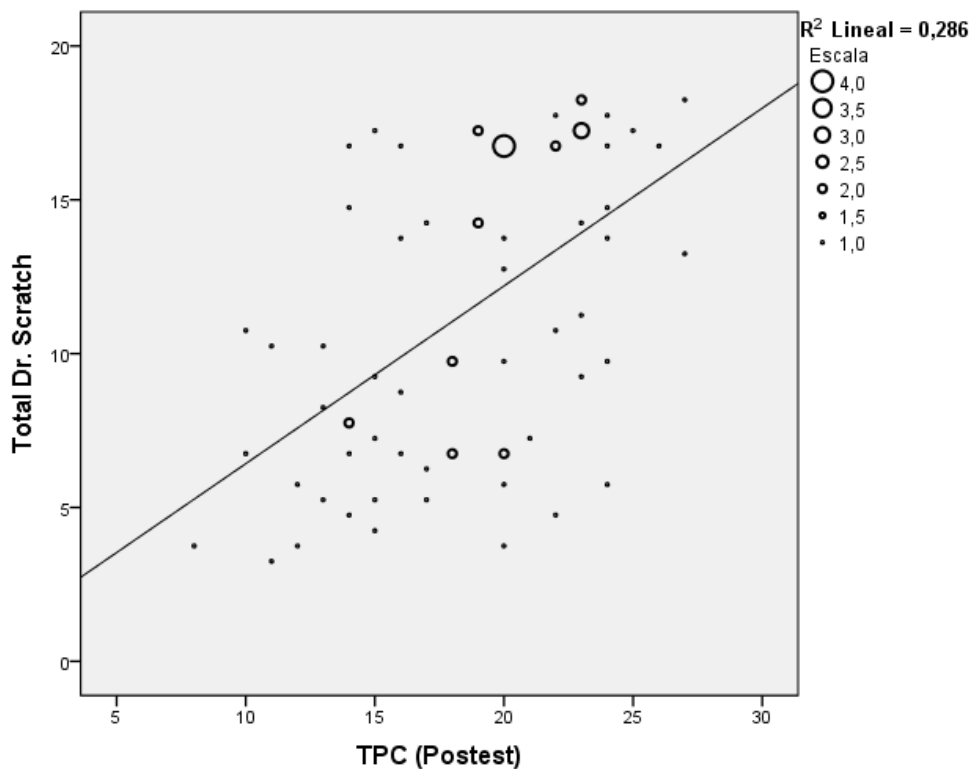


Figura 6.56. Diagrama de dispersión $TPC_{post} * Dr. Scratch$

Tal y como era de esperar, la correlación convergente concurrente $TPC_{post} * Dr. Scratch$, es mayor que la correlación convergente predictiva $TPC_{pre} * Dr. Scratch$; dado que $r = +0,535 > r = +0,439$.

Por otro lado, si segmentamos la muestra en función de la metodología de programación (*'single programming'* vs. *'pair programming'*), sólo encontramos una correlación estadísticamente significativa en la condición *'single programming'* entre el TPC_{post} * Dr. Scratch ($r = +0,367$; $p_{(r)} = 0,018 < 0,05$). Se detalla en la siguiente Tabla 6.61.

Tabla 6.61. Correlaciones TPC * Dr. Scratch, en función de la metodología de programación

		TPC (<i>Pretest</i>)	TPC (<i>Postest</i>)	
Total Dr. Scratch	<i>Single Programming</i>	Correlación de Pearson	,270	
		N	41	
	<i>Pair Programming</i>	Correlación de Pearson	,140	,263
		N	30	30

* Correlación significativa al nivel $p_{(r)} < 0,05$

Finalmente, en la Tabla 6.62 se muestran las correlaciones del TPC, tanto en condición *pre* como *post*, con cada una de las dimensiones del pensamiento computacional medidas a través de Dr. Scratch, a saber: 'Abstracción'; 'Paralelismo'; 'Pensamiento Lógico'; 'Sincronización'; 'Control de Flujo'; 'Interactividad con el Usuario'; y 'Representación de la Información'.

Tabla 6.62. Correlaciones TPC * Dimensiones del Pensamiento Computacional medidas por Dr. Scratch

	Abstracción	Paralelismo	Pensamiento Lógico	Sincronización	Control de Flujo	Interactividad con el usuario	Representación de la Información
TPC (<i>Pretest</i>)	,147	,377**	,418**	,409**	,415**	,296*	,359**
TPC (<i>Postest</i>)	,378**	,452**	,474**	,438**	,449**	,376**	,454**

* Correlación significativa al nivel $p_{(r)} < 0,05$

** Correlación significativa al nivel $p_{(r)} < 0,01$

Al calcular la correlación del TPC_{pre} con cada una de las dimensiones del pensamiento computacional medidas por Dr. Scratch: los valores oscilan entre una correlación positiva, de intensidad baja y estadísticamente no significativa con la dimensión 'Abstracción' ($r = +0,147$; $p_{(r)} = 0,221 > 0,05$) y una correlación estadísticamente muy significativa, de signo positivo e intensidad moderada con la dimensión 'Pensamiento Lógico' ($r = +0,418$; $p_{(r)} = 0,000 < 0,01$). Análogamente, al calcular la correlación del TPC_{post} con cada una de las dimensiones del pensamiento computacional medidas por Dr. Scratch: todas las correlaciones son positivas y estadísticamente muy significativas ($p_{(r)} < 0,01$), oscilando desde $r = +0,376$ en 'Interactividad con el Usuario' y $r = +0,474$ en 'Pensamiento Lógico'. Para todas las dimensiones medidas por Dr. Scratch, las correlaciones en situación *post* son mayores que en situación *pre*. Estos resultados se discuten en el apartado 6.6.

6.5.2.3.4. Validez factorial

Con respecto al objetivo específico de investigación **O_{3i}**, es decir, estudiar la validez factorial del TPC; en primer lugar, comprobamos si la matriz de puntuaciones de los 1.251 sujetos de la muestra a lo largo de los 28 ítems del TPC (puntuaciones recodificadas como $0 = error$, $1 = acierto$) cumple con los requisitos estadísticos para proceder a la factorización. Así es, dado que se encuentra una

medida de adecuación muestral $KMO = 0,874 \approx 1$; y en la prueba de esfericidad de Bartlett se halla un $\chi^2 = 3796,915$ ($p_{(\chi^2)} = 0,000 < 0,01$).

Comenzamos con un análisis factorial exploratorio de la matriz de puntuaciones (1.251 sujetos-filas * 28 ítems-columnas) según los siguientes parámetros: método de extracción por ‘factorización de eje principal’; basado en autovalores > 1 ; método de rotación ‘Promax con normalización Kaiser’ con $Kappa = 4$. Tal y como puede observarse en la siguiente Tabla 6.63, se obtiene una solución inicial de 7 factores, que explicaría un 41,87% de la varianza total de la matriz de puntuaciones.

Tabla 6.63. Varianza explicada por la solución inicial de 7 factores

Factor	Autovalores iniciales			Sumas de las saturaciones al cuadrado de la extracción			Suma de las saturaciones al cuadrado de la rotación
	Total	% de la varianza	% acumulado	Total	% de la varianza	% acumulado	Total
1	4,461	15,933	15,933	3,720	13,287	13,287	2,286
2	1,675	5,981	21,914	,921	3,288	16,575	2,633
3	1,259	4,496	26,411	,523	1,869	18,444	1,804
4	1,184	4,227	30,638	,489	1,748	20,192	2,056
5	1,104	3,943	34,581	,311	1,112	21,304	2,700
6	1,029	3,674	38,255	,266	,949	22,253	1,938
7	1,014	3,622	41,878	,230	,822	23,075	,951
8	,988	3,530	45,408				
9	,971	3,468	48,876				
10	,926	3,308	52,184				
11	,907	3,238	55,422				
12	,884	3,155	58,578				
13	,871	3,110	61,687				
14	,849	3,031	64,719				
15	,826	2,950	67,669				
16	,816	2,915	70,584				
17	,795	2,838	73,422				
18	,778	2,779	76,201				
19	,762	2,723	78,924				
20	,733	2,619	81,543				
21	,728	2,600	84,143				
22	,677	2,417	86,560				
23	,662	2,363	88,924				
24	,653	2,334	91,257				
25	,646	2,307	93,565				
26	,632	2,256	95,821				
27	,608	2,172	97,992				
28	,562	2,008	100,000				

Método de extracción: Factorización de Ejes principales.

Dado el gráfico de sedimentación (Figura 6.57) y la matriz de estructura resultante (Tabla 6.64), se decide explorar con más profundidad el contenido y propiedades de los 3 primeros factores, que explican un 26,41% de la varianza total (Tabla 6.63)

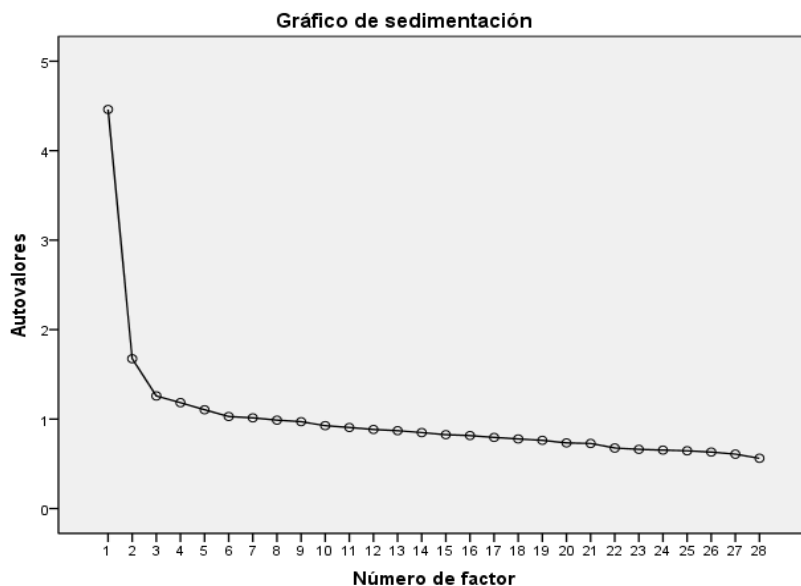


Figura 6.57. Gráfico de sedimentación del análisis factorial exploratorio inicial

Tabla 6.64. Matriz de estructura resultante del análisis factorial exploratorio inicial (sólo se muestran cargas > 0,30)

	Factor						
	1	2	3	4	5	6	7
Pregunta 2	,544						
Pregunta 6	,472						
Pregunta 9	,472						
Pregunta 3	,460				,360		
Pregunta 11	,456	,312			,357	,395	
Pregunta 5	,447						
Pregunta 13	,368						
Pregunta 1							
Pregunta 28		,581			,330		
Pregunta 27		,476				,303	
Pregunta 26		,462					
Pregunta 20		,445	,365		,338	,384	
Pregunta 24		,430			,317		
Pregunta 10		,301					
Pregunta 22		,329	,488		,315	,470	
Pregunta 25			,405				
Pregunta 23			,319				
Pregunta 18		,307		,680	,338		
Pregunta 17				,378	,330	,362	-,302
Pregunta 19				,345			
Pregunta 7	,355	,334		,385	,582		
Pregunta 14		,334	,422	,412	,525	,318	-,348
Pregunta 4		,336	,418	,352	,422		
Pregunta 16			,356		,374	,346	
Pregunta 8					,367		
Pregunta 21			,325			,356	
Pregunta 12							
Pregunta 15							

Método de extracción: Factorización del eje principal.

Método de rotación: Normalización Promax con Kaiser.

Tal y como puede observarse en la anterior Tabla 6.64, los 3 factores principales estarían internamente compuestos por los siguientes ítems (ítems con saturaciones $\geq 0,4$ en el factor; a excepción del ítem 16 que, con saturación sólo de 0,356 en el factor 3, se incluye para que éste llegue a contener un mínimo de 5 ítems):

- **Factor 1 (en verde):** compuesto por los ítems 2, 3, 5, 6, 9 y 11. Si agrupamos las especificaciones de estos seis ítems, comprobamos que su contenido fundamental está referido a los conceptos computacionales de ‘direcciones-secuencias básicas’ y ‘bucles’, tanto ‘repetir veces’ como ‘repetir hasta’ (Tabla 6.65). Si calculamos una subescala derivada, que denominaremos ‘*Secuencias y Bucles*’, a partir de dichos seis ítems; obtenemos una fiabilidad para la misma de $\alpha = 0,621$ (Tabla 6.66).

Tabla 6.65. Especificaciones de los ítems que componen el Factor 1

Ítem	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							
			Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)	
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros
Ítem 2	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No
Ítem 3	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No
Ítem 5	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No
Ítem 6	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No
Ítem 9	Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No
Ítem 11	Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No

Tabla 6.66. Estadísticos y fiabilidad de la subescala ‘*Secuencias y Bucles*’

Media	Varianza	Desviación típica	Alfa de Cronbach	N de elementos
5,15	1,461	1,209	,621	6

- **Factor 2 (en azul):** compuesto por los ítems 20, 24, 26, 27 y 28. Si agrupamos las especificaciones de estos cinco ítems, comprobamos que su contenido fundamental (aunque no sólo) está referido al concepto computacional de ‘funciones simples’ (Tabla 6.67). Si calculamos una subescala derivada, que denominaremos ‘*Funciones*’, a partir de dichos cinco ítems; obtenemos una fiabilidad para la misma de $\alpha = 0,589$ (Tabla 6.68).

Tabla 6.67. Especificaciones de los ítems que componen el Factor 2

Ítem	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							
			Direcc.	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)	
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros
Ítem20	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No
Ítem24	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No
Ítem26	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No
Ítem27	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No
Ítem28	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No

Tabla 6.68. Estadísticos y fiabilidad de la subescala 'Funciones'

Media	Varianza	Desviación típica	Alfa de Cronbach	N de elementos
3,01	2,244	1,498	,589	5

- **Factor 3 (en rojo):** compuesto por los ítems 4, 14, 16, 22 y 25. Si agrupamos las especificaciones de estos cinco ítems, comprobamos que su contenido fundamental (aunque no sólo) está referido a los conceptos computacionales de 'condicional simple' y 'condicional mientras que' (Tabla 6.69). Si calculamos una subescala derivada, que denominaremos '**Condicionales**', a partir de dichos cinco ítems; obtenemos una fiabilidad para la misma de $\alpha = 0,575$ (Tabla 6.70).

Tabla 6.69. Especificaciones de los ítems que componen el Factor 3

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Direcciones	Concepto computacional abordado							
			Bucles (loops)		Condicionales (conditionals)			Funciones (functions)		
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros	
Ítem4	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No
Ítem14	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No
Ítem16	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No
Ítem22	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No
Ítem25	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No

Tabla 6.70. Estadísticos y fiabilidad de la subescala 'Condicionales'

Media	Varianza	Desviación típica	Alfa de Cronbach	N de elementos
2,03	2,180	1,476	,575	5

Dado que el anterior análisis factorial no resulta plenamente concluyente, ni en la estructura ni en las propiedades métricas de los 3 factores estudiados, se aborda un análisis unifactorial complementario según los siguientes parámetros: método de extracción por 'factorización de eje principal'; número fijo de factores = 1 (por tanto, no hay rotación). Como resultado del mismo, el único factor extraído explica un 15,93% de la varianza total de las puntuaciones. En la siguiente Tabla 6.71 se detalla la matriz factorial resultante.

 Tabla 6.71. Matriz factorial^a resultante del análisis unifactorial (sólo se muestran cargas > 0,30)

	Factor
	1
Pregunta 14	,485
Pregunta 7	,478
Pregunta 20	,456
Pregunta 4	,451
Pregunta 22	,424
Pregunta 11	,420
Pregunta 24	,412
Pregunta 3	,401
Pregunta 27	,400

Pregunta 18	,386
Pregunta 17	,385
Pregunta 13	,381
Pregunta 16	,380
Pregunta 28	,378
Pregunta 2	,368
Pregunta 6	,361
Pregunta 26	,358
Pregunta 21	,351
Pregunta 10	,339
Pregunta 8	,318
Pregunta 5	,316
Pregunta 9	,310
Pregunta 25	
Pregunta 19	
Pregunta 1	
Pregunta 12	
Pregunta 23	
Pregunta 15	

Método de extracción: Factorización del eje principal.

a. 1 factores extraídos. Requeridas 3 iteraciones.

Tal y como puede observarse, la mayoría de los ítems presentan saturaciones medias ($\approx 0,4$) en el único factor extraído. Jugando con el punto de corte, encontramos una posible subescala de 13 elementos formada por los ítems con saturaciones $\geq 0,38$ en el unifactor (en color gris en la tabla anterior). Esta subescala de 13 elementos, que aglutina los ítems con mayor carga en el unifactor y a la que denominaremos '*Común-Principal*' (Tabla 6.72), ya presenta niveles de fiabilidad aceptables ($\alpha = 0,741$).

Tabla 6.72. Estadísticos y fiabilidad de la subescala '*Común-Principal*'

Media	Varianza	Desviación típica	Alfa de Cronbach	N de elementos
7,45	9,304	3,050	,741	13

Como síntesis, al combinar los resultados de ambos análisis factoriales, se nos revela una versión depurada de nuestro test, de tan sólo 20 ítems de longitud, y a la que llamaremos **TPC-R**. Esta versión reducida está compuesta por todos los ítems que aparecen, bien en cualquiera de los 3 factores iniciales rotados, bien en el unifactor general (o en ambos). Es decir, el TPC-R incluye todos los ítems que componen las anteriores subescalas; concretamente los ítems: 2, 3, 4, 5, 6, 7, 9, 11, 13, 14, 16, 17, 18, 20, 22, 24, 25, 26, 27 y 28. En la Tabla 6.73 se presentan los estadísticos y fiabilidad de esta versión TPC-R.

Tabla 6.73. Estadísticas y fiabilidad de la versión reducida TPC-R

Media	Varianza	Desviación típica	Alfa de Cronbach	N de elementos
12,64	15,529	3,941	,782	20

Así pues, aplicando una versión reducida de nuestro test de tan sólo 20 ítems (lo que supone una reducción en la longitud del test del 28,57%), podríamos obtener una puntuación total TPC-R ya con buena fiabilidad y homóloga a la puntuación en el TPC original; y además, obtener 4 puntuaciones derivadas de las respectivas 4 subescalas (una '*Común-Principal*', y tres específicas: '*Secuencias y Bucles*', '*Funciones*' y '*Condicionales*'). En cualquier caso, ello exigiría confirmación empírica a través de una nueva ronda de aplicaciones con el instrumento TPC-R, algo que excede esta tesis doctoral y que queda pendiente para trabajos futuros. En la Tabla 6.74 se resumen las características de las distintas escalas y subescalas definidas.

Tabla 6.74. Resumen de escalas y subescalas definidas

		Nº de Ítems	Fiabilidad (α)	Media	Desviación típica	Ítems que forman la escala o subescala
Escalas	TPC	28	0,793	16,38	4,824	Todos
	TPC-R	20	0,782	12,64	3,941	2, 3, 4, 5, 6, 7, 9, 11, 13, 14, 16, 17, 18, 20, 22, 24, 25, 26, 27 y 28
Subescalas	Común-Principal	13	0,741	7,45	3,050	3, 4, 7, 11, 13, 14, 16, 17, 18, 20, 22, 24 y 27
	Secuencias y Bucles	6	0,621	5,15	1,209	2, 3, 5, 6, 9 y 11
	Funciones	5	0,589	3,01	1,498	20, 24, 26, 27 y 28
	Condicionales	5	0,575	2,03	1,476	4, 14, 16, 22 y 25

En la Tabla 6.75 se muestra la matriz de correlaciones entre las puntuaciones totales de los sujetos para cada una de las escalas y subescalas definidas (la puntuación es siempre calculada como suma de aciertos en los ítems que componen cada una de ellas). Todas las correlaciones son positivas y estadísticamente muy significativas ($p_{(r)} < 0,01$). Y la configuración de las intensidades es coherente: una intensidad muy alta entre TPC y TPC-R ($r = +0,962$; en tono oscuro en la tabla y diagrama de dispersión en la Figura 6.58); también intensidades muy altas entre la subescala '*Común-Principal*' con respecto a TPC y TPC-R ($r > 0,90$; en tono oscuro en la tabla); una intensidad moderada-alta tanto de TPC, TPC-R y '*Común-Principal*' con respecto del resto de subescalas ($r \approx +0,7$; en tono medio en la tabla); y una intensidad moderada-baja de las subescalas específicas entre sí ($r \approx +0,3$; en tono claro).

Tabla 6.75. Matriz de correlaciones entre las escalas y subescalas definidas

	TPC	TPC-R	Común-Principal	Secuencias y bucles	Funciones	Condicionales
TPC	1	,962**	,925**	,628**	,723**	,745**
TPC-R		1	,958**	,668**	,757**	,758**
Común-Principal			1	,566**	,685**	,765**
Secuencias y bucles				1	,339**	,296**
Funciones					1	,417**
Condicionales						1

** Correlación significativa al nivel $p_{(r)} < 0,01$

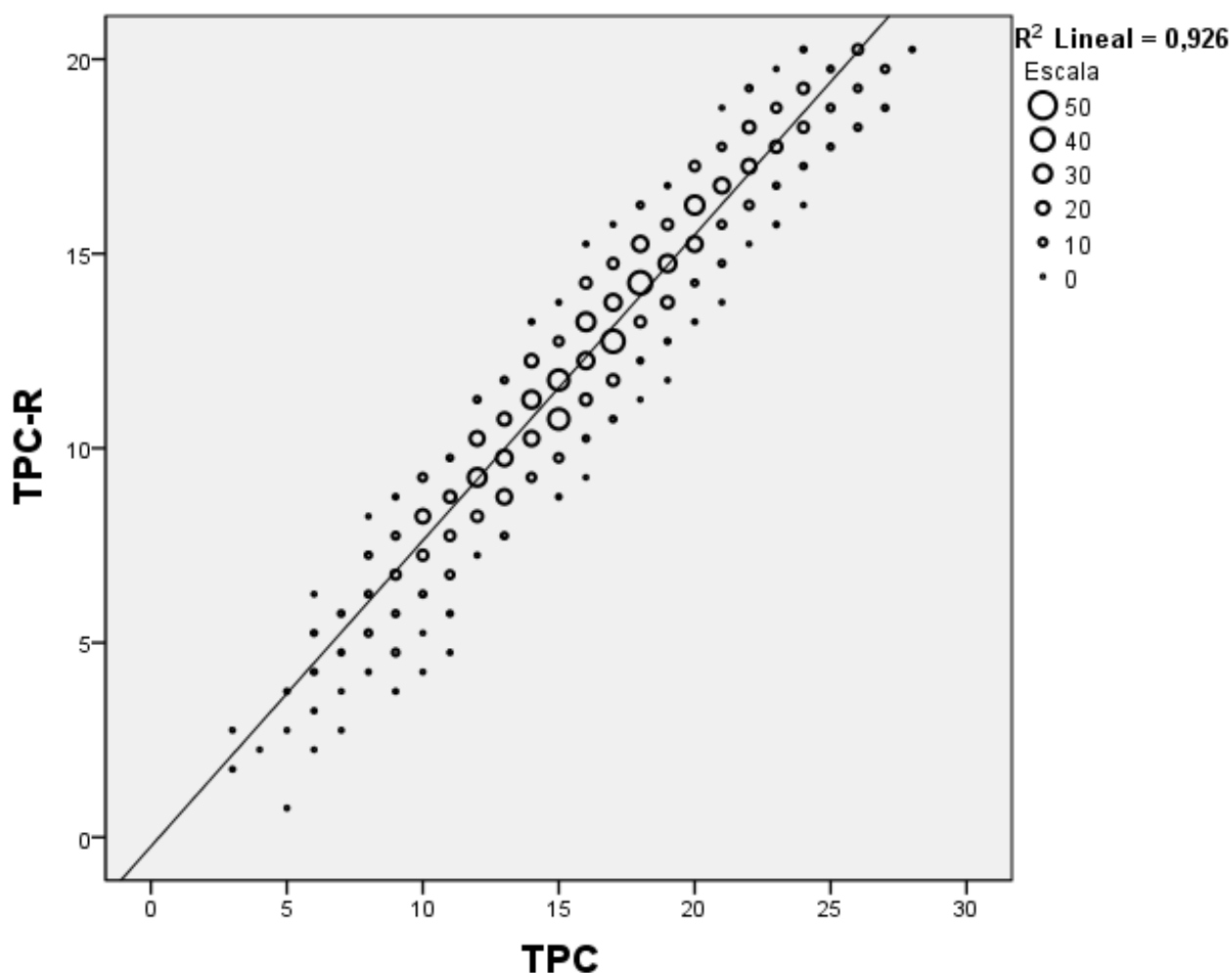


Figura 6.58. Diagrama de dispersión TPC * TPC-R

Finalmente, en la siguiente Tabla 6.76 se muestran las correlaciones de cada una de las escalas y subescalas definidas, con las medidas de validez criterial concurrente utilizadas en nuestro estudio (en negrita se marcan las correlaciones que serán comentadas a continuación)

Tabla 6.76. Matriz de correlaciones de las escalas y subescalas, con las medidas de validez criterial concurrente

	PMA (R)	PMA (E)	PMA (N)	PMA (V)	RP30	FI-R (AE)	FI-R (ICI)	BFQ (Co)	BFQ (Ap)	BFQ (Ex)	BFQ (Am)	BFQ (Ie)
TPC	,442**	,439**	-,157	,273**	,669**	,229	,204	,267**	,407**	,304**	,133	,092
TPC-R	,444**	,423**	-,160*	,265**	,668**	,314*	,193	,237*	,421**	,265**	,135	,112
Común-Princ.	,417**	,409**	-,195*	,237**	,691**	,350**	,156	,267**	,473**	,248*	,140	,114
Sec. y Bucles	,296**	,272**	-,119	,148	,400**	,141	,314*	,064	,146	,219*	,091	,142
Funciones	,316**	,244**	-,019	,168*	,526**	,070	,157	,269**	,396**	,248*	,189	,009
Condicionales	,263**	,334**	-,062	,246**	,464**	,376**	,206	,150	,287**	,175	,110	,063

* Correlación significativa al nivel $p_{(r)} < 0,05$

** Correlación significativa al nivel $p_{(r)} < 0,01$

De la observación detallada de la anterior Tabla 6.76, derivamos algunos resultados de especial interés (en negrita en la tabla):

- Tanto la versión reducida TPC-R como la subescala '*Común-Principal*', presentan correlaciones con respecto a las medidas de validez criterial concurrente muy similares a las obtenidas con el TPC. Ello aporta evidencia de la bondad psicométrica de ambas. Sin embargo, encontramos algunas pequeñas diferencias de interés:
 - Tanto el TPC-R como la subescala '*Común-Principal*' llegan a correlacionar negativamente de manera significativa ($p_{(r)} < 0,05$) con el PMA-N (Factor Numérico). Esta significación no se encontró en el caso del TPC ($r = -0,157$; $p_{(r)} = 0,052 > 0,05$)
 - El TPC-R llega a correlacionar de manera significativa con el 'Índice de Percepción y Atención' (A-E) del Test FI-R ($r = +0,314$; $p_{(r)} = 0,014 < 0,05$); dicha correlación con el A-E se vuelve incluso muy significativa en el caso de la subescala '*Común-Principal*' ($r = +0,350$; $p_{(r)} = 0,006 < 0,01$). Sin embargo, esta significación no se encontró en el caso del TPC ($r = +0,229$; $p_{(r)} = 0,076 > 0,05$).
 - Adicionalmente, mientras que las correlaciones del TPC con respecto a las dimensiones 'Apertura' (Ap) y 'Extraversión' (Ex) del BFQ-NA eran ambas muy significativas y de intensidad muy similar (0,40 y 0,30 respectivamente); esas mismas correlaciones difieren más entre sí cuando se calculan respecto de la subescala '*Común-Principal*' (0,47 y 0,24 respectivamente)
- La subescala '*Secuencias y Bucles*' correlaciona positiva y significativamente ($r = +0,314$; $p_{(r)} = 0,014 < 0,05$) con el 'Índice de Control de la Impulsividad' (ICI) del Test FI-R; correlación que no llega a ser significativa ni en el caso del TPC ($r = +0,204$; $p_{(r)} = 0,115 > 0,05$), ni en el caso del TPC-R ($r = +0,193$; $p_{(r)} = 0,135 > 0,05$). Por otro lado, la subescala '*Secuencias y Bucles*' es la única que no arroja correlación significativa con la dimensión 'Apertura' del BFQ-NA ($r = +0,146$; $p_{(r)} = 0,149 > 0,05$).
- La subescala '*Funciones*', en comparación con las otras subescalas específicas, presenta correlaciones muy significativas, y de especial intensidad, con las dimensiones 'Conciencia' ($r = +0,269$; $p_{(r)} = 0,007 < 0,01$) y 'Apertura' ($r = +0,396$; $p_{(r)} = 0,000 < 0,01$) del BFQ-NA.
- La subescala '*Condicionales*', en comparación con las otras subescalas específicas, presenta correlaciones muy significativas, y de especial intensidad, con el 'Factor Verbal' del PMA ($r = +0,246$; $p_{(r)} = 0,002 < 0,01$), y con el 'Índice de Percepción y Atención' (A-E) del Test FI-R ($r = +0,376$; $p_{(r)} = 0,003 < 0,01$)
- Finalmente, hay que destacar que las medidas criterioles PMA-R ('Factor Razonamiento'), PMA-E ('Factor Espacial'), y RP30 ('Resolución de problemas'), presentan correlaciones muy significativas con todas las escalas y subescalas definidas.
- A la inversa, las dimensiones 'Amabilidad' (Am) e Inestabilidad Emocional (Ie) del BFQ-NA no llegan a correlacionar de manera significativa con ninguna de las escalas y subescalas definidas. Lo discutimos al final del siguiente apartado.

6.6. Discusión y conclusiones parciales

Con respecto al objetivo específico **O_{3a}** podemos concluir que:

- Se ha conseguido diseñar una primera versión del instrumento TPC (*versión 1.0*) con arreglo a las directrices propuestas por Buffum *et al.* (2015) sobre la construcción de instrumentos de medida en el área de las Ciencias de la Computación para edades ‘middle school’: se ha identificado el propósito del test; se ha definido operacionalmente el constructo a medir; se han explicitado las especificaciones del test (test de elección múltiple con 4 opciones de respuesta); y se han generado 40 ítems candidatos caracterizados en los 5 ejes de diseño (‘concepto computacional abordado’, ‘entorno-interfaz del ítem’, ‘estilo de las alternativas de respuesta’, ‘existencia o inexistencia de anidamiento’, y ‘tarea requerida’), cuya elección se ha fundamentado en la parte teórica. Con ello, también alineamos el TPC con los estándares fijados por la ‘American Educational Research Association’ (AERA), la ‘American Psychological Association’ (APA), y el ‘National Council on Measurement in Education’ (NCME) para la construcción de tests en el ámbito psicoeducativo (AERA, APA, & NCME, 2014)
- Adicionalmente, este diseño inicial del TPC (*versión 1.0*) es coherente con los otros dos instrumentos psicométricos de medida del pensamiento computacional (PC) en edades ‘middle school’ o ‘high school’ que revisamos en la parte teórica (epígrafe 3.3.2), a saber respectivamente: el ‘Test for Measuring Basic Programming Abilities’ (Mühling, Ruf, & Hubwieser, 2015), y el ‘Commutative Assessment’ (Weintrop & Wilensky, 2015a)

Con respecto al objetivo específico **O_{3b}** podemos concluir que:

- Se ha sometido al TPC (*versión 1.0*) a un proceso de validación de contenido a través del juicio de 20 expertos. El grupo de expertos se ha constituido mayoritariamente con profesores del área de Informática de etapa universitaria y pre-universitaria; que han valorado el TPC a través de una mezcla de preguntas dicotómicas, escalas tipo Likert y comentarios abiertos. Todo ello cumple con las directrices que Buffum *et al.* (2015) sugieren para la revisión de los ítems candidatos a la versión final del instrumento.
- La fiabilidad inter-jueces (‘inter-rater reliability’), mostrada por los expertos, fue excelente en lo relativo a su valoración de la ‘dificultad’ de los 40 ítems candidatos ($\alpha = 0,968$), y moderada en lo relativo a la ‘aceptación’ de los ítems para formar parte de la versión final del TPC ($\alpha = 0,467$); lo que otorga bondad métrica al proceso de validación de contenido. Sin embargo, se encontró una fiabilidad inter-jueces muy discreta en lo relativo a la ‘pertinencia percibida para medir el PC’ de los ítems ($\alpha = 0,294$); lo que parece un claro indicador de que aún no existe entre los expertos un consenso claro sobre cuál es el corazón central del constructo.
- Como resultado del proceso de validación de contenido, se llega a la versión final del TPC (*versión 2.0*), de 28 ítems de longitud para ser completados en un tiempo máximo de 45 minutos. Con estas especificaciones, cumplimos las directrices de Buffum *et al.* (2015), que

sugieren instrumentos de, al menos, 20 ítems de longitud, y que puedan ser completados por los estudiantes en una sesión ordinaria de clase.

- Las dos decisiones fundamentales tomadas para la depuración de la versión original del TPC a su versión final han sido: la exclusión de los ítems relativos al concepto computacional ‘funciones con parámetros’, y la eliminación del estilo ‘textual’ en las alternativas de respuesta; ambas debidas a su excesiva dificultad para la población objetivo (1º-2º ESO). Estas decisiones son consistentes con el instrumento ‘*Commutative Assessment*’ (Weintrop & Wilensky, 2015a), que apuesta por introducir la medida de parámetros y variables, alternando lenguajes visuales y textuales, a partir de la entrada del estudiante en la ‘*high school*’, esto es, a partir del Grado 9 (equivalentes a nuestro 3º ESO).

Con respecto a los objetivos específicos **O_{3c}** y **O_{3d}**, podemos concluir que:

- Se ha conseguido virtualizar-digitalizar el TPC a través de la tecnología *Google Drive Forms*, de manera que sea accesible desde cualquier dispositivo electrónico fijo o portátil (incluidas tabletas) con conexión a Internet.
- Se ha podido garantizar una adecuada legibilidad y navegabilidad de dicha virtualización del TPC. Así, los sujetos de la población objetivo son capaces de: acceder al TPC a partir del enlace URL que se les proporciona; leer y comprender autónomamente las instrucciones y ejemplos; navegar a lo largo del TPC, emitiendo las correspondientes respuestas y enviándolas a su conclusión. Se garantiza igualmente la recepción de las respuestas en la base de datos asociada al test.

Con respecto al objetivo específico **O_{3e}**, podemos concluir que:

- Se ha conseguido aplicar el TPC sobre una muestra representativa y de tamaño suficiente de la población objetivo. Concretamente, se ha acumulado una muestra de tamaño total $N=1.251$ sujetos, de los cuales 735 (58,8%) corresponden al foco central del instrumento (1º y 2º de la ESO). Los sujetos provienen de 24 centros educativos distintos (tanto públicos como privados), y de 7 provincias diferentes de la geografía española, asegurando la representatividad de la muestra a nivel nacional.
- Adicionalmente, se observa un desequilibrio de género en la muestra ($\approx 60\%$ vs. 40% a favor de los chicos). Dado que el TPC se aplicó mayoritariamente en aulas de la ESO y, dentro de las mismas, durante la clase optativa de Informática; podemos tomar la proporción anterior como indicativa de la elección diferencial por sexo que actualmente existe en lo que a la asignatura optativa de Informática se refiere.

Con respecto al objetivo específico **O_{3f}**, podemos concluir que:

- Las puntuaciones totales en el TPC se distribuyen con un ajuste notable a la curva normal, algo esperable en una prueba supuestamente aptitudinal; y de manera simétrica alrededor de la media ($\bar{X} = 16,38$), que se sitúa muy cerca del punto medio de la escala (14 puntos sobre 28 posibles).

- Las puntuaciones totales en el TPC presentan una buena variabilidad (rango que oscila entre 3 y 28 puntos, sobre un rango posible de 0-28 puntos), lo que permite la construcción de baremos adecuados para la población de referencia.
- El rendimiento en el TPC se incrementa de manera estadísticamente significativa según avanzamos en ciclo educativo y curso académico. Este hecho es coherente con nuestra definición del ‘pensamiento computacional’ como aptitud cognitiva de solución de problemas que, por tanto, estaría vinculada al desarrollo madurativo de los sujetos.
- El rendimiento en el TPC de los chicos es superior al de las chicas ($d = 0,31$). Este resultado es consistente con la correlación positiva encontrada entre el TPC y los factores ‘Espacial’ y ‘Razonamiento’ del PMA; sendas aptitudes en las cuales la psicología y pedagogía diferencial ha establecido tradicionalmente una superioridad masculina (Caselles Pérez, 1997; Linn & Petersen, 1985; Voyer, Voyer, & Bryden, 1995). El rendimiento diferencial en el TPC a favor del colectivo masculino se incrementa a medida que avanzamos en ciclo educativo, y sólo es estadísticamente significativo de 1º Ciclo de la ESO en adelante. Parece, pues, que existe una brecha de género progresiva en lo que a nivel de desarrollo del pensamiento computacional se refiere.
- Siguiendo con la cuestión de género, la percepción de ‘autoeficacia específica’ sobre el rendimiento en el TPC es también superior en el caso de los chicos frente a las chicas. En este caso, el tamaño del efecto asociado asciende hasta $d = 0,42$. Dicho en otras palabras, en la percepción de autoeficacia con el TPC se abre una brecha de género adicional a la ya existente en el desempeño objetivo en el instrumento. Así pues, las chicas muestran significativamente menor confianza que los chicos sobre su respuesta al test. Este débil percepción de autoeficacia en tareas relacionadas con el pensamiento computacional podría afectar negativamente a las chicas al afrontar el aprendizaje de la programación, y podría disuadirlas de elecciones académicas y profesionales vinculadas a la informática; tal y como ya se ha sugerido en estudios previos de género (Soe & Yakura, 2008).
- Se encuentra una correlación significativa, positiva y de intensidad moderada entre la puntuación total en el TPC y la percepción de autoeficacia sobre el rendimiento en el mismo o ‘autoeficacia específica’ ($r = +0,381$); es decir, aunque los sujetos no tenían experiencia previa en pruebas de medida del pensamiento computacional, sí perciben de manera moderada que responden al test siguiendo una cierta lógica. En contraste, hay una correlación baja, aunque estadísticamente significativa ($r = +0,188$) entre la puntuación total en el TPC y la percepción de autoeficacia de los sujetos sobre su nivel general de competencia informática (‘autoeficacia general’), posiblemente porque éstos relacionan dicha competencia con usos-consumos pasivos de la tecnología en vez de con tareas activas y creativas como las que involucran la programación y el pensamiento computacional.
- La percepción de ‘autoeficacia específica’ desciende según avanzamos en ciclo educativo, justo a la inversa de lo que sucede con el rendimiento en el TPC. Este hecho, aparentemente paradójico, se puede explicar dado el mayor sentido autocrítico de los sujetos de mayor edad, que va aparejado con apreciaciones más ajustadas y consistentes; así, la correlación TPC *

‘autoeficacia específica’ se va incrementando con el ciclo educativo, desde $r = +0,361$ en 3º Ciclo de Primaria a $r = +0,527$ en 2º Ciclo de la ESO. Dicho en otras palabras, a medida que los estudiantes son más mayores, son más precisos en la apreciación de cómo ha sido su desempeño en el test.

- El TPC tiene un índice de dificultad promedio, a lo largo de sus 28 ítems, de 0,59 (sin corregir) o 0,45 (corregido). Es decir, podemos afirmar que, globalmente, el TPC presenta un grado de dificultad adecuado-medio (en el rango 0,40 a 0,60) para la población objetivo, tal y como se recomienda para pruebas de aptitud (Navas Ara, 2001)
- El TPC presenta una dificultad progresiva a lo largo de sus 28 ítems (desde valores p_i altos al inicio del test, hacia valores p_i bajos), tal y como se recomienda en pruebas de aptitud en cuya construcción se aconseja comenzar por ítems fáciles para ir complicándolos progresivamente. Detectamos algún ítem excesivamente fácil, con $p_{i\text{-corregido}} \geq 0,95$: en concreto, el ítem 1 se aproxima a ese nivel con $p_{i\text{-corregido}} = 0,94$. Y algunos excesivamente difíciles, con $p_{i\text{-corregido}} \leq 0,05$: en concreto, los ítems 12, 15, y 23.
- La correlación ítem-total promedio del TPC a lo largo de sus 28 ítems es $ITC = 0,39$. Es un indicador de la adecuada homogeneidad interna de nuestro test, dado que se recomiendan valores $> 0,30$ (Buffum *et al.*, 2015) para asegurar que el instrumento está correctamente construido. Encontramos varios ítems con valores $ITC < 0,30$, que deben ser revisados o eliminados de futuras versiones del instrumento: en concreto, los ítems 1, 12, 15 y 23.
- El índice de discriminación promedio del TPC a lo largo de sus 28 ítems es $D = 0,38$. Es un indicador de la adecuada potencia discriminatoria del test, dado que se recomiendan valores $> 0,30$ (Buffum *et al.*, 2015) para asegurar que el instrumento está correctamente construido. Encontramos varios ítems con valores $D < 0,30$, que, por su bajo poder discriminatorio, son candidatos a ser revisados o eliminados en futuras versiones del test: es el caso de los ítems 1, 12, 15 y 23 (de los que ya conocíamos sus debilidades psicométricas), a los que se unen ahora los ítems 2, 5, 6 y 9 (que son ítems muy fáciles, con $p_i \approx 0,90$, lo que incide en su baja capacidad de discriminación).
- Tomando las anteriores conclusiones en conjunto, consideramos que hay tres ítems que merecen nuestra especial atención para ser revisados o eliminados en futuras versiones del instrumento: ítems 12, 15 y 23. ¿Qué puede estar fallando en estos ítems? (Anexo F para verlos en detalle), hacemos un análisis crítico de los mismos a continuación:
 - En el ítem 12 (Anexo F) hay un anidamiento simple de un bucle definido (‘repetir veces’) dentro de un bucle indefinido (‘repetir hasta’); en esta línea, Mühlhng y Hubwieser (2015) han reportado recientemente la dificultad que supone para estudiantes ‘middle school’ resolver ítems en donde se presenta un bucle indefinido (los autores lo llaman ‘loop with exit condition’) en anidamiento. Además, se añade cierta dificultad porque ‘el artista’ aparece orientado de espaldas al sujeto que contesta el ítem; pudiéndose generar cierta duda con respecto al punto de inicio del lápiz. Una posible revisión del ítem consistiría en tumbar la escalera que lleva hasta la flor, situando al artista de perfil.

- En el ítem 15 (Anexo F) hay un doble anidamiento: bucle definido (‘repetir veces’), dentro de condicional simple (‘si’), y éste a su vez dentro de bucle indefinido (‘repetir hasta’). Por lo tanto, a la dificultad señalada en el punto anterior, se le añade la relativa a un segundo anidamiento. Además, las cuatro alternativas de respuesta no son equivalentes: las tres primeras son colores (‘azul’, ‘amarillo’, ‘rojo’), mientras que la cuarta es la afirmación “tanto la opción A como la opción C son correctas”. Por todo ello, el ítem 15 parece un clarísimo candidato para ser eliminado en una futura versión mejorada del TPC.
- En el ítem 23 (Anexo F) hay un anidamiento simple de un bucle definido (‘repetir veces’) seguido por un condicional simple (‘si’), dentro de un bucle indefinido (‘mientras que’). Además de su dificultad inherente, reconocemos un posible fallo en la forma en que se ha escrito el programa que debe completar el sujeto. En concreto, el sujeto debe completar el bucle definido: “repetir ¿? veces”; siendo las alternativas de respuesta: “1 vez”, “2 veces”, “3 veces”, y “5 veces”. La formulación del reactivo induce, por la pista gramatical que contiene, a eliminar automáticamente la opción de respuesta “1 vez”, que precisamente es la alternativa correcta. Por ello, proponemos una revisión del ítem para futuras versiones del test, reformulado según la siguiente Figura 6.59.


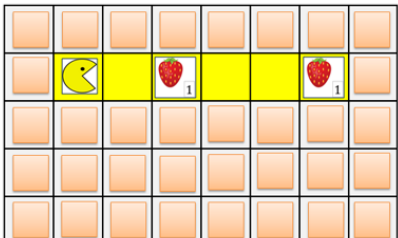
<p>¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas?</p>  	Opción A ✔ 1 vez
	Opción B 2 veces
	Opción C 3 veces
	Opción D 5 veces

Figura 6.59. Propuesta de revisión futura para el ítem 23. El bucle definido “repetir ¿? veces” se reformula como “repetir????”

Con respecto a los objetivos específicos O_{3g} y O_{3h} , podemos concluir que:

- La fiabilidad del TPC, como consistencia interna, es $\alpha = 0,793 \approx 0,80$; lo cual, puede considerarse como ‘buena fiabilidad’ (Navas Ara, 2001). Con vistas a una posible mejora futura de nuestro instrumento, destacamos que la consistencia interna del TPC sólo se incrementa eliminando el ítem 15 (nuevo $\alpha = 0,798$; lo que supone un incremento, prácticamente inapreciable, de +0,005). Eliminando el ítem 12 o el ítem 23, la fiabilidad no se resiente (se mantiene $\alpha = 0,793$).

- El Alfa de Cronbach aumenta con el ciclo educativo y curso académico (llega hasta un máximo de $\alpha = 0,854$ en 4º de la ESO). Es decir, el TPC gana en consistencia interna según aumenta la edad de los sujetos, que cometen un menor número de fallos o aciertos ‘inesperados’ en los ítems del test, dado su rendimiento global en el mismo.
- Por otro lado, nos parece de gran interés el resultado que indica un Alfa de Cronbach distinto en función del dispositivo electrónico (ordenador o tableta) en el que fue aplicado el TPC. Así, la consistencia interna del TPC es mayor cuando se aplica en dispositivos tipo ‘tablet’ (la consistencia interna sube a $\alpha = 0,817$); efecto especialmente notable en el 1º Ciclo de Educación Secundaria, foco central del instrumento, en donde el valor Alfa de Cronbach asciende hasta un notable $\alpha = 0,836 \approx 0,84$. En síntesis, la precisión de nuestro test aumenta cuando se aplica sobre dispositivos móviles; quizás porque éstos permiten al sujeto girar la pantalla a un lado y a otro, orientando a su gusto los reactivos del test y reduciendo su carga cognitiva ‘espacial’, evitando en última instancia cometer ‘fallos tontos o inesperados’. Esta interpretación es apoyada por los resultados que se obtienen al comparar el rendimiento promedio en el TPC entre sujetos que lo realizaron en ordenador fijo y sujetos que lo realizaron en tableta: por ejemplo, en 1º Ciclo de la ESO, $\bar{X}_{\text{ordenador}} = 16,01$ vs. $\bar{X}_{\text{tableta}} = 18,24$ ($t = -4,116$; $p_{(t)} = 0,000 < 0,01$; $d = 0,46$). En el futuro, cuando hayamos acumulado una muestra de mayor tamaño que haya realizado el TPC sobre tableta, y si se siguen manteniendo estas diferencias significativas, será necesario establecer baremos diferenciados para el instrumento en función del dispositivo de aplicación.
- Finalmente, la fiabilidad del TPC como estabilidad temporal (ρ_{xx}), calculada como correlación entre las puntuaciones totales en el mismo, derivadas de dos aplicaciones sucesivas a los mismos sujetos (10-12 semanas entre ambas aplicaciones), es $\rho_{xx} = 0,704 \approx 0,70$; que puede considerarse un valor más que aceptable (Navas Ara, 2001)

Con respecto a los objetivos específicos $O_{31} \dots O_{31}$, podemos concluir que:

- El TPC presenta una correlación estadísticamente muy significativa ($p_{(r)} < 0,01$), positiva y de intensidad moderada con el PMA-R (factor ‘razonamiento’; $r = +0,442$) y con el PMA-E (factor ‘espacial’; $r = +0,439$), y de intensidad sólo baja con el PMA-V (factor ‘verbal’; $r = +0,273$). Por otro lado, el TPC no correlaciona de manera estadísticamente significativa ($p_{(r)} > 0,05$) con el PMA-N (factor ‘numérico’; $r = -0,157$).
- El modelo de regresión múltiple construido a partir de las puntuaciones en la batería PMA correlaciona $r = +0,540$ con el TPC; lo que supone un R^2_{ajustado} de 0,27 (el 27,0% de la varianza del TPC es explicado a partir de una combinación lineal de las aptitudes primarias medidas a través del PMA). Dicho modelo de regresión es capaz de explicar-predecir, de manera estadísticamente muy significativa, las diferencias en las puntuaciones del TPC ($F = 13,457$; $p_{(F)} = 0,000 < 0,01$). Ahora bien, sólo el PMA-R ($\beta_t = 0,265$) y el PMA-E ($\beta_t = 0,308$) son capaces, de manera específica y estadísticamente muy significativa ($p_{(t)} < 0,01$), de explicar-predecir cambios en la variable dependiente del modelo (TPC). Estos resultados son consistentes con la diferenciación que hacen O’Brien *et al.* (2005) sobre los ‘programmers’ frente a los ‘interfacers’; destacando los primeros por sus habilidades lógicas y espaciales.

Con respecto al objetivo específico O_{3m} , podemos concluir que:

- Existe una correlación estadísticamente muy significativa ($p_{(r)} < 0,01$), de signo positivo y de intensidad moderada-alta ($r = +0,669$) entre el TPC y el Test de Resolución de Problemas RP30; lo que supone un notable coeficiente de determinación ($R^2 = 0,447$) entre las puntuaciones arrojadas por ambos instrumentos (44,7% de varianza compartida)

Tomados conjuntamente los resultados de validez criterial (concurrente) entre el TPC y el PMA, y entre el TPC y el RP30; podemos afirmar que son muy consistentes con recientes hallazgos (Ambrosio, Xavier, & Georges, 2014) que vinculan el constructo psicológico ‘pensamiento computacional’ con algunos de los factores cognitivos definidos desde el modelo de inteligencia Cattell-Horn-Carroll (CHC) (McGrew, 2009). En concreto, Ambrosio *et al.* (2014) relacionan el pensamiento computacional con los factores *Gf* (*‘Fluid reasoning’*), *Gv* (*‘Visual processing’*), y *Gsm* (*‘Short-term memory’*) del modelo CHC. Siguiendo la descripción que McGrew (2009) ofrece de estos tres factores de inteligencia, podemos alinearlos claramente con algunos de los instrumentos que hemos utilizado:

- *Gf* (*‘Fluid reasoning’*): es decir, ‘razonamiento fluido’. Se define como “*el uso deliberado y controlado de operaciones mentales dirigidas a resolver nuevos problemas que no pueden ser ejecutados automáticamente. Estas operaciones mentales habitualmente incluyen: realización de inferencias, formación de conceptos, clasificación, generación y comprobación de hipótesis, identificación de relaciones, comprensión de las implicaciones, resolución de problemas, extrapolación y transformación de información. El razonamiento inductivo y deductivo son considerados los principales indicadores de Gf*” (McGrew, 2009, p. 5). El factor *Gf* es abordado de manera clara por el PMA-R y el RP30.
- *Gv* (*‘Visual processing’*): es decir, ‘procesamiento visual-espacial’. Se define como “*la capacidad para generar, almacenar, recuperar, y transformar imágenes y sensaciones visuales. El factor Gv se mide habitualmente por tareas (a partir de estímulos figurativos o geométricos) que requieren la percepción y transformación de formas, figuras o imágenes visuales y/o tareas que requieren mantener la orientación espacial de un objeto que puede moverse o girar en un espacio definido*” (McGrew, 2009, p. 5). El factor *Gv* es abordado de manera clara por el PMA-E y el RP30.
- *Gsm* (*‘Short-term memory’*): es decir, ‘memoria a corto plazo’. Se define como “*la capacidad de aprehender y mantener la conciencia sobre un número limitado de elementos de información, en la situación inmediata (hechos ocurridos en el último minuto o similar). Es un sistema de capacidad limitada que pierde rápidamente la información a través del decaimiento de las trazas de la memoria, a menos que el sujeto active otros recursos cognitivos para mantener la información en la conciencia inmediata*” (McGrew, 2009, p. 5). El factor *Gsm* es abordado de manera clara por el RP30.

Así las cosas, encontramos una correspondencia clara entre la magnitud de las correlaciones encontradas entre el TPC con los distintos factores del PMA y con el RP30, y la composición factorial esperada del pensamiento computacional desde el modelo CHC (Tabla 6.77). Es a nuestro entender, una potente evidencia de validez criterial (concurrente) de nuestro TPC.

Tabla 6.77. Correlaciones TPC*PMA y TPC*RP30, y contingencia con los factores *Gf*, *Gv*, y *Gsm* del modelo CHC

	PMA-N	PMA-V	PMA-E	PMA-R	RP30
TPC	-,157	,273**	,439**	,442**	,669**
¿Se mide a lo largo de los siguientes instrumentos?					
<i>Gf</i>	No	No	No	Sí	Sí
<i>Gv</i>	No	No	Sí	No	Sí
<i>Gsm</i>	No	No	No	No	Sí

** Correlación significativa al nivel $p_{(r)} < 0,01$

Con respecto al objetivo específico **O_{3n}**, podemos concluir que:

- Encontramos correlaciones no estadísticamente significativas ($p_{(r)} > 0,05$), de sentido positivo y de intensidad baja, entre el TPC y las dos puntuaciones derivadas de la aplicación del test FI-R: el ‘Índice de Percepción y Atención’ ($r = +0,229$) y el ‘Índice de Control de la Impulsividad’ ($r = +0,204$). Se intuye, pues, cierta relación directa de la aptitud perceptivo-atencional con nuestro TPC, algo esperable dadas algunas definiciones del pensamiento computacional que incluyen la ‘persistencia’ como elemento relevante en el mismo (CSTA & ISTE, 2015). En cualquier caso, nuestros resultados no son concluyentes y requieren de mayor investigación al respecto.

Con respecto al objetivo específico **O_{3ñ}**, podemos concluir que:

- El TPC correlaciona de manera estadísticamente muy significativa ($p_{(r)} < 0,01$), en sentido positivo, con intensidad moderada con la dimensión ‘Apertura’ ($r = +0,407$) y con intensidad baja con las dimensiones ‘Extraversión’ ($r = +0,304$) y ‘Conciencia’ ($r = +0,267$). No se encuentran correlaciones estadísticamente significativas ($p_{(r)} > 0,05$) ni con la dimensión ‘Amabilidad’ ($r = +0,133$), ni con la dimensión ‘Inestabilidad Emocional’ ($r = +0,092$).
- El modelo de regresión múltiple construido a partir de las puntuaciones en las dimensiones de personalidad del BFQ-NA correlaciona $r = +0,529$ con el TPC; lo que supone un $R^2_{ajustado} = 0,241$ (el 24,1% de la varianza del TPC es explicado a partir de una combinación lineal de las dimensiones de personalidad medidas a través del BFQ-NA). Dicho modelo de regresión es capaz de explicar-predecir, de manera estadísticamente muy significativa, las diferencias en las puntuaciones del TPC ($F = 7,216$; $p_{(F)} = 0,000 < 0,01$). Ahora bien, sólo la dimensión ‘Apertura’ ($\beta_{t(Ap)} = 0,618$) y la dimensión ‘Extraversión’ ($\beta_{t(Ex)} = 0,365$) son capaces, de manera específica y estadísticamente muy significativa ($p_{(t)} < 0,01$), de explicar-predecir cambios en la variable dependiente (TPC).
- En conjunto, estos resultados son parcialmente consistentes con la investigación previa sobre la relación entre las distintas dimensiones de personalidad del modelo de 5 factores (*‘big five’*) con variables cognitivas (Barbaranelli, Caprara, Rabasca, & Pastorelli, 2003; Chamorro-Premuzic & Furnham, 2008). En dichos estudios previos, se señalan como dimensiones de personalidad que correlacionan positivamente con variables cognitivas (tipo *Gf* o *CI* de coeficiente intelectual) y de rendimiento académico, a la ‘Apertura’³⁸⁹ y la

³⁸⁹ De hecho, a la dimensión ‘Apertura’ también se la denomina, a veces, ‘Intelecto’ (Barbaranelli, Caprara, Rabasca, & Pastorelli, 2003)

‘Conciencia’ (de hecho, en nuestro estudio encontramos una correlación altísima entre ambas dimensiones de personalidad: $r = +0,834$); pero no se señala a la ‘Extraversión’ (incluso, [Barbaranelli et al., 2013](#) informan de leves correlaciones negativas entre ‘Extraversión’ y rendimiento académico). Así pues, parece que nuestro TPC introduce una correlación positiva no esperada con la dimensión ‘Extraversión’, que requiere de una investigación más profunda.

Tomados conjuntamente los modelos de regresión sobre el TPC, uno construido a partir de las aptitudes mentales primarias del PMA y otro construido a partir de las dimensiones de personalidad del BFQ-NA, es especialmente destacable que su poder explicativo-predictivo se sitúe en niveles similares ($R^2_{\text{ajustado}} = 0,27$ en el modelo PMA vs. $R^2_{\text{ajustado}} = 0,24$ en el modelo BFQ-NA). Ello nos indica que para explicar el nivel de desempeño con el TPC es necesario tomar conjuntamente variables cognitivas y de personalidad; tal y como sugieren las recientes definiciones operativas del pensamiento computacional ([CSTA & ISTE, 2015](#)), que definen a éste como un conjunto particular de habilidades de solución de problemas, potenciadas por actitudes esenciales como: la apertura y la confianza para tratar con la ambigüedad y la complejidad, la persistencia en el trabajo con tareas difíciles, la capacidad para manejarse con problemas abiertos (*‘open-ended’*), o la capacidad para comunicarse con otros para llegar a una meta o solución común.

Con respecto a los objetivos específicos O_{3o} y O_{3p} , de validez criterial (predictiva), podemos concluir que:

- El TPC, aplicado al principio del trimestre, correlaciona en sentido positivo y de manera estadísticamente muy significativa ($p_{(r)} < 0,01$) con las calificaciones académicas emitidas al final de dicho trimestre; concretamente, con intensidad moderada en la asignatura de Informática ($r = +0,412$) y con intensidad baja en las asignaturas de Lengua ($r = +0,361$) y Matemáticas ($r = +0,333$). Este último resultado es consistente con los hallazgos de Lewis y Shah ([2012](#)), que informan de de una correlación positiva, en niños de 5º de Primaria, entre el nivel de desempeño en retos o puzzles de programación (*‘programming quizzes’*) y sus calificaciones académicas en matemáticas.
- El modelo de regresión múltiple construido a partir de las calificaciones en dichas tres asignaturas al final del trimestre, correlaciona $r = +0,446$ con la puntuación en el TPC al inicio del mismo; lo que supone un $R^2_{\text{ajustado}} = 0,184$. Dicho modelo es capaz de explicar-predecir, de manera retrospectiva y estadísticamente muy significativa, las puntuaciones totales en el TPC a partir de las calificaciones en las tres asignaturas, emitidas 12 semanas más tarde ($F = 12,773$; $p_{(F)} < 0,01$); aunque sólo la nota de Informática ($\beta_t = 0,300$) es la que, de manera específica y estadísticamente muy significativa ($p_{(t)} < 0,01$) explica-predice diferencias en el TPC.
- Encontramos sendas correlaciones positivas y estadísticamente muy significativas ($p_{(r)} < 0,01$) entre el TPC, aplicado al principio del trimestre, y las estadísticas de desempeño (o *‘tracking’*) al final mismo de los estudiantes que siguieron el curso *‘K-8 Intro to Computer Science’* ([Code.org, 2015a](#)): tanto con el número de ‘niveles-pantallas completados’ ($r = +0,331$) como con el número de ‘líneas de código escritas’ ($r = +0,254$)

- Es importante hacer 2 matizaciones a las conclusiones anteriores de validez criterial (predictiva). En primer lugar, la correlación predictiva del TPC con la nota de Informática sólo aparece cuando el currículum seguido durante el trimestre incluye contenidos relativos a la programación y el pensamiento computacional ($r = +0,483$, $p_{(r)} < 0,01$), no así cuando el currículum seguido es el tradicional³⁹⁰, basado en el uso de herramientas ofimáticas u otras aplicaciones, y en la navegación web ($r = -0,003$; $p_{(r)} > 0,05$). Este valor predictivo diferencial del TPC sugiere que ambos tipos de currículum implican procesos cognitivos distintos. En segundo lugar, la correlación predictiva del TPC con las estadísticas finales de desempeño en el curso de Code.org, se maximiza cuando el ritmo de avance por dicho curso lo marca el propio estudiante o metodología ‘*self-paced*’ (en vez de un ritmo homogéneo para todo el aula marcado por el profesor o metodología ‘*teacher-paced*’). Dicho de otra manera, que el profesor permita aprovechar esta plataforma de aprendizaje para individualizar el ritmo de avance de los estudiantes, es un hecho capital para equiparar su *potencial computacional* (aptitud medida por el TPC) con su *rendimiento*; equiparación que es una de las premisas y aspiraciones básicas establecidas desde la Pedagogía Diferencial (Jiménez Fernández & González Galán, 2010; Tourón, 2010)

Con respecto al objetivo específico **O_{3q}**, podemos concluir que:

- Centrándonos en la sub-muestra de estudiantes, todos ellos de 1º Ciclo de la ESO, que siguieron el curso *K-8* de Code.org según una metodología ‘*self-paced*’ que, como hemos visto, es el contexto en el cual su *potencial computacional* pudo manifestarse en forma de *rendimiento*; definimos a los sujetos de ‘alta capacidad computacional’ como aquéllos que completaron dicho curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) a un ritmo notablemente más rápido que sus compañeros de aula, y solicitaron acelerar a un curso de nivel superior: el denominado ‘*Computer Programming*’ de la plataforma Khan Academy (2015).
- Se encuentra una diferencia estadísticamente significativa en el TPC aplicado al comienzo del trimestre ($t = -2,300$; $p_{(t)} = 0,026 < 0,05$), y con un tamaño del efecto ‘grande’ ($d = 0,95$), a favor del grupo categorizado posteriormente como de ‘alta capacidad computacional’ frente al grupo ‘normal’. La media en el TPC para el grupo de ‘alta capacidad computacional’ fue de 21,57; puntuación que se sitúa casi una desviación típica por encima de la media en el grupo ‘normal’, y equivalente a casi un percentil 90 en el baremo de 1º Ciclo de la ESO.
- El área bajo la curva COR de potencia diagnóstica del TPC sobre la ‘alta capacidad computacional’ es $A_{COR} = 0,75$. Este resultado es estadísticamente significativo ($p = 0,037 < 0,05$); y puede interpretarse como una potencia discriminante ‘buena’ (Navas Ara, 2001) del TPC sobre la ‘alta capacidad computacional’. Adicionalmente, encontramos un punto de corte de extraordinario interés: según la curva COR, una puntuación $\geq 24,5$ en el TPC (es decir, una puntuación 25, 26, 27 o 28), en 1º Ciclo de la ESO, consigue detectar un 42,9% de ‘verdaderos positivos’, y sólo un 7,9% de ‘falsos positivos’ en lo relativo a la ‘alta capacidad computacional’, tal y como la hemos definido.

³⁹⁰ Denominado en el ámbito anglosajón como currículum ICT (“*Information and Communication Technologies*”); que podría traducirse como currículum TIC (“*Tecnologías de la Información y la Comunicación*”)

- En conjunto, se encuentran evidencias para afirmar la validez discriminante del TPC sobre la ‘alta capacidad computacional’, si bien el pequeño tamaño de la sub-muestra utilizada para este estudio (N=45) aconseja cierta prudencia y nuevas replicaciones.

Con respecto al objetivo específico O_{3r} , podemos concluir que:

- Las puntuaciones totales en el TPC correlacionan de manera estadísticamente muy significativa ($p_{(r)} < 0,01$), en sentido positivo y con intensidad moderada, con las puntuaciones totales en las tareas Bebras B-1 ($r = +0,511$), y tareas Bebras B-2 ($r = +0,523$). No se encuentra correlación estadísticamente significativa entre el TPC y el total de las tareas Bebras B-3 ($r = +0,175$). En conjunto, estos resultados apoyan la validez convergente entre el TPC y las tareas Bebras, si bien no son plenamente concluyentes.
- Si hacemos un análisis pormenorizado de las tareas B-2, encontramos que:
 - El TPC correlaciona de manera estadísticamente muy significativa ($p_{(r)} < 0,01$), en sentido positivo y con intensidad moderada, con la tarea B2-1 ($r = +0,427$), denominada ‘*Water Supply*’ (**Anexo H**). Esta tarea consiste en una estructura binaria: una serie de tuberías con grifos que pueden estar ‘abiertos’ o ‘cerrados’; para solucionar la tarea el sujeto debe aplicar el pensamiento condicional sobre dicha estructura. La tarea ‘*Water Supply*’ está alineada con los ítems del TPC en los cuales el sujeto debe aplicar igualmente el pensamiento condicional binario sobre, por ejemplo, las intersecciones que aparecen el laberinto. Consideramos, pues, que el valor de correlación encontrado es consistente. Adicionalmente, esta es la única tarea en la que se encontró un rendimiento significativamente superior en los chicos (51% de acierto) frente a las chicas (36% de acierto) ($t = 2,147$; $p_{(t)} = 0,033 < 0,05$); lo cual es consistente con la reciente investigación sobre desempeño en tareas Bebras (Bellettini *et al.*, 2015; Hubwieser & Mühling, 2014, 2015; Kalelioglu, Gülbahar, & Madran, 2015; Lee, Lin, & Lin, 2014), *corpus* que reporta rendimientos significativamente superiores en los chicos frente a las chicas, en la mayoría de las tareas Bebras.
 - El TPC correlaciona de manera estadísticamente muy significativa ($p_{(r)} < 0,01$), en sentido positivo y con intensidad moderada, con la tarea B2-3 ($r = +0,483$), denominada ‘*Abacus*’ (**Anexo H**). En esta tarea, el sujeto debe ser capaz de descomponer el estímulo que se le plantea (el ‘ábaco’), aplicando-secuenciando sobre el mismo la regla o algoritmo que se le indica. Por tanto, la tarea ‘*Abacus*’ está alineada con todos los ítems del TPC que implican la secuenciación de un conjunto de instrucciones; y especialmente con los ítems del TPC que abordan el concepto computacional de ‘secuencias y direcciones básicas’. Consideramos, pues, que el valor de correlación encontrado es consistente.
 - Sin embargo, el TPC no correlaciona significativamente con la tarea B2-2 ($r = +0,055$) denominada ‘*Fast Laundry*’ (**Anexo H**). Para contestar correctamente a esta tarea, el sujeto debe entender que se pueden ir ejecutando varias acciones simultáneamente y en paralelo. Por tanto, la tarea ‘*Fast Laundry*’ se alinea con el

aspecto del pensamiento computacional habitualmente denominado ‘paralelización’, que no es abordado explícitamente en nuestro TPC (no hay ningún ítem del TPC que exija al sujeto ejecutar 2 o más algoritmos simultáneamente). Así, parece consistente que ésta sea la única tarea Bebras B2 en la cual no se encuentre correlación con el TPC.

Con respecto al objetivo específico O_{3s}, podemos concluir que:

- Encontramos correlaciones estadísticamente muy significativas ($p_{(r)} < 0,01$), positivas y de intensidad moderada, del TPC con Dr. Scratch. Tanto la correlación convergente-predictiva, es decir, $TPC_{pre} * Dr. Scratch$ ($r = +0,439$); como la correlación convergente-concurrente, es decir, $TPC_{post} * Dr. Scratch$ ($r = +0,535$). De manera coherente, el segundo valor es mayor que el primero. En conjunto, estos resultados aportan evidencia de la validez convergente de nuestro TPC respecto a Dr. Scratch.
- Tal y como era de esperar, las correlaciones son mayores en la situación de ‘*single programming*’, pues a cada sujeto le corresponde una puntuación diferente en Dr. Scratch, que a su vez puede correlacionar con sus respectivas puntuaciones en el TPC *pre* y *post*; frente a la situación de ‘*pair programming*’, en la que a cada par de sujetos le corresponde la misma puntuación en Dr. Scratch, reduciendo la probabilidad de correlacionar con sus puntuaciones individuales en el TPC *pre* y *post*.
- Con respecto a las correlaciones encontradas entre el TPC, tanto *pre* como *post*, con las distintas dimensiones del pensamiento computacional que mide Dr. Scratch, aportamos una posible explicación: los 28 ítems del TPC miden, de menor a mayor dificultad, conceptos computacionales que se alinean progresivamente con las siguientes dimensiones de Dr. Scratch (Tabla 6.78): ‘*Control de Flujo*’, ‘*Sincronización*’, ‘*Pensamiento Lógico*’, ‘*Representación de la Información*’ y ‘*Abstracción*’. Consistentemente, las 3 primeras dimensiones, que son las abordadas por la primera parte del TPC (la más sencilla, y más abordable para el sujeto en situación *pre*), son las únicas con correlaciones $r > +0,40$ con el TPC tanto en situación *pre* como *post* (valores r ligeramente mayores con el *post*). La dimensión ‘*Representación de la Información*’, que aparece hacia el final del TPC, experimenta un incremento más notable en la correlación entre la situación *pre* (0,35) y la *post* (0,45). En cuanto a la dimensión ‘*Abstracción*’, abordada en la parte final y más compleja del TPC (ítems 25-28, sobre ‘*Funciones simples*’, que son menos abordables para los sujetos en situación pretest) no tiene correlación significativa en el *pre* pero sí luego en el *post*.

Tabla 6.78. Alineamiento de los conceptos computacionales medidos por el TPC, con las dimensiones del pensamiento computacional medidas por Dr. Scratch

<i>Conceptos Computacionales medidos por el TPC (de menor a mayor dificultad)</i>	<i>Dimensiones del pensamiento computacional medidas por Dr. Scratch</i>
Secuencias básicas (direcciones y giros) – Ítems 1 a 4	Control de Flujo
Bucle ‘repetir veces’ – Ítems 5 a 8	Control de Flujo

<i>Conceptos Computacionales medidos por el TPC (de menor a mayor dificultad)</i>	<i>Dimensiones del pensamiento computacional medidas por Dr. Scratch</i>
Bucle ‘repetir hasta’ – Ítems 9 a 12	Control de Flujo + Sincronización
Condicional simple ‘if’ – Ítems 13 a 16	Control de Flujo + Sincronización + Pensamiento Lógico
Condicional compuesto ‘if/else’ – Ítems 17 a 20	Control de Flujo + Sincronización + Pensamiento Lógico
Condicional ‘mientras’ (‘while conditional’) – Ítems 21 a 24	Control de Flujo + Sincronización + Pensamiento Lógico + Representación de la Información
Funciones simples – Ítems 25 a 28	Control de Flujo + Abstracción

- Adicionalmente, encontramos una diferencia estadísticamente muy significativa ($t = -3,334$; $p_{(t)} = 0,001 < 0,01$), con un tamaño del efecto asociado $d = 0,37$, entre la puntuación media en el TPC en condición *pre* ($\bar{X} = 17,07$) y en condición *post* ($\bar{X} = 18,58$). Ello es un indicador de que nuestro TPC es sensible para detectar mejoras *pre-post* en pensamiento computacional tras un periodo de aprendizaje en un contexto ‘open-ended’, como la plataforma Scratch, y no sólo en contextos ‘close-ended’ (como veremos al evaluar los cursos K-8 y K-5 de Code.org en el Capítulo 7).

Tomando conjuntamente los objetivos O_{3r} y O_{3s} , de validez convergente de nuestro TPC con otras medidas del pensamiento computacional; podemos afirmar que, de manera global, se encuentran evidencias de dicha validez, tanto con las tareas Bebras como con Dr. Scratch. Aunque serían deseables valores algo mayores de correlación (p.e. Carlson & Herdman (2012) fijan en $r \geq 0,7$ el punto de corte para hablar de validez convergente ‘buena’), hay que recordar que cada una de las medidas utilizadas parte de perspectivas distintas. Así:

- El TPC es una prueba de respuesta individual y de ‘ejecución máxima’; es decir, se mide la aptitud ‘pensamiento computacional’ en cada sujeto a través de un test de elección múltiple de 28 ítems (4 opciones de respuesta, sólo 1 correcta), y los sujetos tratan de obtener la máxima puntuación posible.
- Las tareas Bebras no forman por sí mismas un instrumento de medida del pensamiento computacional con propiedades psicométricas probadas. Aunque son tareas de respuesta individual y de ‘ejecución máxima’ y, como en el TPC, con formato de elección múltiple con 4 opciones de respuesta; las tareas Bebras no miden ‘aptitud’ en sentido estricto, sino más bien ‘competencia’, en un sentido análogo al de las tareas que se plantean en las pruebas PISA (Hubwieser & Mühling, 2014). Dicho de otra manera, las tareas Bebras no se enfocan a la medida del ‘pensamiento computacional’ en un estado aptitudinal puro (enfoque del TPC), sino más bien a la medida de cómo el sujeto es capaz de transferir dicho pensamiento computacional a problemas cotidianos y significativos.
- Por su parte, Dr. Scratch es una herramienta que, estrictamente, mide y analiza proyectos, no sujetos; ello permite tener tanto situaciones ‘single programming’ (1 proyecto hecho por 1 sujeto, y a dicho sujeto se le asigna la puntuación obtenida por su proyecto con Dr. Scratch), y ‘pair programming’ (1 proyecto hecho por 2 sujetos, y a ambos se les asigna la puntuación

obtenida por el proyecto común en Dr. Scratch; resintiéndose los valores de correlación con el TPC). Además, no se puede asegurar que el proyecto Scratch analizado sea el de ‘ejecución máxima’ del estudiante: puede haber sido seleccionado y enviado por el profesor/a para el estudio por otras razones (p.e. que se asemeje lo más posible al ‘prototipo’ de proyecto que el docente ha ‘impartido’ en la clase)

Dadas estas circunstancias, los valores obtenidos de validez convergente ($r \approx +0,5$) nos parecen muy relevantes. Estos valores apoyan la afirmación de que TPC, tareas Bebras y Dr. Scratch miden el pensamiento computacional de manera convergente, aunque no idéntica, potenciando su uso complementario. Y es que, haciendo un análisis crítico de las tres medidas, observamos que las virtudes de cada una son las debilidades de las otras:

- El TPC es un instrumento de tipo aptitudinal-diagnóstico de orientación sumativa. Como virtudes tiene que permite: aplicaciones *pre* (sin experiencia previa en programación); cribados masivos; construcción de baremos a gran escala; diagnóstico de necesidades educativas especiales en pensamiento computacional (tanto por alta como por baja aptitud); o evaluación *pre-post* de currículos de programación (tal y como veremos en el Capítulo 7). Como debilidades, el TPC ofrece una evaluación estática y descontextualizada; excesivamente centrada en ‘conceptos computacionales’ en terminología de Brennan y Resnick (2012) (no contempla ni ‘prácticas computacionales’ ni ‘perspectivas computacionales’).
- Las tareas Bebras son un conjunto de problemas que, aunque no constituyen por sí mismas un instrumento de medida validado, sirven para promover el pensamiento computacional en el aula desde una perspectiva competencial. Al estilo de las pruebas PISA, las tareas Bebras pueden servir para realizar oleadas sucesivas de evaluación del pensamiento computacional, desde una perspectiva internacional y comparada. Adicionalmente, las tareas Bebras que se vayan liberando de las sucesivas oleadas de evaluación, pueden incorporarse como elementos de enseñanza-aprendizaje del pensamiento computacional en el aula, dado que son tareas con carácter cotidiano y significativo. En esa misma línea, el Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado (INTEF), viene promoviendo desde hace un par de años la incorporación de preguntas liberadas de las pruebas PISA (y TIMMS³⁹¹) a la práctica de enseñanza-aprendizaje en el aula (INTEF, 2014)
- Por último, Dr. Scratch es un instrumento de evaluación de proyectos de orientación formativa. Como virtudes tiene que permite: una evaluación continua a lo largo del proceso de aprendizaje; un ‘*feed-back*’ formativo para estudiante y profesor (Moreno-León, Robles, & Román-González, 2015); que abarca no sólo ‘conceptos computacionales’ sino también ‘prácticas computacionales’ (principalmente ‘iteración’ al promover la mejora continua de un proyecto en ciclos sucesivos, y ‘depuración’ al ofrecer indicadores/avisos sobre malos hábitos de programación). Como debilidades tiene que: no permite aplicaciones *pre-test* (es decir, exige un periodo mínimo inicial de aprendizaje con Scratch), y puede resultar muy condicionado por la metodología de enseñanza-aprendizaje de Scratch utilizada por el profesor/a en el aula.

³⁹¹ TIMMS: *Trends in International Mathematics and Science Study*

Con respecto al objetivo específico **O_{3t}**, podemos concluir que:

- Encontramos una solución factorial inicial de 7 factores, que explica un 41,87% de la varianza total de las puntuaciones en el TPC. Una vez analizada la estructura y el contenido de los mismos, consideramos que sólo los 3 primeros son susceptibles de ser razonablemente interpretados, y de generar subescalas con una fiabilidad mínimamente suficiente ($\alpha > 0,5$). Estos 3 factores, que explican el 26,41% de la varianza total de las puntuaciones, se describen en los siguientes términos:
 - **Factor 1:** compuesto por los ítems 2, 3, 5, 6, 9 y 11; todos ellos giran en torno de los conceptos computacionales ‘direcciones-secuencias básicas’ y ‘bucles’ (tanto ‘repetir veces’ como ‘repetir hasta’). La subescala derivada (*‘Secuencias y Bucles’*) presenta una fiabilidad suficiente de $\alpha = 0,621$. Esta subescala, que está compuesta por ítems fáciles ($\bar{X} = 5,15$ sobre 6 puntos posibles) y situados al comienzo del TPC, es la única que correlaciona de manera significativa con el ‘Índice de Control de la Impulsividad’ del FI-R ($r = +0,314$; $p_{(r)} = 0,014 < 0,05$); dicho de otra manera, una puntuación anormalmente baja en la subescala *‘Secuencias y Bucles’* podría servir de indicador de que el sujeto tiene dificultades para controlar sus impulsos. Además, esta subescala es la única que no correlaciona con la dimensión de personalidad ‘Apertura’ (Ap) del BFQ-NA, que es la habitualmente más vinculada a variables cognitivas y de rendimiento. Todo ello nos lleva a afirmar que la subescala *‘Secuencias y Bucles’* es, en comparación al resto de subescalas, de menor carga cognitiva y de mayor carga atencional.
 - **Factor 2:** compuesto por los ítems 20, 24, 26, 27 y 28; la mayoría de ellos giran en torno del concepto computacional ‘funciones simples’. La subescala derivada (*‘Funciones’*) presenta una fiabilidad suficiente de $\alpha = 0,589$. Esta subescala, que está compuesta por ítems de dificultad media-baja ($\bar{X} = 3,01$ sobre 5 puntos posibles) y situados al final del TPC, presenta correlaciones muy significativas, y de especial intensidad, con las dimensiones ‘Conciencia’ ($r = +0,269$; $p_{(r)} = 0,007 < 0,01$) y ‘Apertura’ ($r = +0,396$; $p_{(r)} = 0,000 < 0,01$) del BFQ-NA. Este resultado es consistente dado que, para tener un buen rendimiento en los ítems de la subescala *‘Funciones’* el sujeto debe conjugar la dimensión cognitiva (identificada con ‘Apertura’) con la exigencia de persistir en el esfuerzo en estos ítems finales del TPC (persistencia que se identifica con la dimensión ‘Conciencia’).
 - **Factor 3:** compuesto por los ítems 4, 14, 16, 22 y 25; la mayoría de ellos giran en torno de los conceptos computacionales ‘condicional simple’ y ‘condicional mientras que’. La subescala derivada (*‘Condicionales’*) presenta una fiabilidad suficiente $\alpha = 0,575$. Esta subescala está compuesta por ítems de dificultad media-alta ($\bar{X} = 2,03$ sobre 5 puntos posibles) y situados en la parte intermedia del TPC. En comparación con las otras subescalas específicas, *‘Condicionales’* presenta correlaciones muy significativas, y de especial intensidad, con el ‘Factor Verbal’ del PMA ($r = +0,246$; $p_{(r)} = 0,002 < 0,01$), y con el ‘Índice de Percepción y Atención’ (A-E) del Test FI-R ($r = +0,376$; $p_{(r)} = 0,003 < 0,01$). Lo primero es consistente con otros estudios que

señalan precisamente a los condicionales como el concepto computacional más próximo al lenguaje natural humano y, por tanto, el más próximo a la aptitud verbal de los sujetos (Howland & Good, 2015). Lo segundo sugiere que la ejecución de condiciones ('estructuras de selección'; en términos de Mühling *et al.*, 2015) es más exigente, desde un punto de vista perceptivo-atencional, que la ejecución de secuencias y bucles ('estructuras de secuenciación' y 'estructuras de repetición', en términos de dichos autores).

- Aun siendo de gran interés, consideramos que la anterior estructura de 3 factores no es del todo concluyente dado que: sólo explica una cuarta parte de la varianza total; los factores 2 ('Funciones') y 3 ('Condicionales') no son del todo coherentes en su composición interna (contienen algún ítem que no aborda explícitamente el concepto computacional debido); y los ítems que componen cada uno de los factores presentan cargas en los mismos de magnitud sólo moderada (entre 0,40 y 0,60). Así las cosas, complementamos la anterior estructura con un análisis unifactorial, que tiene como objeto comprobar qué ítems componen un único factor principal que explique la mayor cantidad de varianza posible (concretamente, se obtiene un 15,93% de varianza explicada por este unifactor). Se comprueba que dicho unifactor principal está atravesado, de manera moderada, por la gran mayoría de los ítems del TPC (véase, 22 de los 28 ítems cargan en este unifactor por encima de 0,30; pero todos por debajo de 0,50). Jugando con el punto de corte, encontramos una subescala muy interesante de 13 elementos, formada por los ítems con saturaciones $\geq 0,38$ en el unifactor (ítems 3, 4, 7, 11, 13, 14, 16, 17, 18, 20, 22, 24 y 27), ya con fiabilidad aceptable $\alpha = 0,741$, y que denominamos '**Común-Principal**'. La subescala '**Común-Principal**' presenta correlaciones con respecto a las medidas de validez criterial concurrente similares a las obtenidas con el TPC. Sin embargo, encontramos 2 diferencias que nos inclinan a afirmar la especial bondad psicométrica de esta subescala.
 - Mientras que el TPC, sorprendentemente, no correlacionó de manera significativa con el 'Índice de Percepción y Atención' (A-E) del FI-R ($r = +0,229$; $p_{(r)} = 0,076 > 0,05$); la subescala '**Común-Principal**' sí lo hace, y de manera muy significativa ($r = +0,350$; $p_{(r)} = 0,006 < 0,01$)
 - Mientras que el TPC, sorprendentemente, correlacionó de manera positiva y muy significativa con la dimensión 'Extraversión' (Ex) del BFQ-NA ($r = +0,30$), con una intensidad muy próxima a la correlación con 'Apertura' (Ap) ($r = +0,40$); la subescala '**Común-Principal**' debilita la correlación con la primera dimensión ($r = +0,24$) al tiempo que intensifica la correlación con la segunda ($r = +0,47$).
 - En síntesis, la subescala '**Común-Principal**' resuelve positivamente los dos resultados inesperados de la validación criterial concurrente del TPC; a saber, no haber encontrado correlación significativa con el 'Índice de Percepción y Atención' (A-E), y haber encontrado una correlación excesivamente alta con la dimensión de personalidad 'Extraversión'. Por ello, afirmamos que esta subescala supera en evidencias de validez criterial concurrente al propio TPC.

- Combinando ambos análisis, llegamos a una versión reducida de nuestro instrumento, que denominamos TPC-R, de tan sólo 20 ítems de longitud; compuesta por todos los ítems que aparecen en las anteriores subescalas: ítems 2, 3, 4, 5, 6, 7, 9, 11, 13, 14, 16, 17, 18, 20, 22, 24, 25, 26, 27 y 28. Consideramos que el TPC-R aporta ciertas virtudes:

 - Supone una reducción en la longitud del test del 28,57%; con las ventajas que ello implica para futuras aplicaciones (p.e. en términos de tiempo necesario para la aplicación del test)
 - El valor de fiabilidad obtenido para el TPC-R ($\alpha = 0,782$) es prácticamente igual que el obtenido para el TPC ($\alpha = 0,793$). Por tanto, la reducción de longitud, de 28 a 20 ítems, prácticamente no incide en la fiabilidad del instrumento.
 - Simplemente aplicando esta versión reducida de 20 ítems, podemos obtener una puntuación total TPC-R (ya con buena fiabilidad y homóloga a la puntuación en el TPC original); y además, obtener las 4 puntuaciones derivadas de las respectivas 4 subescalas (una '*Común-Principal*', y tres específicas: '*Secuencias y Bucles*', '*Funciones*' y '*Condicionales*'). Así, el TPC-R consigue la misma eficacia psicométrica que el TPC, pero con mayor eficiencia.
 - Igualmente, hay que destacar que el TPC-R no incluye en su seno a ninguno de los elementos que se revelaron como débiles durante el análisis de ítems: ítems 12, 15 y 23. Ello es una evidencia adicional de la bondad psicométrica de la versión reducida TPC-R.
 - Finalmente, el comportamiento del TPC-R con respecto a las medidas de validez criterial concurrente, también resuelve las insuficiencias del TPC. Así, el TPC-R correlaciona de manera significativa con el 'Índice de Percepción y Atención' (A-E) del Test FI-R ($r = +0,314$; $p_{(r)} = 0,014 < 0,05$), intensifica la correlación con la dimensión de personalidad 'Apertura' ($r = +0,42$) y debilita la correlación con la dimensión 'Extraversión' ($r = +0,26$)
 - En síntesis, y dadas las virtudes que muestra la versión TPC-R, apostamos por utilizarla en la próxima ronda de aplicaciones empíricas de nuestro instrumento, en la que trataremos de replicar estos mismos resultados, añadiendo un análisis factorial de tipo confirmatorio. Una línea futura de trabajo que queda fuera de esta tesis doctoral.
- Por otro lado, se ha evidenciado que las pruebas PMA-R, PMA-E y RP30 son medidas muy robustas de validez criterial concurrente del TPC, puesto que correlacionan muy significativamente con todas las escalas y subescalas que lo componen. Dicho de otra manera, y de cara a futuras investigaciones en el área, parece que el 'razonamiento lógico', la 'aptitud espacial' y la 'habilidad de solución de problemas' son los principales criterios de validez concurrente para el constructo 'pensamiento computacional'.
- Por el contrario, se ha evidenciado que las dimensiones 'Am' e 'Ie' del BFQ-NA son medidas nulas de validez criterial concurrente del TPC, puesto que no correlacionan con ninguna de las escalas o subescalas que lo componen. Dicho de otra manera, y de cara a futuras

investigaciones en el área, parece que la ‘amabilidad’ y la ‘inestabilidad emocional’ no son en absoluto criterios de validez concurrente para el constructo ‘pensamiento computacional’.

- Recapitulando, y para finalizar la discusión relativa a la validez factorial del TPC: hemos encontrado una estructura factorial mixta (Figura 6.60), que combina un factor común (representado por la subescala ‘Común-Principal’), con 3 factores específicos (representados por las subescalas ‘Secuencias y Bucles’, ‘Funciones’ y ‘Condicionales’) relativamente independientes entre sí (correlaciones $r \approx +0,3$ entre las subescalas específicas); y dicha combinación compone la versión reducida TPC-R, que posee varias ventajas psicométricas respecto del TPC, tal y como acabamos de justificar ¿Tiene sentido la estructura factorial encontrada con respecto a los criterios-ejes de diseño con que fue construido el TPC? Nuestra respuesta es que sí, dadas las siguientes razones:
 - De los 5 ejes de diseño con que fue construido el TPC (‘Concepto computacional abordado’, ‘Entorno-Interfaz del ítem’, ‘Estilo de las alternativas de respuesta’, ‘Existencia o inexistencia de anidamiento’ y ‘Tarea requerida’), los factores específicos se organizan alrededor del eje fundamental de diseño, esto es, alrededor del ‘Concepto computacional abordado’; emergiendo 3 factores que cubren todos los conceptos incluidos en el TPC (secuencias, bucles, condicionales y funciones).
 - Ahora bien, tal y como se aprecia en la [Figura 6.13](#) sobre las especificaciones de los 28 ítems del TPC en los 5 ejes de diseño, éstos se cruzan entre sí a lo largo de todo el instrumento. Así, por ejemplo, aunque el ítem 6 y 28 abordan conceptos computacionales diferentes (‘repetir veces’ y ‘funciones simples’), ambos comparten el requerir al sujeto una tarea de ‘completamiento’. Esta comunalidad puede provocar que la ortogonalidad (independencia) de las subescalas específicas correspondientes (siguiendo con el ejemplo, ‘Secuencias y Bucles’ y ‘Funciones’) se resienta.
 - Además, según se aprecia igualmente en [Figura 6.13](#), los conceptos computacionales van apareciendo progresivamente a medida que se avanza por los ítems del test; pero no lo hacen de manera independiente unos de otros, sino más bien acumulativa. Por ejemplo, en los ítems 1-12 se abordan las direcciones y secuencias básicas así como los bucles; en los siguientes ítems 13-24 se abordan los distintos condicionales, pero en dichos ítems aparecen también direcciones y bucles. Dicho de otra manera, en nuestro TPC los conceptos computacionales no se miden ‘unos y luego otros’, sino más bien ‘unos sobre otros’. Esta aproximación acumulativa provoca igualmente que se resienta la ortogonalidad de las subescalas específicas correspondientes.
 - Por tanto, aunque los distintos conceptos computacionales gozan de cierta independencia entre sí, todos parecen descansar sobre un factor ‘Común-Principal’, acaso eso que hemos venido a definir como ‘pensamiento computacional’. En esta misma línea se manifiestan Mühling *et al.* (2015, p. 4), cuando afirman en la fundamentación de su instrumento: “*Parece que los conceptos [computacionales] que nos ocupan forman parte en última instancia del mismo constructo (...) estos conceptos están lo suficientemente relacionados entre sí como para formar la base de una misma habilidad psicológica [el pensamiento computacional]*”

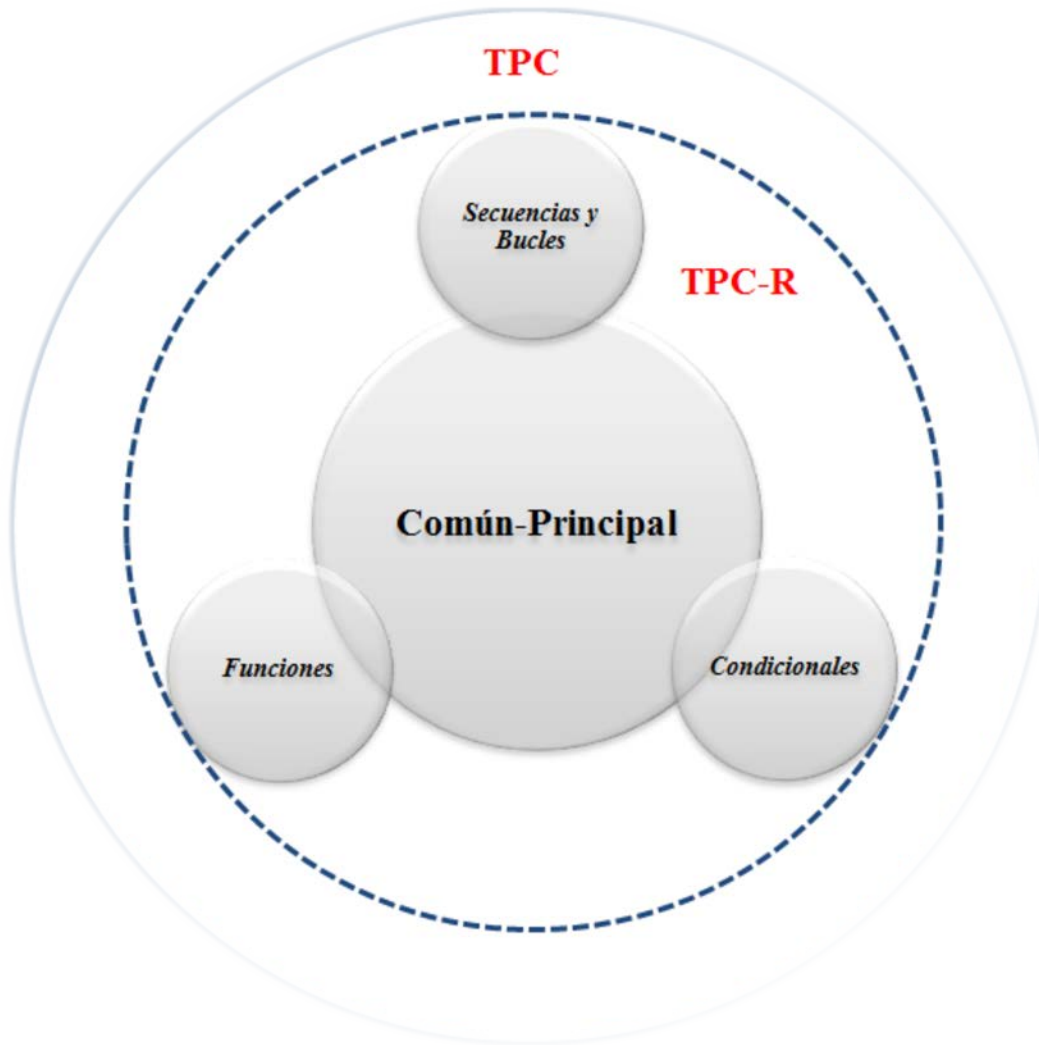


Figura 6.60. Modelo de validez factorial del TPC: una estructura factorial mixta

CAPÍTULO 7

TERCER ESTUDIO: EVALUACIÓN DE PROGRAMAS DE CODIGOALFABETIZACIÓN Y DESARROLLO DEL PENSAMIENTO COMPUTACIONAL

En este capítulo se procede a la evaluación de programas de códigoalfabetización y desarrollo del pensamiento computacional. En primer lugar, se realiza una evaluación principal, en profundidad, sobre el programa-curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) implementado en 1º y 2º Ciclo de la ESO durante el 2º trimestre del año académico 14/15 en 12 centros educativos de la Comunidad Valenciana. En segundo lugar, se realiza una evaluación complementaria, de menor profundidad, sobre el programa-curso ‘*K-5 Computer Science Fundamentals (Course 2)*’ (Code.org, 2015b) implementado en 3º Ciclo de Educación Primaria durante el 2º trimestre del año académico 14/15 en 1 centro educativo de la Comunidad de Andalucía (Sevilla). Para finalizar, se presenta un breve estudio de casos de alumnos con altas capacidades computacionales de primer ciclo de la ESO que, adicionalmente al programa-curso ‘*K-8 Intro to Computer Science*’ seguido por sus compañeros de aula ordinaria, aceleraron hacia un programa-curso de mayor dificultad a través de la plataforma Khan Academy: ‘*Computer Programming*’ (Khan Academy, 2015); mediante una metodología de ‘clase invertida’ (*flipped classroom*) (Tourón & Santiago, 2015).

7.1. Evaluación principal: el curso ‘*K-8 Intro to Computer Science*’ de Code.org

7.1.1. Antecedentes

El curso ‘*K-8 Intro to Computer Science*’ (a partir de ahora simplemente ‘**K-8**’) ha sido descrito con detalle en el sub-epígrafe 3.2.2.1 del Capítulo 3, en la parte teórica de esta tesis doctoral. Tan sólo consideramos conveniente recordar aquí que el curso sigue un modelo ‘*b-learning*’ (*blended learning*) de enseñanza de las Ciencias de la Computación, lo cual implica que los estudiantes aprenden a partir de una mezcla de actividades autoguiadas de ‘*coding on-line*’; y de actividades sin conexión (*unplugged*), que son dirigidas de manera más tradicional por el profesor sin necesidad de uso del ordenador. El conjunto de actividades resultante se estructura en 20 etapas: 9 etapas de ‘*coding on-line*’ y 11 etapas ‘*unplugged*’, que se van alternando a lo largo del itinerario curricular del curso. A su vez, las etapas de ‘*coding on-line*’ están internamente compuestas por una serie de niveles-retos³⁹² de programación; con un total de 98 niveles-retos existentes a lo largo de las 9 etapas. Cada etapa está diseñada para ser abordada en una sesión de clase de alrededor de 45-60 minutos de duración. Habitualmente, este curso se integra en el aula ordinaria impartándose 2 etapas semanales (2 sesiones de clase a la semana), con lo cual su completamiento lleva alrededor de las 10 semanas (aproximadamente un trimestre escolar).

La implantación del curso supone, por tanto, un enriquecimiento curricular trimestral en aula ordinaria sobre códigoalfabetización y desarrollo del pensamiento computacional.

³⁹² A lo largo de la tesis doctoral se utilizan indistintamente los términos ‘niveles-retos’ o ‘niveles-pantallas’ para describir la composición interna de las etapas de ‘*coding on-line*’ que componen los cursos de Code.org.

7.1.2. Hipótesis específicas

En este apartado 7.1 se aborda el objetivo general **O₄** [‘Evaluar el programa-curso ‘K-8’, desde el marco fijado por nuestras hipótesis directivas de investigación’]. Así pues, los objetivos específicos, que serán expresados a modo de hipótesis específicas dado el carácter fundamentalmente *cuasi-experimental* de esta evaluación; están enmarcados en las hipótesis directivas que fijamos en el [apartado 2.6](#) de la parte teórica:

- En el marco de la hipótesis directiva de *adecuación*, nos planteamos las siguientes hipótesis específicas:
 - **H_{4a}**: El curso K-8 mejora la aptitud ‘razonamiento lógico’ de los estudiantes³⁹³
 - **H_{4b}**: El curso K-8 mejora la aptitud ‘espacial’ de los estudiantes
 - **H_{4c}**: El curso K-8 mejora la aptitud ‘numérica’ de los estudiantes
 - **H_{4d}**: El curso K-8 mejora la aptitud ‘verbal’ de los estudiantes
 - **H_{4e}**: El curso K-8 mejora la aptitud ‘resolución de problemas’ de los estudiantes
 - **H_{4f}**: El curso K-8 mejora la aptitud ‘perceptivo-atencional’ de los estudiantes
 - **H_{4g}**: El curso K-8 mejora la aptitud ‘pensamiento computacional’ de los estudiantes
 - **H_{4h}**: El curso K-8 mejora el rendimiento académico de los estudiantes en Informática
 - **H_{4i}**: El curso K-8 mejora el rendimiento académico de los estudiantes en Matemáticas
 - **H_{4j}**: El curso K-8 mejora el rendimiento académico de los estudiantes en Lengua
- En el marco de la hipótesis directiva de *viabilidad*, nos planteamos las siguientes hipótesis específicas³⁹⁴:
 - **H_{4k}**: Más del 50% de los profesores informan de que en los grupos de clase experimentales se han dedicado al menos 8 semanas lectivas al curso K-8
 - **H_{4l}**: Más del 50% de los profesores informan de que en los grupos de clase experimentales se han impartido al menos el 80% de las 20 etapas que componen el curso K-8.
 - **H_{4m}**: Más del 80% de los profesores informan de que en los grupos de clase experimentales se han realizado tanto etapas ‘*on line*’ como etapas ‘*unplugged*’

³⁹³ En comparación con el ‘currículum TIC’ tradicional (este matiz rige para todas las hipótesis específicas dentro de la directiva de *adecuación*). El ‘currículum TIC’ es el seguido por los grupos de control, tal y como se explica más adelante.

³⁹⁴ Para establecer los puntos de corte (puntos de decisión estadística) en las hipótesis específicas dentro de las directivas de *viabilidad* y *relevancia*, nos hemos basado en los resultados previos obtenidos durante el primer estudio empírico, de carácter exploratorio, sobre el evento ‘La Hora del Código’ (Capítulo 5)

- **H_{4n}**: Los profesores consideran que el grado de aprendizaje de sus estudiantes con el curso K-8 ha sido de al menos 4 sobre 5 (Mediana ≥ 4)
 - **H_{4ñ}**: Los profesores muestran una satisfacción global con respecto al curso K-8 en un grado de al menos 4 sobre 5 (Mediana ≥ 4)
 - **H_{4o}**: Más del 80% de los profesores manifiesta su intención de impartir el curso K-8 de nuevo en el próximo curso académico.
 - **H_{4p}**: Más del 50% de los estudiantes completan al menos el 80% de los 98 niveles-pantallas ‘*on-line*’ que componen el curso K-8
 - **H_{4q}**: Los estudiantes manifiestan una percepción de autoeficacia en ‘*coding*’ tras el curso K-8 en un grado de al menos 4 sobre 5 (Mediana ≥ 4)
 - **H_{4r}**: Los estudiantes manifiestan motivación por continuar aprendiendo ‘*coding*’ tras el curso K-8 en un grado de al menos 4 sobre 5 (Mediana ≥ 4)
 - **H_{4s}**: Más del 80% de los estudiantes manifiesta su intención de continuar aprendiendo ‘*coding*’ a través de plataformas-tutoriales en Internet tras el curso K-8
 - **H_{4t}**: Los estudiantes consideran que su grado de aprendizaje con el curso K-8 ha sido de al menos 4 sobre 5 (Mediana ≥ 4)
 - **H_{4u}**: Los estudiantes hacen una valoración global con respecto al curso K-8 en un grado de al menos 4 sobre 5 (Mediana ≥ 4)
- En el marco de la hipótesis directiva de *relevancia*, nos planteamos las siguientes hipótesis específicas:
- **H_{4v}**: Más del 80% de los estudiantes considera importante el ‘*coding*’ para su formación como ciudadano del siglo XXI, tras el curso K-8.

7.1.3. Método

Aplicamos metodologías distintas, aunque complementarias entre sí, en función del tipo de hipótesis específicas sometidas a contraste:

- ✓ Las hipótesis específicas enunciadas en el marco de la directiva de *adecuación* son contrastadas a través de un *diseño cuasi-experimental de dos grupos no equivalentes* (Colás y Buendía, 1992); que incluye medidas psicométricas *pretest* y *postest*, tanto en el grupo experimental (= de tratamiento) como en el grupo de control, habiéndose constituido dichos grupos de manera natural (no aleatoria).
- ✓ Las hipótesis específicas enunciadas en el marco de la directiva de *viabilidad* son contrastadas a través de una metodología *descriptivo-cuantitativa apoyada en encuestas* (preguntas cerradas) e indicadores objetivos de seguimiento (‘*tracking*’) del curso K-8 (Code.org, 2015a).

- ✓ Las hipótesis específicas enunciadas en el marco de la directiva de relevancia son contrastadas a través de una metodología *interpretativo-cualitativa*, apoyada en encuestas (preguntas abiertas)

7.1.3.1. Muestra

El procedimiento de muestreo es intencional y no probabilístico. Todos los centros participantes en la investigación (a excepción de la Escola La Masía) son Institutos de Educación Secundaria (IES) de la Comunidad Valenciana que están integrados en la red de innovación '*IT Teaching*', cuyo objetivo es la utilización del inglés como lengua vehicular de la asignatura optativa de Informática. Tras haber tenido contacto previo con alguno de los IES durante la investigación de 'La Hora del Código', el conjunto de la red se ofreció para la implantación y evaluación del curso K-8; lo cual supone una experiencia de enriquecimiento curricular trimestral en aula ordinaria, sobre codigoalfabetización y desarrollo del pensamiento computacional; que involucra a todos los alumnos de dicha asignatura.

7.1.3.1.1. Muestra de estudiantes

La muestra de estudiantes participantes está formada por un total de 526 sujetos (N = 526); procedentes de 12 centros educativos distintos, todos ellos ubicados en la Comunidad Valenciana (Tabla 7.1), e integrados en la red de innovación '*IT Teaching*' (a excepción de La Masía)

Tabla 7.1. Distribución de la muestra de estudiantes K-8 por centro educativo

	Provincia	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Escola La Masía	Valencia	76	14,4	14,4	14,4
IES María Blasco	Alicante	63	12,0	12,0	26,4
IES Camp de Morvedre	Valencia	59	11,2	11,2	37,6
IES Penyagolosa	Castellón	56	10,6	10,6	48,3
IES San Antonio de Benagéber	Valencia	50	9,5	9,5	57,8
IES Andreu Sempere	Alicante	48	9,1	9,1	66,9
IES L'Elia	Valencia	35	6,7	6,7	73,6
IES Juan de Garay	Valencia	33	6,3	6,3	79,8
IES Dr. Lluís Simarro Lacabra	Valencia	28	5,3	5,3	85,2
IES Betxí	Castellón	28	5,3	5,3	90,5
IES El Plá	Alicante	28	5,3	5,3	95,8
IES L'Almadrava	Alicante	22	4,2	4,2	100,0
Total		526	100,0	100,0	

La distribución de la muestra por sexo es de 62,2% de chicos frente al 37,8% de chicas (Tabla 7.2); todos ellos cursando la asignatura optativa de Informática. Por tanto, el desequilibrio de género es atribuible a una elección diferencial entre ambos sexos de la citada asignatura optativa.

Tabla 7.2. Distribución de la muestra de estudiantes K-8 por sexo

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Chico	326	62,0	62,2	62,2
	Chica	198	37,6	37,8	100,0
	Total	524	99,6	100,0	
Perdidos	Sistema	2	,4		
Total		526	100,0		

Con respecto a la distribución de la muestra por curso académico, se muestra el Tabla 7.3. Tal y como puede observarse, más del 90% de los estudiantes son de 1º Ciclo de la ESO, y dentro del mismo hay aproximadamente el doble de estudiantes de 1º ESO que de 2º ESO.

Tabla 7.3. Distribución de la muestra de estudiantes K-8 por curso académico

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
1º ESO	321	61,0	61,0	61,0
2º ESO	162	30,8	30,8	91,8
3º ESO	29	5,5	5,5	97,3
4º ESO	14	2,7	2,7	100,0
Total	526	100,0	100,0	

Finalmente, de los 526 estudiantes, 401 (76,2%) serán sujetos experimentales de nuestro diseño *cuasi-experimental* (que será descrito en detalle en el [sub-epígrafe 7.1.3.3](#)), mientras que los 125 restantes (23,8%) serán sujetos de control (Tabla 7.4).

Tabla 7.4. Distribución de la muestra K-8 en función de la condición ‘experimental’ o ‘control’ de los sujetos

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Control	125	23,8	23,8	23,8
Experimental	401	76,2	76,2	100,0
Total	526	100,0	100,0	

7.1.3.1.2. Muestra de profesores

El número total de profesores que han colaborado directamente con esta investigación es de 16; todos ellos son profesores de Informática en los 12 centros educativos participantes (Tabla 7.5). De los 16 profesores, 9 son mujeres y 7 son hombres; y la edad media se sitúa en 41,8 años.

Tabla 7.5. Profesores participantes en la evaluación K-8

Centro Educativo	Apellidos	Nombre	Edad	Sexo
Escola La Masía	Alcaide Valverde	Miquel	35	Hombre
	Martínez Martínez	Vicente	39	Hombre
IES María Blasco	Pastor Pina	Herminia	N/A	Mujer
IES Camp de Morvedre	Navarro Bayo	María Amparo	41	Mujer
IES Penyagolosa	Bernad Nebot	Carmen	51	Mujer
	López Sánchez	Joaquín	44	Hombre
IES San Antonio de Benagéber	López de la Hoz	Sandra	35	Mujer
	Stabile Rodríguez-Vergés	Lidia	53	Mujer

Centro Educativo	Apellidos	Nombre	Edad	Sexo
IES Andreu Sempere	Pastor Peidro	Daniel	39	Hombre
IES L'Eliana	Piqueras Murcia	Rocío	47	Mujer
IES Juan de Garay	Díaz Ibáñez	Milagros	40	Mujer
IES Dr. Lluís Simarro Lacabra	Mollá Vayá	Gustavo	44	Hombre
	Pérez Ibáñez	Aránzazu	48	Mujer
IES Betxí	Palasí Lallana	Ana Isabel	34	Mujer
IES El Plá	López Ibarra	David	38	Hombre
IES L'Almadrava	Coloma Brotons	Antonio	40	Hombre

7.1.3.2. Instrumentos

Para el contraste de las hipótesis específicas $H_{4a} \dots H_{4f}$ se han utilizado tests psicométricos estandarizados, que ya fueron descritos en el [sub-epígrafe 6.5.1.2](#) del Capítulo 6. Así:

- La hipótesis específica H_{4a} sobre la aptitud ‘razonamiento lógico’ es contrastada mediante el test PMA-R ([TEA Ediciones, 2015a](#))
- La hipótesis específica H_{4b} sobre la aptitud ‘espacial’ es contrastada mediante el test PMA-E ([TEA Ediciones, 2015a](#))
- La hipótesis específica H_{4c} sobre la aptitud ‘numérica’ es contrastada mediante el test PMA-N ([TEA Ediciones, 2015a](#))
- La hipótesis específica H_{4d} sobre la aptitud ‘verbal’ es contrastada mediante el test PMA-V ([TEA Ediciones, 2015a](#))
- La hipótesis específica H_{4e} sobre la aptitud ‘resolución de problemas’ es contrastada mediante el test RP30 ([TEA Ediciones, 2015b](#))
- La hipótesis específica H_{4f} sobre la aptitud ‘perceptivo-atencional’ es contrastada mediante el test FI-R ([TEA Ediciones, 2015c](#))

Para el contraste de la hipótesis específica H_{4g} sobre la aptitud ‘pensamiento computacional’ ya podemos utilizar nuestro Test de Pensamiento Computacional (TPC) ([Román-González, 2014i](#)); dado su proceso previo de validación expuesto en el Capítulo 6.

Para el contraste de las hipótesis específicas $H_{4h} \dots H_{4j}$ sobre rendimiento académico se utilizan las calificaciones escolares de los estudiantes, que nos proporcionan los profesores colaboradores en la investigación.

Para el contraste de las hipótesis específicas $H_{4k} \dots H_{4o}$ se recogen datos a través del “*Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org*”³⁹⁵ ([Román-González, 2015c](#)). En la siguiente Figura 7.1 se muestra el código QR de enlace a dicho cuestionario; cuyo contenido completo se adjunta en el [Anexo J](#).

³⁹⁵ Disponible en <https://goo.gl/Ovswgg>



Figura 7.1. Código QR de enlace al “Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org” (Román-González, 2015c)

Para el contraste de la hipótesis específica H_{4p} se descargan los datos de seguimiento (*‘tracking’*) de los estudiantes relativos a su desempeño en el curso K-8 (‘niveles superados’ y ‘líneas de código escritas’), que proporciona la plataforma de Code.org.

Finalmente, para el contraste de las hipótesis específicas $H_{4q}...H_{4v}$ se recogen datos a través del “Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org”³⁹⁶ (Román-González, 2015d). En la siguiente Figura 7.2 se muestra el código QR de enlace a dicho cuestionario; cuyo contenido completo se adjunta en el **Anexo K**.



Figura 7.2. Código QR de enlace al “Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org” (Román-González, 2015d)

7.1.3.3. Procedimiento

Algunos de los centros educativos participantes en esta investigación (IES L’Almadrava, IES Penyagolosa e IES San Antonio de Benagéber) ya colaboraron en nuestra recogida de datos sobre el evento ‘La Hora del Código’ en su edición de diciembre de 2013. A raíz de este primer (y muy satisfactorio) contacto, a finales del curso académico 2013/2014 dicho grupo de centros más otros que se incorporaron, todos ellos IES públicos que forman parte de la red de innovación *‘IT Teaching’* de la Comunidad Valenciana, se ofrecieron para realizar una implantación experimental del curso K-8 a lo largo de un trimestre del curso académico siguiente (2014/2015).

Durante el 1º trimestre del curso 2014/2015 se solicitó a estos centros participantes su confirmación y compromiso para colaborar con la investigación, así como el número concreto de grupos y estudiantes que se incorporarían al estudio. Finalmente se convino en contar con grupos de 1º y 2º de la ESO que estuvieran cursando la asignatura optativa de Informática³⁹⁷. Una vez confirmados los grupos participantes, se procedió a asignarles su condición ‘experimental’ o de ‘control’; para ello se respetó la composición natural de los grupos de clase, por lo cual el diseño de investigación llevado a cabo puede etiquetarse como *‘cuasi-experimental de dos grupos no equivalentes’*.

³⁹⁶ Disponible en <https://goo.gl/6X9rev>

³⁹⁷ Posteriormente se incorpora la Escola La Masía, centro privado de Valencia; aportando grupos de 1º, 3º y 4º ESO

Durante el 2º trimestre del curso 2014/2015 se procedió a la implantación experimental del curso K-8. En el marco de la asignatura optativa de Informática (2 sesiones de clase por semana, de 1 hora cada una), los grupos ‘experimentales’ siguieron dicho curso K-8 a lo largo del trimestre, mientras que los grupos ‘control’³⁹⁸ siguieron el currículum TIC tradicional (fundamentalmente, uso de herramientas ofimáticas y navegación web).

Justo al comienzo de dicho 2º trimestre, antes de arrancar el curso K-8, se realizaron las medidas psicométricas en situación *pretest* tanto de los grupos ‘experimentales’ como ‘control’. Típicamente (Tabla 7.6), en cada grupo se tomaron dos medidas *pretest*: nuestro TPC y una de las medidas estandarizadas de TEA Ediciones; siendo aplicadas cada una de estas medidas en una sesión de clase distinta. Análogamente, justo al finalizar el 2º trimestre, una vez realizado el curso K-8, se realizaron las medidas psicométricas en situación *postest* tanto de los grupos ‘experimentales’ como ‘control’. Típicamente, en cada grupo se tomaron como medidas *postest* las mismas que en la situación *pretest* (Tabla 7.6). En la siguiente Figura 7.3 se diagrama el diseño de investigación seguido: diseño *cuasi-experimental* con grupos ‘experimentales’ (G_e) y de ‘control’ (G_c), con medidas *pretest* y *postest*.

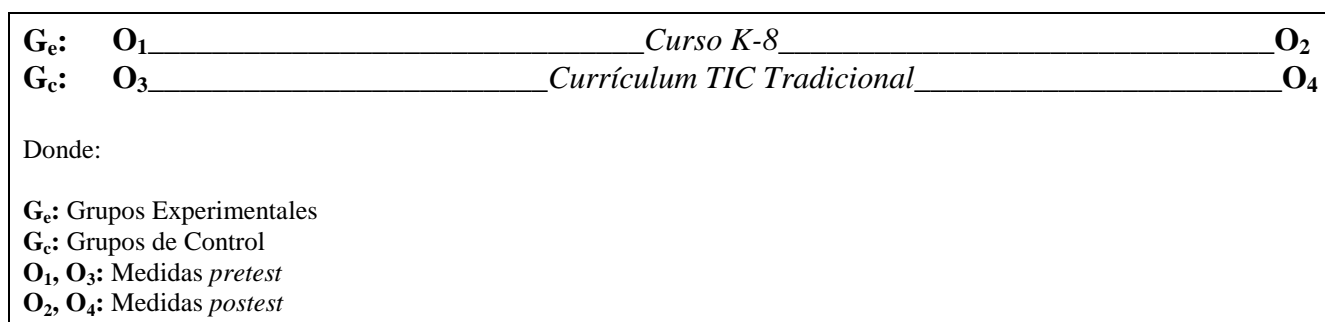


Figura 7.3. Diagrama del diseño *cuasi-experimental* seguido para la evaluación del curso K-8

También al finalizar el 2º trimestre: se descargaron las estadísticas de seguimiento (*‘tracking’*) con el desempeño en el curso K-8 de los estudiantes de los grupos ‘experimentales’; se solicitó a los profesores de los grupos ‘experimentales’ que cumplimentaran el “*Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org*”; y a los estudiantes de los grupos ‘experimentales’ que hicieran lo propio con el “*Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org*”.

Finalmente, al finalizar el curso académico 2014/2015 se solicitaron las calificaciones académicas de todos los sujetos participantes, en las asignaturas de Informática, Matemáticas y Lengua, correspondientes a la ‘1ª evaluación’ (medida *pre* de rendimiento académico) y a la ‘2ª evaluación’ (medida *post* de rendimiento académico). En la Tabla 7.6 se detalla la composición de los distintos grupos ‘experimentales’ (G_e) y de ‘control’ (G_c), así como las medidas tomadas en cada uno de ellos³⁹⁹

³⁹⁸ Los grupos ‘control’ pudieron posteriormente realizar el curso K-8, durante el 3º trimestre del curso 2014/2015, respetándose así el principio ético de equidad en la investigación educativa.

³⁹⁹ Para el análisis global de los datos, se acumulan todos los sujetos de una misma condición (‘control’ o ‘experimental’), como componentes de un mismo grupo de análisis (‘grupo control’ o ‘grupo experimental’). Cuando sea posible, se segmentarán los análisis adicionalmente por centro educativo y/o por ciclo educativo para controlar su posible carácter de variables extrañas. Con ello se pretende garantizar un mínimo de validez interna en el diseño, al tiempo que se mantiene su alta validez externa.

Tabla 7.6. Composición y medidas de los grupos participantes en la investigación

Centro Educativo	Grupo	Condición	Tamaño	Medidas <i>pretest</i>	Medidas <i>postest</i>
IES San Antonio de Benagéber	1º ESO-A	Control	N = 15	PMA, RA*	PMA, RA, TPC
	1º ESO-B	Experimental	N = 16		
	2º ESO		N = 19		
IES Penyagolosa	1º ESO-A	Control	N = 23	PMA, RA, TPC	PMA, RA, TPC
	1º ESO-B	Experimental	N = 18		
	2º ESO		N = 15		
IES L'Almadrava	1º ESO	Experimental	N = 22	PMA, RA, TPC	PMA, RA, TPC
IES Camp de Morvedre	1º ESO	Experimental	N = 33	PMA, TPC	PMA, TPC
	2º ESO-A		N = 13		
	2º ESO-B		N = 13	TPC	TPC
IES Juan de Garay	1º ESO	Experimental	N = 17	PMA, TPC	PMA, TPC
	2º ESO	Control	N = 16		
IES Dr. Lluís Simarro Lacabra	1º ESO-A	Control	N = 13	PMA, TPC	PMA, TPC
	1º ESO-B	Experimental	N = 15		
IES María Blasco	1º ESO	Experimental	N = 19	RA, RP30, TPC	RA, RP30, TPC
	2º ESO		N = 44		
IES Betxí	2º ESO	Control	N = 28	RP30	RP30
IES Andreu Sempere	1º ESO-A	Control	N = 19	FI-R, RA	FI-R, RA
	1º ESO-B	Experimental	N = 15	FI-R, RA, TPC	FI-R, RA, TPC
	2º ESO		N = 14		
IES L'Elia	1º ESO	Experimental	N = 35	FI-R, TPC	FI-R, TPC
IES El Plá	1º ESO	Experimental	N = 28	TPC	TPC
Escola La Masía	1º ESO	Experimental	N = 33	BFQ-NA**, TPC	TPC
	3º ESO-A	Control	N = 11		
	3º ESO-B	Experimental	N = 18		
	4º ESO	Experimental	N = 14		

* RA: Rendimiento Académico

** BFQ-NA: Cuestionario "Big-Five" de Personalidad para Niños y Adolescentes (Esta medida no se ha utilizado para la evaluación del curso K-8)

7.1.4. Resultados

Dando respuesta a la hipótesis específica H_{4a} , en la Tabla 7.7 se detalla la prueba t de diferencia de medias en el PMA-R para grupos independientes entre el grupo control y experimental; tanto es situación *pretest* como *postest*. No se encuentran diferencias significativas en ninguna de las situaciones ($p_{(t)} > 0,05$).

Tabla 7.7. Prueba t para muestras independientes en el PMA-R

	Condición	N	Media	Desviación estándar	t	$p_{(t)}$	d
PMA-R (<i>pre</i>)	Control	57	14,61	6,494	1,269	,206	0,20
	Experimental	118	13,36	5,972			
PMA-R (<i>post</i>)	Control	57	17,14	7,296	,773	,440	0,12
	Experimental	118	16,28	6,700			

En la Tabla 7.8 se detalla la prueba t de diferencia de medias en el PMA-R para grupos relacionados entre la situación *pretest* y *postest*; tanto para la condición control como experimental. En ambas condiciones encontramos ganancias estadísticamente muy significativas entre *pre* y *post* ($p_{(t)} < 0,01$).

Tabla 7.8. Prueba t para muestras relacionadas en el PMA-R

		N	Media	Desviación estándar	r	$p_{(r)}$	t	$p_{(t)}$	d
Control	PMA-R (<i>pre</i>)	57	14,61	6,494	,721**	,000	-3,666**	,001	0,37
	PMA-R (<i>post</i>)	57	17,14	7,296					
Experimental	PMA-R (<i>pre</i>)	118	13,36	5,972	,674**	,000	-6,158**	,000	0,46
	PMA-R (<i>post</i>)	118	16,28	6,700					

** Significación al nivel $p < 0,01$

Finalmente, en la Tabla 7.9 se presentan los resultados del análisis de covarianza (ANCOVA) sobre el *postest* en el PMA-R; tomando la condición ‘control’ o ‘experimental’ como factor fijo, y el *pretest* en el PMA-R como covariable. Tal y como puede observarse, no se encuentran diferencias significativas en el *postest* del PMA-R en función de la condición ‘control’ o ‘experimental’, una vez controladas las posibles diferencias iniciales entre ambos grupos en situación *pretest* ($F_{condición} = 0,020$; $p_{(F)} = 0,887 > 0,05$). El tamaño del efecto global del diseño *cuasi-experimental* sobre el PMA-R es prácticamente nulo ($d = 0,06$). Por tanto, **se rechaza la hipótesis H_{4a}** .

Tabla 7.9. ANCOVA sobre el PMA-R

Variable dependiente: PMA-R (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	F	Sig.	d global
Modelo corregido	3961,136 ^a	2	1980,568	79,223**	,000	
Interceptación	942,761	1	942,761	37,711**	,000	
PMA-R (<i>pre</i>)	3932,664	1	3932,664	157,307**	,000	
Condición	,506	1	,506	,020	,887	0,06
Error	4299,984	172	25,000			
Total	56252,000	175				
Total corregido	8261,120	174				

a. R al cuadrado = ,479 (R al cuadrado ajustada = ,473)

** Significación al nivel $p_{(F)} < 0,01$

Dando respuesta a las hipótesis específicas H_{4b} ... H_{4d} , realizamos unos análisis análogos a los anteriores, sobre el PMA-E (Tablas 7.10, 7.11 y 7.12), PMA-N (7.13, 7.14 y 7.15) y PMA-V (7.16, 7.17 y 7.18).

Tabla 7.10. Prueba t para muestras independientes en el PMA-E

	Condición	N	Media	Desviación estándar	t	$p_{(t)}$	d
PMA-E (<i>pre</i>)	Control	61	16,23	11,097	1,702	,091	0,25
	Experimental	130	13,02	14,178			
PMA-E (<i>post</i>)	Control	61	24,02	15,244	1,119	,265	0,18
	Experimental	130	21,30	15,823			

Tabla 7.11. Prueba t para muestras relacionadas en el PMA-E

		N	Media	Desviación estándar	r	$p_{(r)}$	t	$p_{(t)}$	d
Control	PMA-E (<i>pre</i>)	61	16,23	11,097	,808**	,000	-6,704**	,000	0,58
	PMA-E (<i>post</i>)	61	24,02	15,244					
Experimental	PMA-E (<i>pre</i>)	130	13,02	14,178	,656**	,000	-7,534**	,000	0,55
	PMA-E (<i>post</i>)	130	21,30	15,823					

** Significación al nivel $p < 0,01$

Tabla 7.12. ANCOVA sobre el PMA-E

Variable dependiente: PMA-E (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	F	Sig.	d global
Modelo corregido	22464,521 ^a	2	11232,261	87,686**	,000	
Interceptación	9263,178	1	9263,178	72,314**	,000	
PMA-E (<i>pre</i>)	22158,166	1	22158,166	172,980**	,000	
Condición	,373	1	,373	,003	,957	0,04
Error	24082,117	188	128,096			
Total	140404,000	191				
Total corregido	46546,639	190				

a. R al cuadrado = ,483 (R al cuadrado ajustada = ,477)

** Significación al nivel $p_{(F)} < 0,01$

Tal y como puede observarse en las tablas previas, los resultados en las pruebas t para el PMA-E son análogos a los del PMA-R. En cuanto al ANCOVA, tampoco se encuentran diferencias significativas en el *posttest* del PMA-E en función de la condición ‘control’ o ‘experimental’, una vez controladas las posibles diferencias iniciales entre ambos grupos en situación *pretest* ($F_{condición} = 0,003$; $p_{(F)} = 0,957 > 0,05$). El tamaño del efecto global del diseño *cuasi-experimental* sobre el PMA-E es prácticamente nulo ($d = 0,04$). Por tanto, **se rechaza la hipótesis H_{4b}** .

Tabla 7.13. Prueba t para muestras independientes en el PMA-N

	Condición	N	Media	Desviación estándar	t	$p_{(t)}$	d
PMA-N (<i>pre</i>)	Control	51	12,73	7,153	-1,244	,215	0,21
	Experimental	126	14,36	8,182			
PMA-N (<i>post</i>)	Control	51	16,73	8,464	-1,199	,232	0,20
	Experimental	126	18,49	9,039			

Tabla 7.14. Prueba *t* para muestras relacionadas en el PMA-N

		N	Media	Desviación estándar	<i>r</i>	<i>p</i> (<i>r</i>)	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
Control	PMA-N (<i>pre</i>)	51	12,73	7,153	,632**	,000	-4,199**	,000	0,51
	PMA-N (<i>post</i>)	51	16,73	8,464					
Experimental	PMA-N (<i>pre</i>)	126	14,36	8,182	,324**	,000	-4,624**	,000	0,48
	PMA-N (<i>post</i>)	126	18,49	9,039					

** Significación al nivel $p < 0,01$

Tabla 7.15. ANCOVA sobre el PMA-N

Variable dependiente: PMA-N (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	<i>F</i>	Sig.	<i>d</i> global
Modelo corregido	2317,786 ^a	2	1158,893	17,397**	,000	
Interceptación	5613,767	1	5613,767	84,271**	,000	
PMA-N (<i>pre</i>)	2204,486	1	2204,486	33,092**	,000	
Condición	38,453	1	38,453	,577	,448	0,02
Error	11591,163	174	66,616			
Total	71149,000	177				
Total corregido	13908,949	176				

a. R al cuadrado = ,167 (R al cuadrado ajustada = ,157)

** Significación al nivel $p_{(F)} < 0,01$

Tal y como puede observarse en las tablas previas, los resultados en las pruebas *t* para el PMA-N son análogos a los del PMA-R y PMA-E: no se encuentran diferencias significativas entre el grupo control y experimental, ni en situación *pretest* ni *posttest* ($p_{(t)} > 0,05$); pero sí se encuentran ganancias muy significativas entre *pre* y *post*, tanto para el grupo control como para el experimental ($p_{(t)} < 0,01$). En cuanto al ANCOVA, tampoco se encuentran diferencias significativas en el *posttest* del PMA-N en función de la condición ‘control’ o ‘experimental’, una vez controladas las posibles diferencias iniciales entre ambos grupos en situación *pretest* ($F_{condición} = 0,577$; $p_{(F)} = 0,448 > 0,05$). El tamaño del efecto global del diseño *cuasi-experimental* sobre el PMA-N es prácticamente nulo ($d = 0,02$). Por tanto, **se rechaza la hipótesis H_{4c}**.

Tabla 7.16. Prueba *t* para muestras independientes en el PMA-V

	Condición	N	Media	Desviación estándar	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
PMA-V (<i>pre</i>)	Control	57	24,56	6,167	-,267	,790	0,04
	Experimental	130	24,83	6,423			
PMA-V (<i>post</i>)	Control	57	26,70	7,106	-,116	,908	0,02
	Experimental	130	26,83	6,969			

Tabla 7.17. Prueba *t* para muestras relacionadas en el PMA-V

		N	Media	Desviación estándar	<i>r</i>	<i>p</i> (<i>r</i>)	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
Control	PMA-V (<i>pre</i>)	57	24,56	6,167	,719**	,000	-3,197**	,002	0,32
	PMA-V (<i>post</i>)	57	26,70	7,106					
Experimental	PMA-V (<i>pre</i>)	130	24,83	6,423	,527**	,000	-3,491**	,001	0,30
	PMA-V (<i>post</i>)	130	26,83	6,969					

** Significación al nivel $p < 0,01$

Tabla 7.18. ANCOVA sobre el PMA-V

Variable dependiente: PMA-V (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	F	Sig.	d global
Modelo corregido	3098,531 ^a	2	1549,265	47,556**	,000	
Interceptación	1333,932	1	1333,932	40,946**	,000	
PMA-V (<i>pre</i>)	3097,871	1	3097,871	95,091**	,000	
Condición	,079	1	,079	,002	,961	0,02
Error	5994,336	184	32,578			
Total	143318,000	187				
Total corregido	9092,866	186				

a. R al cuadrado = ,341 (R al cuadrado ajustada = ,334)

** Significación al nivel $p_{(F)} < 0,01$

Tal y como puede observarse en las tablas anteriores, los resultados en el PMA-V siguen en la misma línea que los del resto de pruebas de la batería PMA. Las pruebas *t* siguen arrojando el mismo patrón de resultados; y en el ANCOVA tampoco se encuentran diferencias significativas en el *postest* del PMA-V ($F = 0,002$; $p_{(F)} = 0,961 > 0,05$). El tamaño del efecto global del diseño *cuasi-experimental* sobre el PMA-V es prácticamente nulo ($d = 0,02$). Por tanto, **se rechaza la hipótesis H_{4d}** .

Recapitulando los resultados relativos al grupo de hipótesis específicas $H_{4a} \dots H_{4d}$, en las siguientes Figuras 7.4, 7.5, 7.6 y 7.7 se ilustran dichos resultados de manera resumida, respectivamente para cada una de las pruebas que componen la batería PMA. En estas figuras sólo se muestran los tamaños del efecto relativos a las pruebas *t* que resultaron estadísticamente significativas; y el tamaño del efecto global correspondiente al diseño *cuasi-experimental* en su conjunto.

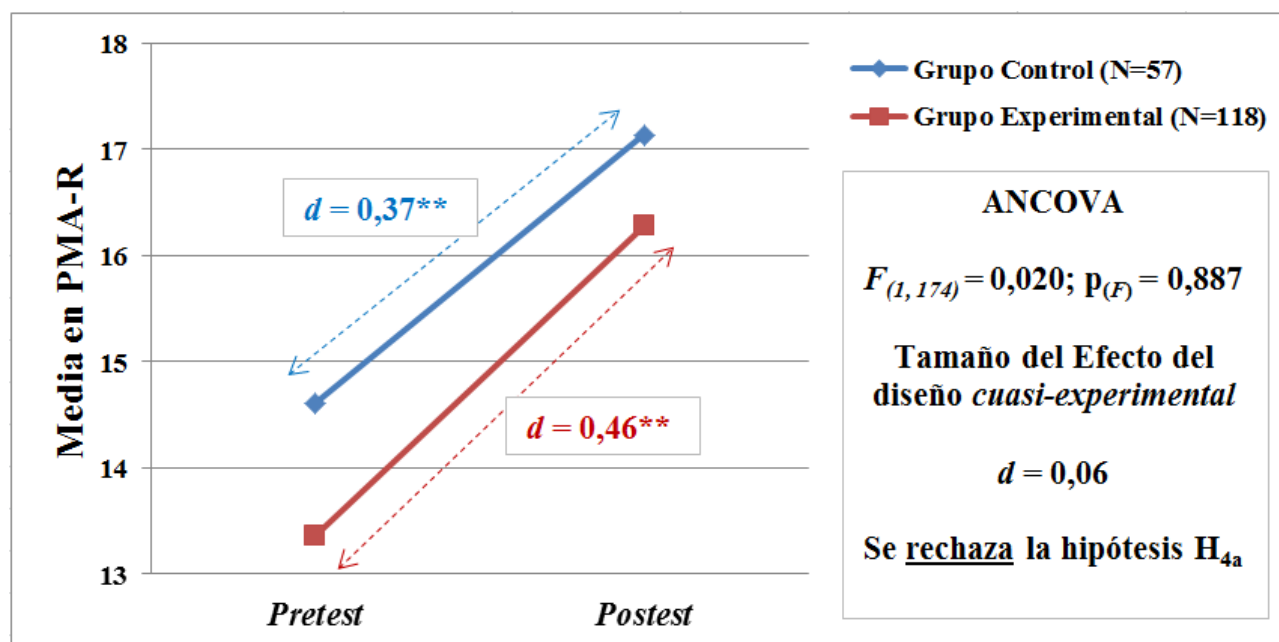


Figura 7.4. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘razonamiento lógico’ (PMA-R). A la izquierda, se muestran las medias en el PMA-R del grupo control y experimental, en situación *pretest* y *postest*; marcándose exclusivamente los tamaños del efecto correspondientes a las pruebas *t* estadísticamente significativas. A la derecha, se muestra un resumen del análisis de covarianza (ANCOVA), el tamaño del efecto global para el diseño *cuasi-experimental*, y la decisión estadística subsiguiente.

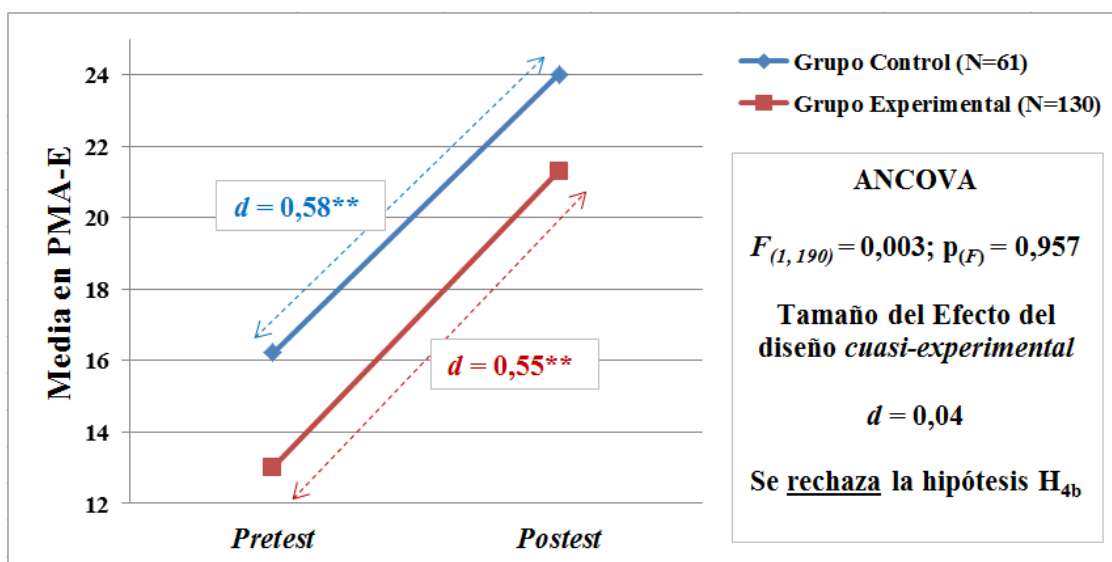


Figura 7.5. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘espacial’ (PMA-E)

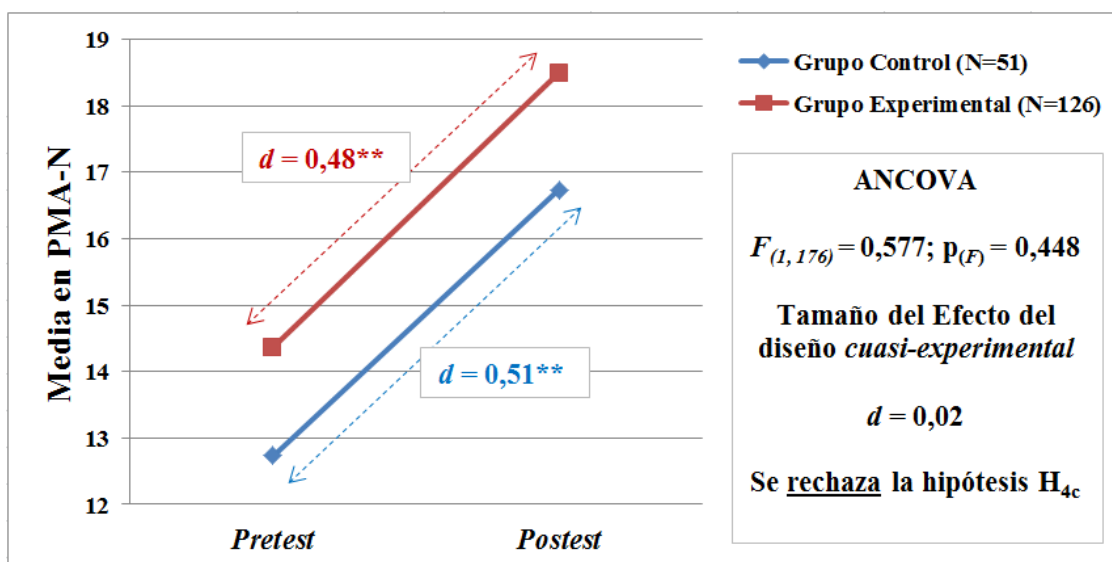


Figura 7.6. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘numérica’ (PMA-N)

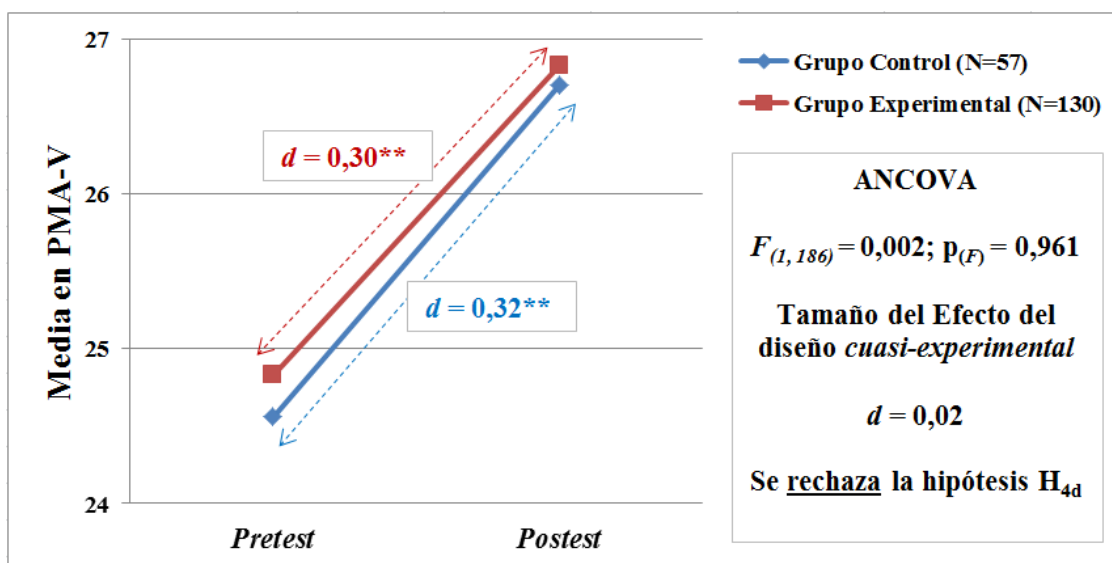


Figura 7.7. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘verbal’ (PMA-V)

Para finalizar con el PMA, si recalculamos los anteriores ANCOVA segmentados en función de los cuatro IES que en el diseño aportan grupos ‘control’ y ‘experimental’ con *pre* y *post* en dicha batería (En [Tabla 7.6](#), IES San Antonio de Benagéber, IES Penyagolosa, IES Juan de Garay y IES Dr. Lluís Simarro Lacabra), obtenemos que: no se encuentra en ninguno de los centros ninguna diferencia significativa para ninguna de las pruebas de la batería PMA (todas las $p_{(F)} > 0,05$); a excepción del ANCOVA sobre el *postest* del PMA-R en el IES San Antonio de Benagéber que arrojó una diferencia significativa a favor del grupo experimental ($F_{(1,41)} = 4,434$; $p_{(F)} = 0,042 < 0,05$).

Dando ahora respuesta a la hipótesis específica H_{4e} sobre la aptitud ‘resolución de problemas’, medida a través del RP30; en la [Tabla 7.19](#) se detalla la prueba *t* de diferencia de medias en el RP30 para grupos independientes entre el grupo control y experimental; tanto es situación *pretest* como *postest*. No se encuentran diferencias significativas en ninguna de las situaciones ($p_{(t)} > 0,05$).

Tabla 7.19. Prueba *t* para muestras independientes en el RP30

	Condición	N	Media	Desviación estándar	<i>t</i>	$p_{(t)}$	<i>d</i>
RP30 (<i>pre</i>)	Control	21	29,19	15,102	1,563	,122	0,42
	Experimental	55	21,91	19,172			
RP30 (<i>post</i>)	Control	21	31,24	13,397	,297	,767	0,07
	Experimental	55	30,05	20,058			

En la [Tabla 7.20](#) se detalla la prueba *t* de diferencia de medias en el RP30 para grupos relacionados entre la situación *pretest* y *postest*; tanto para la condición control como experimental. Sólo encontramos una ganancia estadísticamente muy significativas entre *pre* y *post* ($p_{(t)} < 0,01$) en el grupo experimental.

Tabla 7.20. Prueba *t* para muestras relacionadas en el RP30

		N	Media	Desviación estándar	<i>r</i>	$p_{(r)}$	<i>t</i>	$p_{(t)}$	<i>d</i>
Control	RP30 (<i>pre</i>)	21	29,19	15,102	,460*	,036	-,630	,536	0,14
	RP30 (<i>post</i>)	21	31,24	13,397					
Experimental	RP30 (<i>pre</i>)	55	21,91	19,172	,855**	,000	-5,694**	,000	0,41
	RP30 (<i>post</i>)	55	30,05	20,058					

* Significación al nivel $p < 0,05$

** Significación al nivel $p < 0,01$

Finalmente, en la [Tabla 7.21](#) se presentan los resultados del análisis de covarianza (ANCOVA) sobre el *postest* en el RP30; tomando la condición ‘control’ o ‘experimental’ como factor fijo, y el *pretest* en el RP30 como covariable. Tal y como puede observarse, no se encuentran diferencias significativas en el *postest* del RP30 en función de la condición ‘control’ o ‘experimental’, una vez controladas las posibles diferencias iniciales entre ambos grupos en situación *pretest* ($F_{condición} = 2,444$; $p_{(F)} = 0,122 > 0,05$). El tamaño del efecto global del diseño *cuasi-experimental* sobre el RP30 es $d = 0,34$, que puede considerarse un tamaño entre ‘pequeño’ y ‘moderado’, y ya a tomar en consideración. Por tanto, aunque se rechaza la hipótesis H_{4e} , es plausible considerar que con muestras de tamaño algo mayor ésta podría haberse aceptado. Se ilustra en la [Figura 7.8](#).

Tabla 7.21. ANCOVA sobre el RP30

Variable dependiente: RP30 (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	F	Sig.	d global
Modelo corregido	15772,605 ^a	2	7886,302	60,199	,000**	
Interceptación	2372,125	1	2372,125	18,107	,000**	
RP30 (<i>pre</i>)	15751,317	1	15751,317	120,235	,000**	
Condición	320,239	1	320,239	2,444	,122	0,34
Error	9563,329	73	131,005			
Total	95487,000	76				
Total corregido	25335,934	75				

a. R al cuadrado = ,623 (R al cuadrado ajustada = ,612)

** Significación al nivel $p_{(F)} < 0,01$

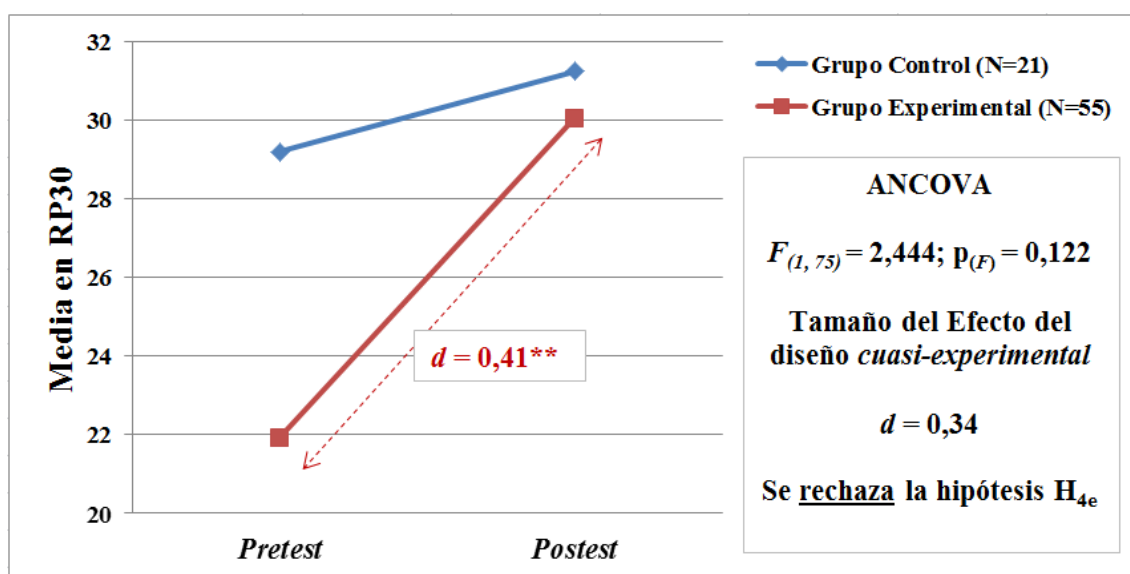


Figura 7.8. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘resolución de problemas’ (RP30)

Dando respuesta a la hipótesis específica H_{4f} sobre la aptitud ‘perceptivo-atencional’ medida a través del FI-R, realizamos unos análisis análogos a los anteriores sobre cada uno de los índices que arroja este instrumento: el Índice ‘A-E’ de Percepción y Atención (Tablas 7.22, 7.23 y 7.24), y el Índice ‘ICI’ de Control de la Impulsividad (Tablas 7.25, 7.26 y 7.27).

Tabla 7.22. Prueba *t* para muestras independientes en el Índice A-E del FI-R

	Condición	N	Media	Desviación estándar	t	p _(t)	d
A-E (<i>pre</i>)	Control	17	29,53	8,186	-,551	,583	0,15
	Experimental	54	30,72	7,659			
A-E (<i>post</i>)	Control	17	36,12	9,656	,099	,921	0,03
	Experimental	54	35,89	7,813			

Tabla 7.23. Prueba *t* para muestras relacionadas en el Índice A-E del FI-R

		N	Media	Desviación estándar	r	p _(r)	t	p _(t)	d
Control	A-E (<i>pre</i>)	17	29,53	8,186	,704**	,002	-3,885**	,001	0,74
	A-E (<i>post</i>)	17	36,12	9,656					
Experimental	A-E (<i>pre</i>)	54	30,72	7,659	,698**	,000	-6,309**	,000	0,67
	A-E (<i>post</i>)	54	35,89	7,813					

** Significación al nivel $p < 0,01$

Tabla 7.24. ANCOVA sobre el Índice A-E del FI-R

Variable dependiente: A-E (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	F	Sig.	d global
Modelo corregido	2303,990 ^a	2	1151,995	32,320**	,000	
Interceptación	787,387	1	787,387	22,090**	,000	
A-E (<i>pre</i>)	2303,314	1	2303,314	64,620**	,000	
Condición	15,978	1	15,978	,448	,505	0,18
Error	2423,784	68	35,644			
Total	96456,000	71				
Total corregido	4727,775	70				

a. R al cuadrado = ,487 (R al cuadrado ajustada = ,472)

** Significación al nivel $p_{(F)} < 0,01$

Tabla 7.25. Prueba *t* para muestras independientes en el Índice ICI del FI-R

	Condición	N	Media	Desviación estándar	t	p _(t)	d
ICI (<i>pre</i>)	Control	17	89,71	8,865	-,775	,441	0,21
	Experimental	54	91,50	8,149			
ICI (<i>post</i>)	Control	17	87,94	12,311	-1,425	,159	0,36
	Experimental	54	91,72	8,531			

Tabla 7.26. Prueba *t* para muestras relacionadas en el Índice ICI del FI-R

		N	Media	Desviación estándar	r	p _(r)	t	p _(t)	d
Control	ICI (<i>pre</i>)	17	89,71	8,865	,271	,292	,557	,586	0,16
	ICI (<i>post</i>)	17	87,94	12,311					
Experimental	ICI (<i>pre</i>)	54	91,50	8,149	,313*	,021	-,167	,868	0,03
	ICI (<i>post</i>)	54	91,72	8,531					

* Significación al nivel $p < 0,05$

Tabla 7.27. ANCOVA sobre el Índice ICI del FI-R

Variable dependiente: ICI (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	F	Sig.	d global
Modelo corregido	739,123 ^a	2	369,561	4,388*	,016	
Interceptación	2001,100	1	2001,100	23,758**	,000	
ICI (<i>pre</i>)	554,277	1	554,277	6,581*	,013	
Condición	128,799	1	128,799	1,529	,220	0,24
Error	5727,497	68	84,228			
Total	592054,000	71				
Total corregido	6466,620	70				

a. R al cuadrado = ,114 (R al cuadrado ajustada = ,088)

* Significación al nivel $p_{(F)} < 0,05$ ** Significación al nivel $p_{(F)} < 0,01$

Tal y como puede verse en las tablas anteriores, el patrón de resultados para el índice ‘A-E’ del FI-R es análogo al encontrado para todas las pruebas de la batería PMA: no hay diferencias significativas entre el grupo control y experimental, ni en situación *pretest* ni *postest*; pero sí hay diferencias muy significativas entre el *pretest* y el *postest*, tanto en el grupo control como en el experimental (Figura 7.9). En el caso del índice ‘ICI’ no se encuentra ninguna *t* significativa (Figura 7.10). En cualquier caso, tanto el ANCOVA sobre el *postest* del ‘A-E’ como el ANCOVA sobre el *postest* del ‘ICI’ arrojan una *F* no estadísticamente significativa ($p_{(F)} > 0,05$); con tamaños globales del efecto pequeños (0,18 y 0,24, respectivamente). Por tanto, **se rechaza la hipótesis H_{4f}**

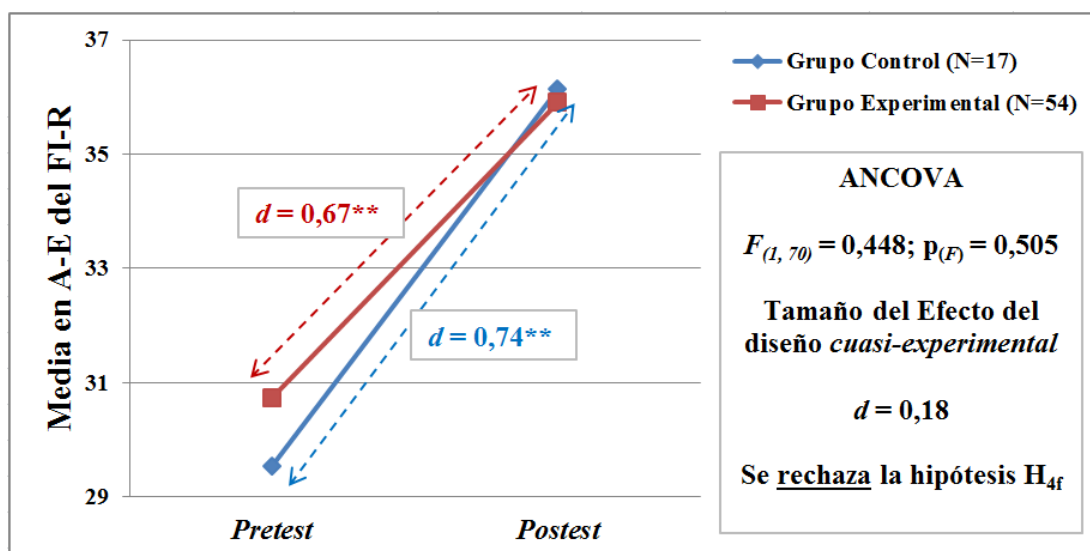


Figura 7.9. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘perceptivo-atencional’ (A-E)

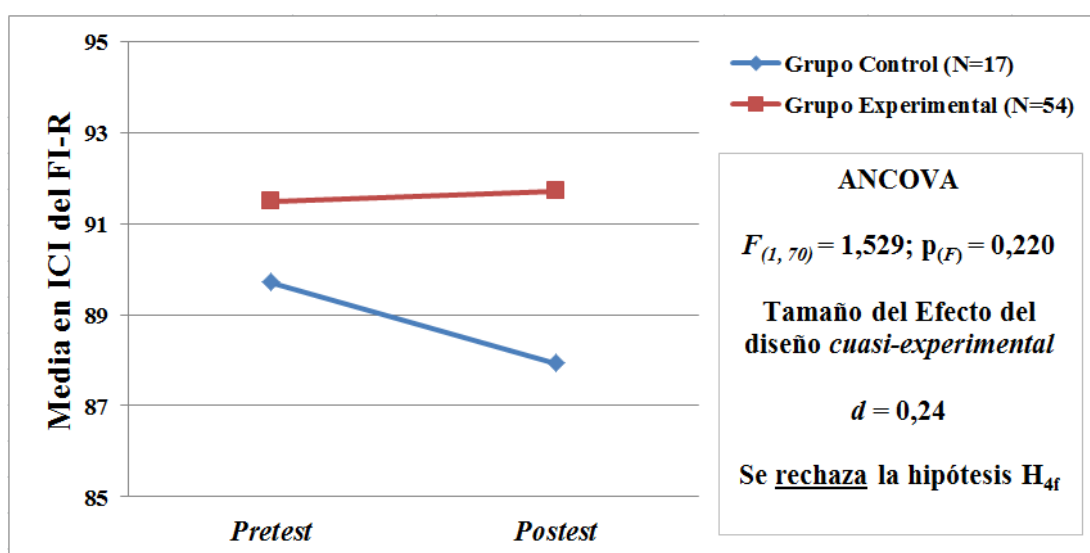


Figura 7.10. Resumen de los resultados del diseño *cuasi-experimental* sobre la aptitud ‘perceptivo-atencional’ (ICI)

Para finalizar con el FI-R, si replicamos los ANCOVA focalizados sobre el único IES que aporta simultáneamente grupos ‘control’ y ‘experimental’ en esta medida (ver [Tabla 7.6](#), IES Andreu Sempere); seguimos sin encontrar diferencias ni en el índice ‘A-E’, ni en el índice ‘ICI’ (en ambos casos $p_{(F)} > 0,05$)

Dando respuesta ahora a la hipótesis específica H_{4g} sobre la aptitud ‘pensamiento computacional’, se presentan los resultados correspondientes en las Tablas 7.28, 7.29 y 7.30.

Tabla 7.28. Prueba *t* para muestras independientes en el TPC

	Condición	N	Media	Desviación estándar	<i>t</i>	$p_{(t)}$	<i>d</i>
TPC (<i>pre</i>)	Control	54	16,67	3,629	,432	,666	0,07
	Experimental	332	16,39	4,391			
TPC (<i>post</i>)	Control	54	17,91	4,602	-3,571**	,000	0,52
	Experimental	332	20,30	4,562			

** Significación al nivel $p < 0,01$

Tabla 7.29. Prueba *t* para muestras relacionadas en el TPC

		N	Media	Desviación estándar	<i>r</i>	<i>p</i> (<i>r</i>)	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
Control	TPC (<i>pre</i>)	54	16,67	3,629	,704** ⁴⁰⁰	,000	-2,771**	,008	0,30
	TPC (<i>post</i>)	54	17,91	4,602					
Experimental	TPC (<i>pre</i>)	332	16,39	4,391	,637**	,000	-18,649**	,000	0,87
	TPC (<i>post</i>)	332	20,30	4,562					

** Significación al nivel $p < 0,01$

Tabla 7.30. ANCOVA sobre el TPC

Variable dependiente: TPC (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	<i>F</i>	Sig.	<i>d</i> global
Modelo corregido	3586,378 ^a	2	1793,189	146,369**	,000	
Interceptación	1378,328	1	1378,328	112,506**	,000	
TPC (<i>pre</i>)	3320,232	1	3320,232	271,014**	,000	
Condición	309,036	1	309,036	25,225**	,000	0,62
Error	4692,184	383	12,251			
Total	162159,000	386				
Total corregido	8278,562	385				

a. R al cuadrado = ,433 (R al cuadrado ajustada = ,430)

** Significación al nivel $p_{(F)} < 0,01$

Tal y como puede observarse en las tablas anteriores, no se encuentran diferencias significativas entre ambos grupos en la situación *pretest*, pero sí (muy significativas) en la situación *posttest*. Por otro lado, tanto el grupo control como el experimental mejoran su rendimiento en el TPC de manera muy significativa entre el *pretest* y el *posttest*, si bien dicha ganancia es mayor en el caso del grupo experimental. Globalmente, el ANCOVA sobre el *posttest* del TPC en función de la condición ‘control’ o ‘experimental’, y controlando las diferencias iniciales en *pretest*, arroja una *F* muy significativa ($F = 25,225$; $p_{(F)} = 0,000 < 0,01$); con un tamaño del efecto del diseño *cuasi-experimental* sobre el TPC de valor $d = 0,62$, que puede considerarse un efecto entre ‘moderado’ y ‘grande’. Así pues, **se acepta la hipótesis H_{4g}** de que el curso K-8, en comparación con el currículum TIC tradicional, mejora la aptitud ‘pensamiento computacional’ (Figura 7.11)

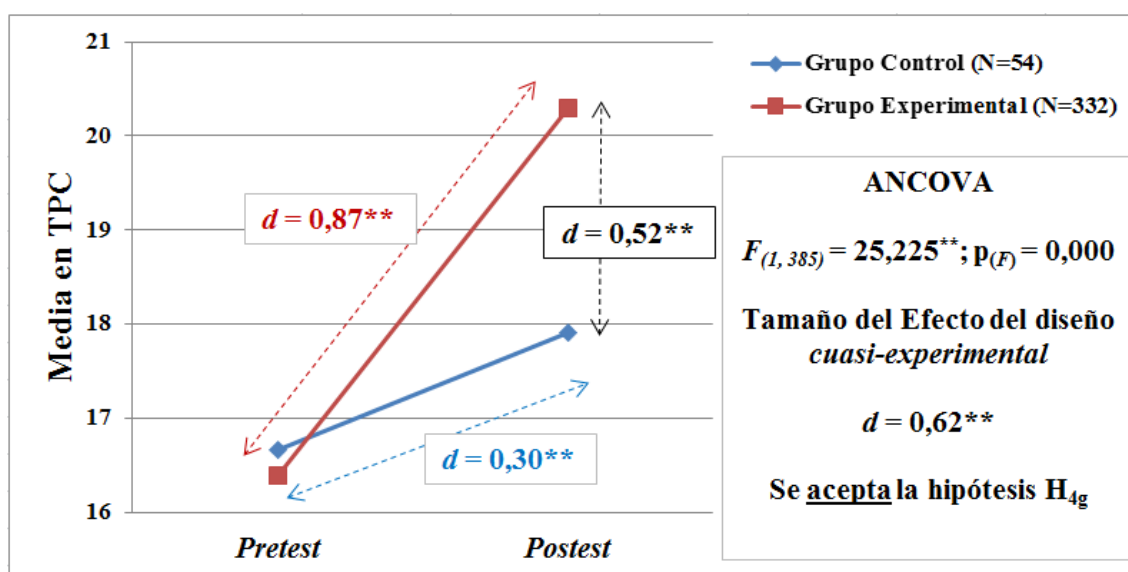


Figura 7.11. Resumen de los resultados del diseño *cuasi-experimental* sobre ‘pensamiento computacional’ (TPC)

⁴⁰⁰ Éste es el valor de fiabilidad como estabilidad del TPC, que ya aportamos en el Capítulo 6 (sub-epígrafe 6.5.2.2.2)

Para finalizar con el TPC, tal y como puede verse en la [Tabla 7.6](#), es la única medida del diseño que se aplica *pre* y *post*, en grupos ‘control’ y ‘experimental’, tanto en primer ciclo de la ESO (todos los IES públicos) como en segundo ciclo de la ESO (Escuela La Masía, privada). Para controlar la amenaza del ‘ciclo educativo’ como variable extraña en el diseño, replicamos el ANCOVA sobre el *postest* del TPC, segmentado por ciclo. Como resultado, obtenemos diferencias significativas a favor del grupo experimental tanto en primer ciclo ($F_{(1,347)} = 9,180$; $p_{(F)} = 0,003 < 0,01$), como en segundo ciclo ($F_{(1,37)} = 72,142$; $p_{(F)} = 0,000 < 0,01$) de la ESO.

Con respecto a las hipótesis específicas de rendimiento académico (RA), $H_{4h} \dots H_{4j}$, relativas respectivamente a las asignaturas de Informática (Inf), Matemáticas (Mat) y Lengua (Len), se presentan los resultados en las siguientes Tablas 7.31 a 7.39.

Tabla 7.31. Prueba *t* para muestras independientes en RA-Inf

	Condición	N	Media	Desviación estándar	<i>t</i>	$p_{(t)}$	<i>d</i>
RA-Inf (<i>pre</i>)	Control	65	7,25	1,531	1,154	,250	0,17
	Experimental	179	6,97	1,722			
RA-Inf (<i>post</i>)	Control	65	7,35	1,690	1,178	,240	0,17
	Experimental	179	7,07	1,678			

Tabla 7.32. Prueba *t* para muestras relacionadas en RA-Inf

		N	Media	Desviación estándar	<i>r</i>	$p_{(r)}$	<i>t</i>	$p_{(t)}$	<i>d</i>
Control	RA-Inf (<i>pre</i>)	65	7,25	1,531	,304*	,014	-,456	,650	0,06
	RA-Inf (<i>post</i>)	65	7,35	1,690					
Experimental	RA-Inf (<i>pre</i>)	179	6,97	1,722	,683**	,000	-,994	,322	0,06
	RA-Inf (<i>post</i>)	179	7,07	1,678					

* Significación al nivel $p < 0,05$

** Significación al nivel $p < 0,01$

Tabla 7.33. ANCOVA sobre el RA-Inf

Variable dependiente: RA-Inf (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	<i>F</i>	Sig.	<i>d</i> global
Modelo corregido	242,018 ^a	2	121,009	65,394	,000**	
Interceptación	112,775	1	112,775	60,944	,000**	
RA-Inf (<i>pre</i>)	238,096	1	238,096	128,668	,000**	
Condición	,695	1	,695	,376	,541	0,00
Error	445,961	241	1,850			
Total	13139,000	244				
Total corregido	687,980	243				

a. R al cuadrado = ,352 (R al cuadrado ajustada = ,346)

** Significación al nivel $p_{(F)} < 0,01$

Tabla 7.34. Prueba *t* para muestras independientes en RA-Mat

	Condición	N	Media	Desviación estándar	<i>t</i>	$p_{(t)}$	<i>d</i>
RA-Mat (<i>pre</i>)	Control	84	5,43	2,202	-2,640**	,009	0,36
	Experimental	179	6,16	1,808			
RA-Mat (<i>post</i>)	Control	84	5,31	2,173	-2,013*	,045	0,26
	Experimental	179	5,84	1,891			

* Significación al nivel $p < 0,05$

** Significación al nivel $p < 0,01$

Tabla 7.35. Prueba *t* para muestras relacionadas en RA-Mat

		N	Media	Desviación estándar	<i>r</i>	<i>p</i> (<i>r</i>)	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
Control	RA-Mat (<i>pre</i>)	84	5,43	2,202	,840**	,000	,883	,380	0,05
	RA-Mat (<i>post</i>)	84	5,31	2,173					
Experimental	RA-Mat (<i>pre</i>)	179	6,16	1,808	,746**	,000	3,224**	,002	0,17
	RA-Mat (<i>post</i>)	179	5,84	1,891					

** Significación al nivel $p < 0,01$

Tabla 7.36. ANCOVA sobre el RA-Mat

Variable dependiente: RA-Mat (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	<i>F</i>	Sig.	<i>d</i> global
Modelo corregido	645,913 ^a	2	322,957	210,814	,000**	
Interceptación	22,948	1	22,948	14,980	,000**	
RA-Mat (<i>pre</i>)	629,947	1	629,947	411,206	,000**	
Condición	,161	1	,161	,105	,746	0,10
Error	398,307	260	1,532			
Total	9497,000	263				
Total corregido	1044,221	262				

a. R al cuadrado = ,619 (R al cuadrado ajustada = ,616)

** Significación al nivel $p_{(F)} < 0,01$

Tabla 7.37. Prueba *t* para muestras independientes en RA-Len

	Condición	N	Media	Desviación estándar	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
RA-Len (<i>pre</i>)	Control	84	5,50	1,898	-1,466	,144	0,19
	Experimental	179	5,84	1,666			
RA-Len (<i>post</i>)	Control	84	5,49	1,794	-1,059	,290	0,14
	Experimental	179	5,73	1,714			

Tabla 7.38. Prueba *t* para muestras relacionadas en RA-Len

		N	Media	Desviación estándar	<i>r</i>	<i>p</i> (<i>r</i>)	<i>t</i>	<i>p</i> (<i>t</i>)	<i>d</i>
Control	RA-Len (<i>pre</i>)	84	5,50	1,898	,848**	,000	,107	,915	0,01
	RA-Len (<i>post</i>)	84	5,49	1,794					
Experimental	RA-Len (<i>pre</i>)	179	5,84	1,666	,730**	,000	1,143	,255	0,07
	RA-Len (<i>post</i>)	179	5,73	1,714					

** Significación al nivel $p < 0,01$

Tabla 7.39. ANCOVA sobre el RA-Len

Variable dependiente: RA-Len (*post*)

Origen	Tipo III de suma de cuadrados	gl	Cuadrático promedio	<i>F</i>	Sig.	<i>d</i> global
Modelo corregido	473,562 ^a	2	236,781	192,414**	,000	
Interceptación	34,573	1	34,573	28,094**	,000	
RA-Len (<i>pre</i>)	470,166	1	470,166	382,068**	,000	
Condición	,015	1	,015	,012	,911	0,06
Error	319,951	260	1,231			
Total	9201,000	263				
Total corregido	793,513	262				

a. R al cuadrado = ,597 (R al cuadrado ajustada = ,594)

** Significación al nivel $p_{(F)} < 0,01$

Tal y como puede observarse en los ANCOVAs anteriores sobre el rendimiento académico en Informática, Matemáticas y Lengua; en ninguna de dichas asignaturas se encuentran diferencias significativas en el *postest* de rendimiento en función de la condición ‘control’ o ‘experimental’, habiéndose controlado las posibles diferencias en situación *pretest* (en todos los casos $p_{(F)} > 0,05$). Todos los tamaños globales del efecto del diseño *cuasi-experimental* sobre el rendimiento académico en las distintas asignaturas son casi nulos ($d \leq 0,10$). Con respecto a las pruebas *t*, en Informática y en Lengua no se encuentra ninguna diferencia entre pares de medias que sea significativa (todas $p_{(t)} > 0,05$). En el caso de la asignatura de Matemáticas, hay una diferencia muy significativa del grupo experimental frente al control en situación *pretest*, que pasa a ser sólo significativa en situación *postest*; debido a un descenso muy significativo (aunque de tamaño pequeño, $d = 0,17$) entre el *pre* y *post* del grupo experimental. Todo ello se ilustra en las siguientes Figuras 7.12, 7.13 y 7.14.

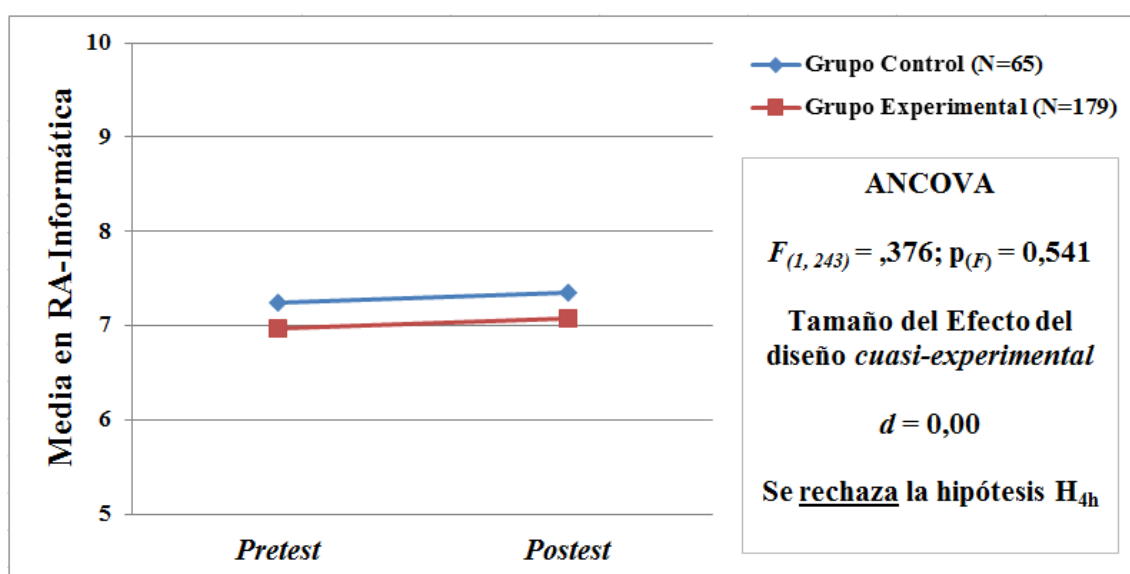


Figura 7.12. Resumen de los resultados del diseño *cuasi-experimental* sobre el rendimiento académico en Informática

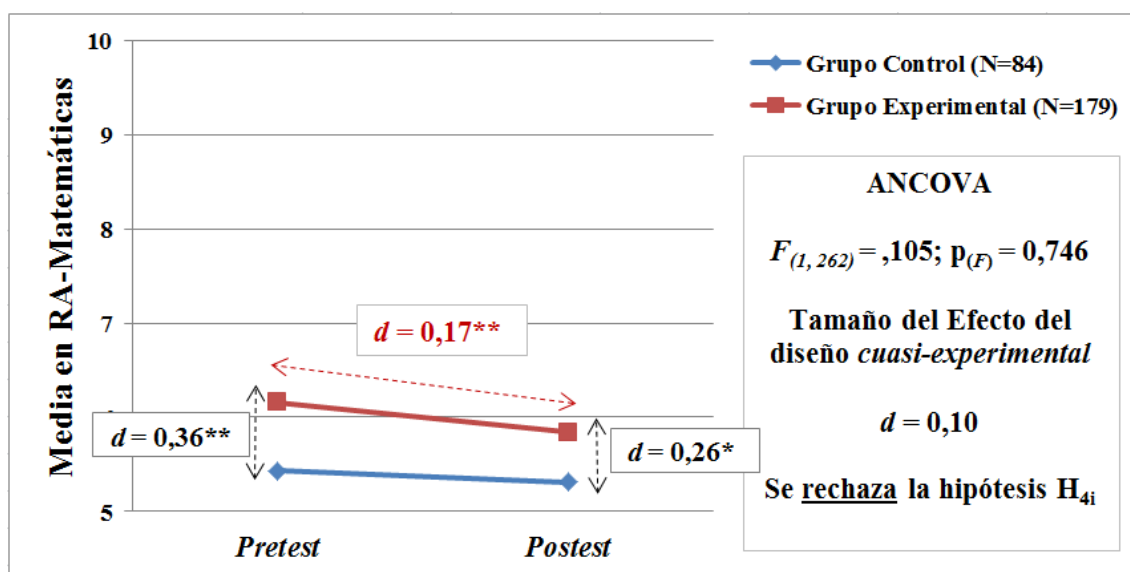


Figura 7.13. Resumen de los resultados del diseño *cuasi-experimental* sobre el rendimiento académico en Matemáticas

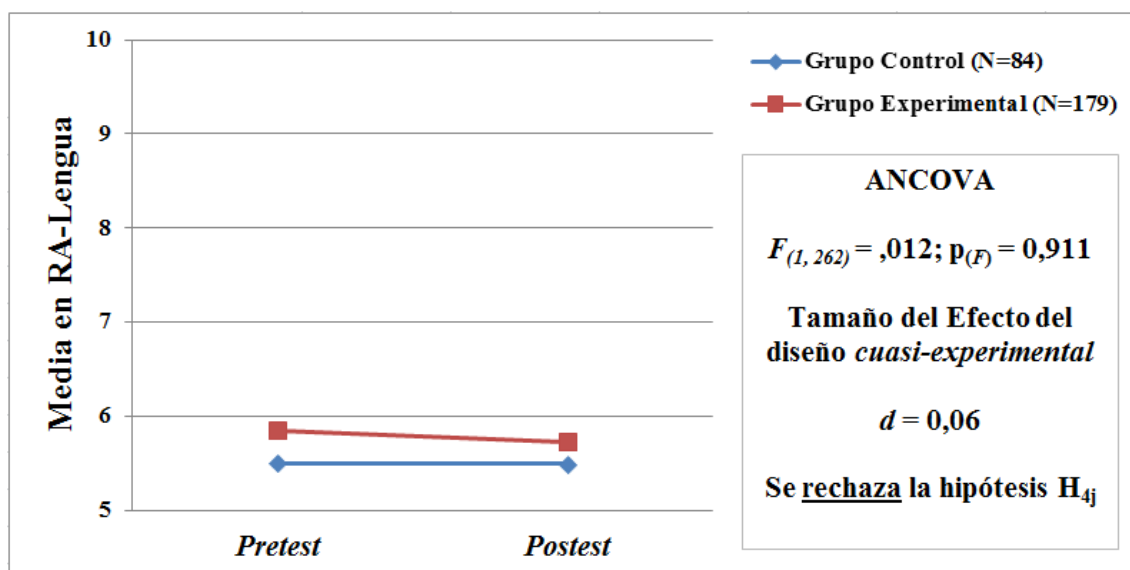


Figura 7.14. Resumen de los resultados del diseño *cuasi-experimental* sobre el rendimiento académico en Lengua

En la siguiente Tabla 7.40 se resumen los resultados relativos a todas las anteriores hipótesis específicas $H_{4a} \dots H_{4j}$; todas ellas enmarcadas en la directiva de *adecuación* y abordadas a través del diseño *cuasi-experimental* descrito. Tal y como puede observarse en la tabla, y según hemos ido viendo en las páginas anteriores, el diseño *cuasi-experimental* sólo genera diferencias estadísticamente significativas, y a favor del grupo experimental, en la variable ‘pensamiento computacional’ medida a través de nuestro TPC. Por tanto, todas las hipótesis específicas enmarcadas en la directiva de *adecuación* $H_{4a} \dots H_{4j}$ son rechazadas; a excepción de la H_{4g} , que se acepta: “(En comparación con el currículum TIC tradicional) El curso K-8 mejora la aptitud ‘pensamiento computacional’ de los estudiantes”. Con tres matices que, igualmente, se indican sobre la Tabla 7.40:

- ✓ Sí se encontró una F significativa sobre el *postest* del PMA-R, a favor del grupo experimental, en el IES San Antonio de Benagéber.
- ✓ Parece plausible que las F sobre el *postest* del RP30 y del índice ‘ICI’ del FI-R, se vuelvan estadísticamente significativas a favor del grupo experimental, replicando el diseño con muestras de mayor tamaño en dichas medidas.
- ✓ La F sobre el *postest* del TPC fue muy significativa, y a favor del grupo experimental, tanto en primer como en segundo ciclo de la ESO.

Tabla 7.40. Resumen de resultados del diseño *cuasi-experimental* sobre todas las variables aptitudinales y de rendimiento

	<i>Muestra</i>		<i>Pretest</i>		<i>Postest</i>		<i>Ganancia</i>		<i>ANCOVA sobre postest</i>						
	Control	Experimental	Control	Experimental	Control	Experimental	Control	Experimental	<i>F</i>	<i>p(F)</i>	<i>d_{global}</i>				
	N	N	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	\bar{X}	<i>s</i>	$\bar{X}_{post} - \bar{X}_{pre}$	$\bar{X}_{post} - \bar{X}_{pre}$			
PMA-R^a	57	118	14,61	6,494	13,36	5,972	17,14	7,296	16,28	6,700	2,53**	2,92**	,020	,887	0,06
PMA-E	61	130	16,23	11,097	13,02	14,178	24,02	15,244	21,30	15,823	7,79**	8,28**	,003	,957	0,04
PMA-N	51	126	12,73	7,153	14,36	8,182	16,73	8,464	18,49	9,039	4,00**	4,13**	,577	,448	0,02
PMA-V	57	130	24,56	6,167	24,83	6,423	26,70	7,106	26,83	6,969	2,14**	2,00**	,002	,961	0,02
RP30^b	21	55	29,19	15,102	21,91	19,172	31,24	13,397	30,05	20,058	2,05	8,14**	2,444	,122	0,34
FI-R (A-E)	17	54	29,53	8,186	30,72	7,659	36,12	9,656	35,89	7,813	6,59**	5,17**	,448	,505	0,18
FI-R (ICI)^c	17	54	89,71	8,865	91,50	8,149	87,94	12,311	91,72	8,531	-1,77	0,22	1,529	,220	0,24
TPC^d	54	332	16,67	3,629	16,39	4,391	17,91	4,602	20,30	4,562	1,24**	3,91**	25,225**	,000	0,62
RA-Inf	65	179	7,25	1,531	6,97	1,722	7,35	1,690	7,07	1,678	0,10	0,10	,376	,541	0,00
RA-Mat	84	179	5,43	2,202	6,16	1,808	5,31	2,173	5,84	1,891	-0,12	-0,32**	,105	,746	0,10
RA-Len	84	179	5,50	1,898	5,84	1,666	5,49	1,794	5,73	1,714	-0,01	-0,11	,012	,911	0,06

** Significativo al nivel < 0,01

PMA-R = Razonamiento Lógico; PMA-E: Espacial; PMA-N: Numérico; PMA-V: Verbal; RP30: Resolución de Problemas; FI-R (A-E): Percepción y Atención; FI-R (ICI): Control de la Impulsividad; TPC: Pensamiento Computacional; RA-Inf: Calificación en Informática; RA-Mat: Calificación en Matemáticas; RA-Len: Calificación en Lengua

a. *Si se encontró una F significativa sobre el postest del PMA-R, a favor del grupo experimental, en el IES San Antonio de Benagéber*

b. *y c. Parece plausible que las F sobre el postest del RP30 y del índice 'ICI' del FI-R, se vuelvan estadísticamente significativas a favor del grupo experimental, replicando el diseño con muestras de mayor tamaño en dichas medidas*

d. *La F sobre el postest del TPC fue muy significativa tanto en primer como en segundo ciclo de la ESO.*

Pasamos ahora a presentar los resultados relativos a las hipótesis específicas enunciadas en el marco de la directiva de *viabilidad*; comenzando por el conjunto de hipótesis $H_{4k...H_{4o}}$ ⁴⁰¹, que se contrastan a partir de los datos recogidos en el “*Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org*” (Román-González, 2015c). A dicho cuestionario contestaron 13 de los 16 profesores participantes en esta investigación.

Así, abordando la hipótesis H_{4k} , en la Tabla 7.41 se detallan las respuestas de los profesores sobre el número de semanas lectivas dedicadas al curso K-8. Tal y como puede verse, más del 50% de los profesores informan de que en los grupos de clase experimentales se han dedicado al menos 8 semanas lectivas al curso K-8. Por tanto, **se acepta la hipótesis H_{4k}** .

Tabla 7.41. Respuestas de los profesores sobre el número de semanas lectivas dedicadas al curso K-8

<i>¿Cuántas semanas lectivas has dedicado en tu/s grupo/s al Curso K-8 de Code.org?</i>					
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado	
Válido	10 semanas o más	5	38,5	38,5	
	9 semanas	2	15,4	15,4	
	8 semanas	2	15,4	15,4	69,2
	7 semanas	1	7,7	7,7	76,9
	6 semanas	3	23,1	23,1	100,0
	Total	13	100,0	100,0	

Siguiendo con la hipótesis H_{4l} , en la Tabla 7.42 se detallan las respuestas de los profesores sobre el porcentaje de etapas del curso K-8 impartidas realmente. Tal y como puede verse, más del 50% de los profesores informan de que en los grupos de clase experimentales se han impartido al menos el 80% de las 20 etapas que componen el curso K-8. Por tanto, **se acepta la hipótesis H_{4l}** .

Tabla 7.42. Respuestas de los profesores sobre el % de etapas del curso K-8 impartidas realmente

<i>De las 20 etapas de las que se compone el Curso K-8 de Code.org, ¿qué % aproximado has impartido en tu/s grupo/s?</i>					
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado	
Válido	100% de las etapas	4	30,8	30,8	
	90%	4	30,8	30,8	61,5
	80%	1	7,7	7,7	69,2
	70%	1	7,7	7,7	76,9
	60%	2	15,4	15,4	92,3
	50% de las etapas o menos	1	7,7	7,7	100,0
	Total	13	100,0	100,0	

Siguiendo con la hipótesis H_{4m} , en la Tabla 7.43 se detallan las respuestas de los profesores sobre qué partes del curso K-8 (*‘on-line’*, *‘unplugged’* o ambas) se han impartido realmente. Tal y como puede verse, menos del 80% de los profesores informan de que en los grupos de clase experimentales se han realizado tanto etapas *‘on line’* como etapas *‘unplugged’*. Por tanto, **se rechaza la hipótesis H_{4m}** .

⁴⁰¹ Dado que estas hipótesis específicas están todas ellas referidas a indicadores descriptivos, en las siguientes tablas de resultados se marca en negrita el punto de corte establecido en la hipótesis, así como la decisión estadística que se deriva del resultado encontrado (en verde se acepta, y en rojo se rechaza). Para el establecimiento del punto de corte nos hemos guiado por los resultados previos encontrados en el estudio exploratorio sobre ‘La Hora del Código’ (Capítulo 5)

Tabla 7.43. Respuestas de los profesores sobre qué partes del curso K-8 (*'on-line'*, *'unplugged'* o ambas) se han impartido realmente

¿Qué parte del Curso K-8 de Code.org has impartido en tu/s grupo/s?

		Frecuencia	Porcentaje	Porcentaje válido
Válido	Tanto etapas <i>'on-line'</i> como <i>'unplugged'</i>	10	76,9	76,9
	Sólo etapas <i>'on-line'</i>	3	23,1	23,1
	Total	13	100,0	100,0

Siguiendo con la hipótesis H_{4n} , en la Tabla 7.44 se detallan las respuestas de los profesores sobre su percepción del grado de aprendizaje de sus estudiantes con el curso K-8 (desde $1=Poco$ aprendizaje hasta $5=Mucho$ aprendizaje). Tal y como puede observarse en la Tabla 7.45, tanto la media, la mediana como la moda se sitúan al menos en una puntuación 4 sobre 5. Por tanto, **se acepta la hipótesis H_{4n} .**

Tabla 7.44. Respuestas de los profesores sobre su percepción del grado de aprendizaje de sus estudiantes con el K-8

Globalmente, ¿cuál crees que ha sido el grado de aprendizaje de tus estudiantes con el Curso K-8?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	5 = Mucho aprendizaje	2	15,4	15,4	15,4
	4	9	69,2	69,2	84,6
	3	2	15,4	15,4	100,0
	Total	13	100,0	100,0	

Tabla 7.45. Estadísticos relativos a la pregunta “Globalmente, ¿cuál crees que ha sido el grado de aprendizaje de tus estudiantes con el Curso K-8?”

N	Válido	13
	Perdidos	0
Media		4,00
Mediana		4,00
Moda		4
Desviación estándar		,577
Varianza		,333
Mínimo		3
Máximo		5

Siguiendo con la hipótesis $H_{4ñ}$, en la Tabla 7.46 se detallan las respuestas de los profesores sobre su nivel de satisfacción global con el curso K-8 (desde $1=Muy$ bajo hasta $5=Muy$ alto). Tal y como puede observarse en la Tabla 7.47, tanto la media, la mediana como la moda se sitúan al menos en una puntuación 4 sobre 5. Por tanto, **se acepta la hipótesis $H_{4ñ}$.**

Tabla 7.46. Respuestas de los profesores sobre su nivel de satisfacción global con el K-8

Como profesor/a responsable de la implantación del Curso K-8 de Code.org, ¿cuál es tu nivel de satisfacción con el curso globalmente?

		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	5 = Muy alto	7	53,8	53,8	53,8
	4	6	46,2	46,2	100,0
	Total	13	100,0	100,0	

Tabla 7.47. Estadísticos relativos a la pregunta “*Como profesor/a responsable de la implantación del Curso K-8 de Code.org, ¿cuál es tu nivel de satisfacción con el curso globalmente?*”

N	Válido	13
	Perdidos	0
Media		4,54
Mediana		5,00
Moda		5
Desviación estándar		,519
Varianza		,269
Mínimo		4
Máximo		5

Acabamos con este grupo de hipótesis específicas con la H_{40} . En la Tabla 7.48 se detallan las respuestas de los profesores sobre su intención de volver a impartir el K-8. Tal y como puede observarse, más del 80% de los profesores (de hecho, el 100%) manifiesta su intención de impartir el curso K-8 de nuevo en el próximo curso académico. Por tanto, **se acepta la hipótesis H_{40}** .

Tabla 7.48. Respuestas de los profesores sobre su intención de volver a impartir el curso K-8

<i>Después de la experiencia, ¿tienes intención de impartir el Curso K-8 nuevamente el próximo año académico?</i>			
		Frecuencia	Porcentaje
Válido	Sí	13	100,0
			Porcentaje válido
			100,0

Adicionalmente, se presenta un resumen de las respuestas de los profesores a las dos preguntas abiertas del “*Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org*” (Román-González, 2015c); a saber: “*Brevemente, indica los aspectos más positivos y más negativos (puntos mejorables) del Curso K-8*” y “*Opcionalmente, relata alguna anécdota o incidente especialmente reseñable que ilustre vuestra experiencia con el Curso K-8*” (Tabla 7.49)

Tabla 7.49. Resumen de las respuestas de los profesores a las preguntas abiertas del cuestionario sobre el curso K-8

Aspectos positivos	Aspectos negativos
<ul style="list-style-type: none"> ✓ El K-8 permite la autonomía de los alumnos. El curso permite atender distintos ritmos de aprendizaje, y que cada alumno alcance metas diferentes. También ofrece flexibilidad (p.e. los alumnos pueden avanzar en casa en caso de ir retrasados) ✓ La alta motivación mostrada por los estudiantes durante el K-8 (incluso para trabajar en casa). Algunos de los aspectos que favorecen esta alta motivación son: aspecto visual y gamificado del curso (personajes conocidos que se mueven a través de programas escritos en lenguaje visual por bloques); las pistas y videos de ayuda para la resolución de los niveles-retos de ‘coding’; sencillez de uso de la plataforma; posibilidad de que el alumno pueda ver su progreso en el curso. ✓ El curso K-8 desarrolla algunos aprendizajes y habilidades importantes como: autogestión del estrés y la frustración (cuando un alumno no consigue resolver un nivel-reto de ‘coding’); trabajo en equipo 	<ul style="list-style-type: none"> ✓ El K-8 no incentiva suficientemente la creatividad: hasta llegar a la etapa 19 no se permite que el alumno evoque-escriba un programa desde un lienzo en blanco (‘open-ended’) ✓ Algunos profesores señalan problemas de temporalización del curso: p.e. que haciendo el 100% de las 20 etapas es demasiado extenso; y al ritmo de 2 horas de clase semanales, contando con festivos, vacaciones e incidencias varias puede llevar más allá de un trimestre. También se señala que, al impartir el curso en inglés, el ritmo se ralentiza. ✓ Para el (pequeño) grupo de alumnos que no les ha gustado el curso, éste puede resultar “<i>como una tortura</i>”, dado el carácter similar y repetitivo de los niveles-retos de ‘coding’ propuestos. Este carácter repetitivo puede fatigar. ✓ También se señala la dificultad de algunos conceptos computacionales para algunos estudiantes de 1º ESO: en especial, los bucles anidados, y el cálculo de ángulos

Aspectos positivos	Aspectos negativos
<p>y colaboración entre compañeros; atención, concentración, compromiso con la tarea (p.e. algunos profesores relatan cómo los alumnos “<i>admiten que el K-8 les ha ayudado sorprendentemente a fijar la atención durante más tiempo</i>”).</p> <ul style="list-style-type: none"> ✓ Adicionalmente, se señala que el curso K-8 ha permitido trabajar transversalmente la competencia en lengua inglesa (la interfaz del curso utilizada fue siempre en lengua inglesa). ✓ Globalmente, los profesores señalan que el K-8 ha permitido desarrollar el pensamiento computacional de los alumnos; y que es muy útil para adquirir los conceptos computacionales básicos, y como paso previo a la introducción de los lenguajes de programación textuales. ✓ En general, se señala que a los alumnos les han gustado más las etapas ‘<i>on-line</i>’ que las ‘<i>unplugged</i>’. Respecto de las etapas ‘<i>unplugged</i>’ se valoran positiva y especialmente la etapa 6 (‘Algoritmos’ a través del juego Tangram), 8 (‘Funciones’ a través de construcción de pulseras de abalorios), y 12 (concepto computacional ‘funciones’ trabajado a través de los estribillos de canciones). Una profesora señala que, para la planificación de las etapas ‘<i>unplugged</i>’, fue de mucha utilidad el foro en línea que utilizaron para comunicarse todos los profesores de la red ‘<i>IT Teaching</i>’: en este foro los profesores fueron compartiendo testimonios gráficos del desarrollo del curso K-8, algunos de los cuales se adjuntan en el Anexo L. 	<p>(necesario para girar correctamente a los personajes).</p> <ul style="list-style-type: none"> ✓ Aparecen algunas críticas a las actividades ‘<i>unplugged</i>’: p.e. que algunas resultan demasiado sencillas o “<i>tontas</i>” para 2º ESO. Se señalan como especialmente prescindibles las etapas 14 (‘Abstracción’) y 20 (‘Cierre’). ✓ Problemas de conectividad en los centros que dificultan la realización del curso.

Anécdotas

- ✓ “*Un día no había conexión a Internet en el aula, y los alumnos/as se enfadaron mucho y no sabían qué podían hacer en el aula con los ordenadores porque no podían seguir el curso online*”
- ✓ “*Algunos de los alumnos que prácticamente no hacen nada en otros temas de informática o en otras asignaturas se han sentido muy motivados. Algunos incluso han sido de los más aventajados en este tema en concreto*”
- ✓ “*Para mí ha sido un shock ver cómo uno de los alumnos que acabó muy pronto el curso, se puso a diseñar proyectos con bananas y demás fauna y cómo sus compañeros admiran lo que su compañero es capaz de realizar*”⁴⁰²
- ✓ “*Les gustó mucho el tema de las funciones explicado con el estribillo de las canciones*” [Etapa 12]
- ✓ “*Normalmente al acabar la lección, les dejo 5 minutos para que puedan jugar a juegos online (si se han comportado bien)... Pues después del curso algunos en lugar de jugar a estos juegos, se conectaban a code.org para programar el Flappy Bird*”⁴⁰³

Decisiones

- ✓ “*El próximo curso sólo realizarán el K-8 los de 1º ESO, y en 2º y 3º ESO trabajarán con Scratch*”

⁴⁰² Testimonio de la profesora de 1º ESO del IES María Blasco; en dicho aula se siguió una metodología ‘*self-paced*’ (al ritmo del propio alumno), que permitió emerger la alta capacidad computacional. Dos de los estudiantes de este grupo acabaron el K-8 en apenas un par de semanas y aceleraron al curso ‘*Computer Programming*’ de Khan Academy. Volveremos a ellos en el estudio de casos que se presenta en el [apartado 7.3](#).

⁴⁰³ Dentro de la plataforma Code.org, además del curso K-8, hay otros muchos entornos de programación como, por ejemplo, uno para aprender a codificar el conocido juego “*Flappy Bird*”: <https://studio.code.org/flappy/1>

Pasamos ahora a la hipótesis específica H_{4p} , que es contrastada a través de los datos de seguimiento (*'tracking'*) que aporta la plataforma de Code.org. En la Tabla 7.50 (y Figura 7.15) se presentan los estadísticos relativos a los 'Niveles Completados' por los estudiantes. Tal y como puede observarse, el percentil 50 (mediana) se sitúa en 85 niveles completados (lo que supone un 86,7% sobre el total de 98 niveles-pantallas que componen el curso K-8). Así pues, podemos afirmar que más del 50% de los estudiantes completan al menos el 80% de los 98 niveles-pantallas *'on-line'* que componen el curso K-8; y, por tanto, **se acepta la hipótesis H_{4p}** .

Tabla 7.50. Estadísticos sobre los 'niveles-pantallas' completados por los estudiantes del curso K-8

<i>Niveles-Pantallas Completados [K-8 Code.org]</i>		
N	Válido	385
	Perdidos	0
Media		82,21
Mediana		85,00
Moda		98
Desviación estándar		21,842
Asimetría		-,132
Mínimo		24
Máximo		157
Percentiles	10	53,00
	20	63,20
	30	71,00
	40	78,00
	50	85,00
	60	92,00
	70	97,00
	80	98,00
	90	108,00

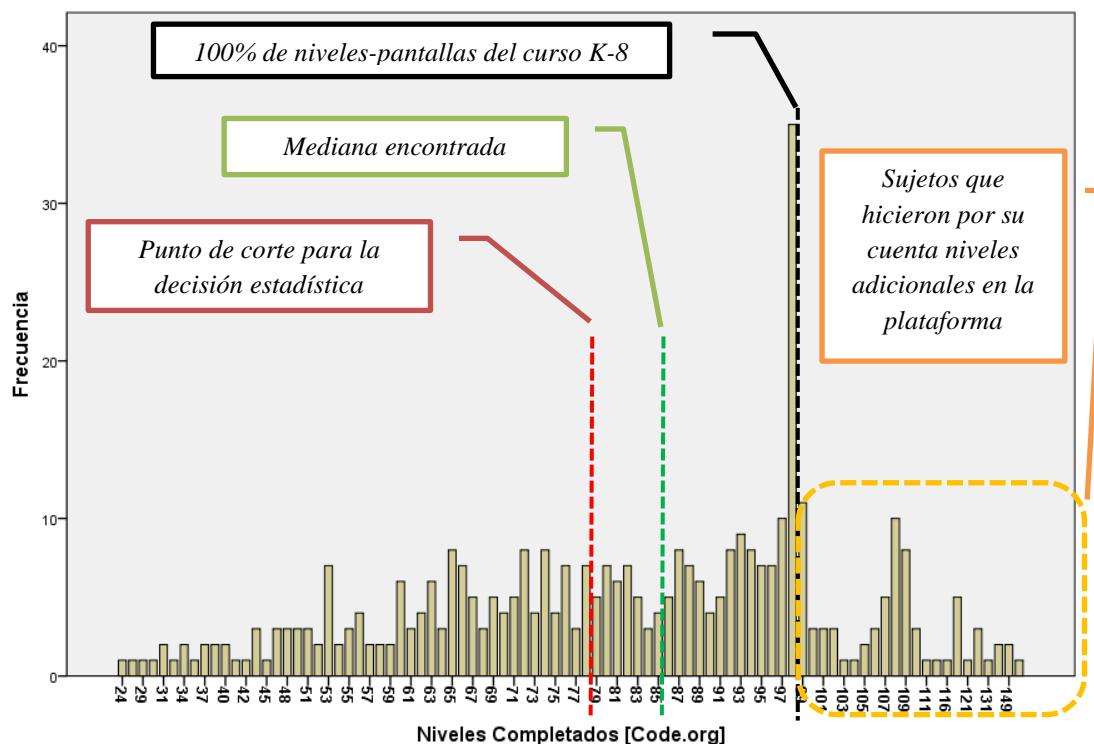


Figura 7.15. Distribución de frecuencias del número de 'niveles-pantallas' completados por los estudiantes K-8

Tal y como puede observarse en la anterior figura, nos encontramos con aproximadamente un 20% de estudiantes, en la parte derecha de la distribución, que no sólo completaron el 100% de niveles del curso K-8 (98 niveles), sino que adicionalmente completaron niveles en otros entornos de programación que ofrece la plataforma Code.org (por ejemplo, ‘Código Flappy’⁴⁰⁴ o ‘Play Lab’⁴⁰⁵).

Pasamos ahora a presentar los resultados relativos al conjunto de hipótesis $H_{4q} \dots H_{4u}$, que se contrastan a partir de los datos recogidos en el “Cuestionario de satisfacción para estudiantes sobre el Curso K-8 de Code.org” (Román-González, 2015d). A dicho cuestionario contestaron 267 de los 401 estudiantes participantes en los grupos experimentales de esta investigación.

Así, abordando la hipótesis H_{4q} , en la Tabla 7.51 se detallan las respuestas de los estudiantes sobre su percepción de autoeficacia en ‘coding’ tras la realización del curso K-8 (entre 1=Se me da muy mal y 5=Se me da muy bien). Tal y como puede observarse en la Tabla 7.52, los estudiantes manifiestan una percepción de autoeficacia en un grado de al menos 4 sobre 5 (mediana ≥ 4); por tanto, **se acepta la hipótesis H_{4q}** .

Tabla 7.51. Respuestas de los estudiantes sobre su percepción de autoeficacia en ‘coding’ tras el curso K-8

<i>Tras tu experiencia con el Curso K-8 ¿cómo dirías que se te da la ‘programación informática’ (‘coding’)?</i>				
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Se me da muy bien	40	15,0	15,0
	Se me da bien	159	59,6	74,5
	Se me da regular	58	21,7	96,3
	Se me da mal	8	3,0	99,3
	Se me da muy mal	2	,7	100,0
	Total	267	100,0	100,0

Tabla 7.52. Estadísticos relativos a la pregunta sobre percepción de autoeficacia en ‘coding’ tras el curso K-8

N	Válido	267
	Perdidos	0
Media		3,85
Mediana		4,00
Moda		4
Desviación estándar		,731
Varianza		,534
Mínimo		1
Máximo		5

Siguiendo con la hipótesis H_{4r} , en la Tabla 7.53 se detallan las respuestas de los estudiantes sobre su motivación por continuar aprendiendo ‘coding’ tras el curso K-8 (entre 1=Nada motivado y 5=Muy motivado). Tal y como puede observarse en la Tabla 7.54, los estudiantes manifiestan motivación en un grado de al menos 4 sobre 5 (mediana ≥ 4); por tanto, **se acepta la hipótesis H_{4r}** .

⁴⁰⁴ <https://studio.code.org/flappy/1>

⁴⁰⁵ <https://studio.code.org/s/playlab/stage/1/puzzle/1>

Tabla 7.53. Respuestas de los estudiantes sobre su motivación por continuar aprendiendo 'coding' tras el K-8

Tras el Curso K-8, ¿cómo de motivado te sientes para continuar aprendiendo 'programación informática' ('coding')?

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Muy motivado	50	18,7	18,7
	Bastante motivado	96	36,0	54,7
	Algo motivado	89	33,3	88,0
	Poco motivado	23	8,6	96,6
	Nada motivado	9	3,4	100,0
	Total	267	100,0	100,0

Tabla 7.54. Estadísticos relativos a la pregunta sobre motivación por continuar aprendiendo 'coding' tras el K-8

N	Válido	267
	Perdidos	0
Media		3,58
Mediana		4,00
Moda		4
Desviación estándar		,998
Varianza		,996
Mínimo		1
Máximo		5

Siguiendo con la hipótesis H_{4s} , en la Tabla 7.55 se detallan las respuestas de los estudiantes sobre su intención de continuar aprendiendo 'coding' a través de plataformas-tutoriales en Internet tras el curso K-8 ('sí' o 'no'). Tal y como puede observarse en la tabla, algo menos del 80% de los estudiantes manifiesta dicha intención; por tanto, **se rechaza la hipótesis H_{4s}** .

Tabla 7.55. Respuestas de los estudiantes sobre su intención de continuar aprendiendo 'coding online' tras el K-8

¿Te gustaría continuar aprendiendo 'programación informática' ('coding') a través de plataformas-tutoriales en Internet?

	Frecuencia	Porcentaje	Porcentaje válido
Válido	Sí	209	78,3
	No	58	21,7
	Total	267	100,0

Siguiendo con la hipótesis H_{4t} , en la Tabla 7.56 se detallan las respuestas de los estudiantes sobre en qué grado consideran que han aprendido con el curso K-8 (entre 1=No he aprendido nada y 5=He aprendido mucho). Tal y como puede observarse en la Tabla 7.57, los estudiantes consideran que han aprendido en un grado de al menos 4 sobre 5 (mediana ≥ 4); por tanto, **se acepta la hipótesis H_{4t}** .

Tabla 7.56. Respuestas de los estudiantes sobre su grado de aprendizaje con el curso K-8

Globalmente, ¿cuánto crees que has aprendido con el Curso "K-8 Intro to Computer Science"?

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	5 = He aprendido mucho	83	31,1	31,1
	4	125	46,8	77,9
	3	48	18,0	95,9
	2	9	3,4	99,3
	1= No he aprendido nada	2	,7	100,0
	Total	267	100,0	100,0

Tabla 7.57. Estadísticos relativos a la pregunta sobre grado de aprendizaje percibido con el curso K-8

N	Válido	267
	Perdidos	0
Media		4,04
Mediana		4,00
Moda		4
Desviación estándar		,833
Varianza		,694
Mínimo		1
Máximo		5

Para acabar con este grupo de hipótesis específicas, abordamos la H_{4u} ; en la Tabla 7.58 se detallan las respuestas de los estudiantes sobre su valoración global del curso K-8 (entre $1=No\ me\ ha\ gustado\ nada$ y $5=Me\ ha\ gustado\ mucho$). Tal y como puede observarse en la Tabla 7.59, los estudiantes valoran globalmente el curso K-8 en un grado de al menos 4 sobre 5 (mediana ≥ 4); por tanto, se **acepta la hipótesis H_{4u}** .

Tabla 7.58. Respuestas de los estudiantes sobre su valoración global del curso K-8

Globalmente, ¿qué valoración haces del Curso "K-8 Intro to Computer Science"?

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
5 = Me ha gustado mucho	87	32,6	32,6	32,6
4	108	40,4	40,4	73,0
3	59	22,1	22,1	95,1
2	6	2,2	2,2	97,4
1 = No me ha gustado nada	7	2,6	2,6	100,0
Total	267	100,0	100,0	

Tabla 7.59. Estadísticos relativos a la pregunta sobre la valoración global del curso K-8

N	Válido	267
	Perdidos	0
Media		3,98
Mediana		4,00
Moda		4
Desviación estándar		,936
Varianza		,876
Mínimo		1
Máximo		5

Para complementar los resultados relativos a las hipótesis previas $H_{4p} \dots H_{4u}$, en la siguiente Tabla 7.60 se presentan las pruebas t de diferencia de medias en las anteriores variables según sexo. Tal y como puede observarse, no se encuentran diferencias significativas por sexo en niveles-pantallas completados, es decir, en una medida de desempeño objetivo en el curso K-8. En lo relativo a las preguntas del cuestionario para estudiantes, sí se encuentran diferencias significativas a favor de los chicos ($p_{(t)} > 0,05$) en motivación por continuar aprendiendo ‘coding’ tras el curso K-8, y en la valoración global del curso.

Tabla 7.60. Pruebas *t* de diferencia de medias en las preguntas del cuestionario para estudiantes del K-8, según sexo

	Sexo	N	Media	Desviación estándar	<i>t</i>	<i>P</i> (<i>t</i>)	<i>d</i> de Cohen
Niveles-pantallas completados en K-8	Chico	247	82,77	21,342	,663	,508	0,07
	Chica	138	81,22	22,754			
Autoeficacia en 'coding' tras K-8	Chico	174	3,90	,738	1,421	,156	0,19
	Chica	93	3,76	,713			
Motivación seguir aprendiendo 'coding' tras K-8	Chico	174	3,68	,997	2,202*	,029	0,28
	Chica	93	3,40	,980			
Determinación aprender 'coding' en entornos 'on-line' tras K-8	Chico	174	,79	,410	,248	,805	0,05
	Chica	93	,77	,420			
Percepción global de aprendizaje con K-8	Chico	174	3,99	,870	-1,261	,208	0,17
	Chica	93	4,13	,755			
Valoración global del K-8	Chico	174	4,07	,931	2,249*	,025	0,28
	Chica	93	3,81	,924			

* Significativo al nivel $p(t) > 0,05$

Finalizamos este epígrafe de resultados con los relativos a la hipótesis específica H_{4v} , ya en el marco de la directiva de *relevancia*. Tal y como hemos ido viendo a lo largo de la tesis, el marco de *relevancia* exige de un abordaje más cualitativo. En este sentido, la hipótesis H_{4v} es contrastada a través de las respuestas a la última pregunta (abierta) del “Cuestionario de satisfacción para estudiantes sobre el Curso K-8 de Code.org” (Román-González, 2015d); que dice: “¿Consideras importante la 'programación informática' ('coding') para tu formación como ciudadano del siglo XXI? ¿Sí, No? ¿Por qué? Razona por favor tu respuesta”. En la Figura 7.16 se muestra el código QR de enlace a un documento PDF⁴⁰⁶ con las respuestas ‘en bruto’ a dicha pregunta.



Figura 7.16. Código QR de enlace a las respuestas abiertas del “Cuestionario de satisfacción para estudiantes sobre el Curso K-8 de Code.org”

En la siguiente Tabla 7.61 se muestran las frecuencias de respuesta a dicha pregunta tras ser categorizadas como ‘sí’, ‘no’ o ‘tal vez / no sé / depende’. Tal y como puede observarse, tras el curso K-8, más del 80% de los estudiantes considera importante el ‘coding’ para su formación como ciudadano del siglo XXI; por tanto, se acepta la hipótesis H_{4v}

Tabla 7.61. Respuestas de los estudiantes acerca de la importancia del ‘coding’ para su formación como ciudadanos

	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Sí	227	85,0	85,0
	No	24	9,0	94,0
	Tal vez / No lo sé / Depende	16	6,0	100,0
	Total	267	100,0	100,0

⁴⁰⁶ Disponible en <https://db.tt/DMynZaLZ>

Replicando el análisis cualitativo de contenido que hicimos en el estudio exploratorio sobre ‘La Hora del Código’, aplicamos ahora el sistema de categorías allí generado sobre las razones que aportan los estudiantes del K-8 que defienden la importancia del ‘coding’ para su formación como ciudadanos del siglo XXI. Tal y como puede observarse en la Figura 7.17, las razones que esgrimen los estudiantes son, de mayor a menor frecuencia: ‘utilitarias’ (≈ “*porque el ‘coding’ me servirá para encontrar trabajo...*”); ‘de desarrollo’ (≈ “*porque con el ‘coding’ aprendo/desarrollo la lógica...*”); ‘sociológicas’ (≈ “*porque el mundo es cada vez más tecnológico/digital y...*”); ‘genéricas’ (≈ “*el ‘coding’ es importante...*”); ‘críticas’ (≈ “*porque el ‘coding’ me permite controlar las máquinas, en vez de que las máquinas me controlen a mí...*”); ‘escolares’ (≈ “*porque el ‘coding’ mejora mis resultados en el colegio, me hace falta en algunos trabajos escolares...*”); ‘expresivo-comunicativas’ (≈ “*porque con el ‘coding’ puedo crear una aplicación, un videojuego, una web*”); ‘emotivas’ (≈ “*porque con el ‘coding’ me siento bien, me gusta...*”)

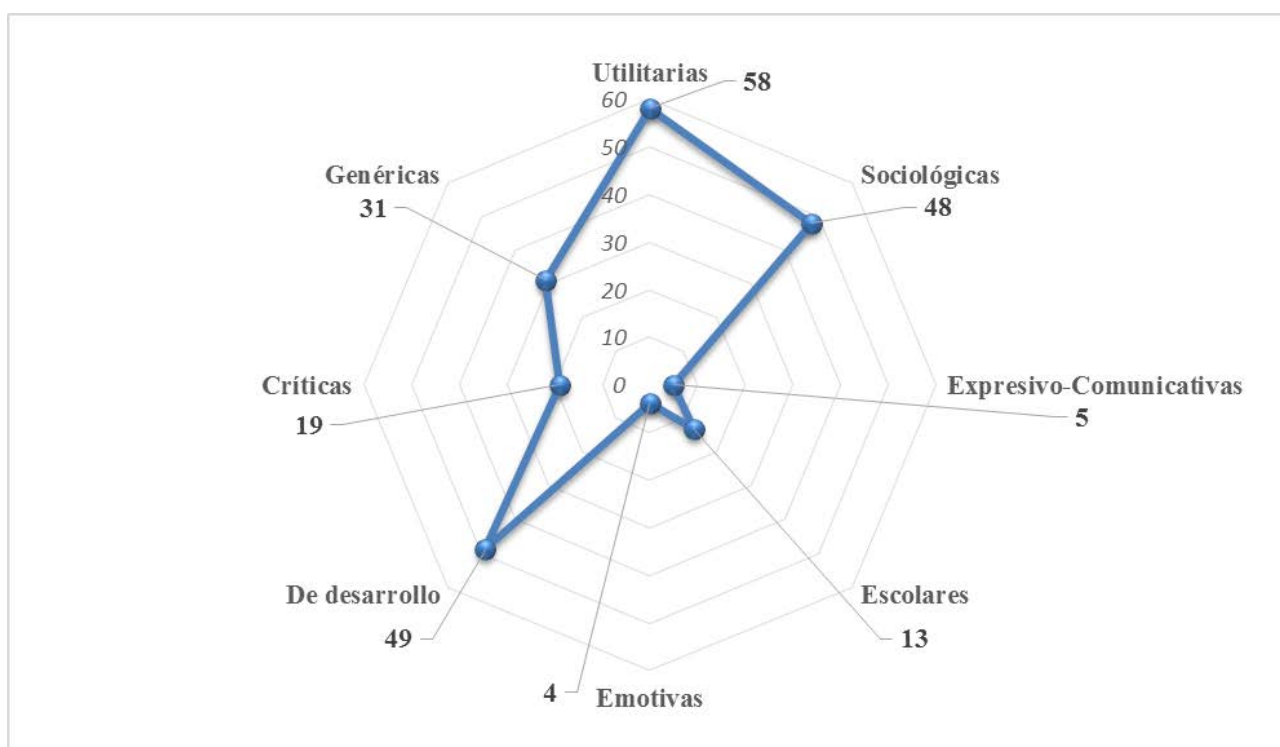


Figura 7.17. Diagrama radial de las razones esgrimidas por los estudiantes K-8 que defienden la importancia del ‘coding’

7.1.5. Discusión y conclusiones parciales

Con respecto a las hipótesis específicas $H_{4a} \dots H_{4d}$, podemos concluir que:

- En comparación con el currículum TIC tradicional, el curso K-8 no mejora significativamente ninguna de las aptitudes mentales primarias de los estudiantes, medidas a través de la batería PMA. Tanto los grupos experimentales como de control muestran ganancias estadísticamente significativas y de tamaño ‘pequeño’ a ‘moderado’ entre el *pretest* y *posttest*. Dado que este patrón de resultados se produce por igual en todas las aptitudes medidas por el PMA, para ambos grupos, y que el intervalo entre ambas aplicaciones de la batería es de sólo 10-12 semanas, la explicación más plausible para dichas ganancias *pretest-posttest* es el efecto del entrenamiento de los sujetos con las pruebas.

- En uno de los centros participantes, IES San Antonio de Benagéber, sí se encuentra un efecto significativo del diseño a favor del grupo experimental en el PMA-R. Este resultado es consistente dado que el PMA-R es el test de la batería PMA en el cual se encontró una mayor correlación con el TPC durante su validación criterial concurrente ($r = +0,442$; ver [sub-epígrafe 6.5.2.3.1.1](#)). Dado que el K-8 pretende ser un curso de códigoalfabetización y desarrollo del pensamiento computacional (PC), es coherente que pueda producir también mejoras en las variables que correlacionan con el PC, tal y como el razonamiento lógico (PMA-R). En cualquier caso, dado que el efecto significativo sólo se encontró en un centro, los resultados no son plenamente concluyentes y requieren de más investigación.

Con respecto a la hipótesis específica H_{4e} , podemos concluir que:

- En comparación con el currículum TIC tradicional, el curso K-8 no mejora significativamente la aptitud de ‘resolución de problemas’ de los estudiantes medida a través del RP30. Sin embargo, en este caso se encuentra una ganancia estadísticamente muy significativa, y de tamaño ‘moderado’ entre el *pretest* y *postest* del grupo experimental; no siendo así en el grupo de control. El ANCOVA sobre el *postest* no resulta estadísticamente significativo, pero se aproxima bastante al límite de significatividad ($p_{(F)} = 0,122 > 0,05$); llegando el tamaño global del efecto del diseño *cuasi-experimental* sobre el RP30 a un valor entre ‘pequeño’ y ‘moderado’ $d = 0,34$, que debe tomarse en consideración. Es plausible considerar que una replicación con muestras de tamaño algo mayor arrojará resultados estadísticamente favorables al K-8 respecto del currículum TIC tradicional en la aptitud ‘resolución de problemas’.

Con respecto a la hipótesis específica H_{4f} , podemos concluir que:

- En comparación con el currículum TIC tradicional, el curso K-8 no mejora significativamente la aptitud ‘perceptivo-atencional’ de los estudiantes medida a través del FI-R, en ninguno de los dos índices que aporta la aplicación de esta prueba: índice A-E (‘percepción y atención’) e índice ICI (‘control de la impulsividad’). Sin embargo, el patrón de resultados es diferencial entre ambos índices, lo cual merece un comentario:
 - En el caso del índice A-E, tanto el grupo control como experimental muestran ganancias estadísticamente muy significativas, y de tamaño entre ‘moderado’ y ‘grande’, desde el *pretest* al *postest*. Al igual que en la batería PMA, la explicación más plausible de dicha ganancia en ambos grupos es el efecto del entrenamiento de los sujetos en las pruebas. Sin embargo, dado el considerable tamaño de las ganancias *pre-post* ($d \approx 0,70$), también es plausible considerar que ambos currículum, tanto el TIC tradicional como el K-8, estén produciendo mejoras en el índice A-E. Ello podría confirmarse en una replicación del diseño añadiendo una tercera condición de control con sujetos que no estén siguiendo ningún currículum de Informática en absoluto.
 - En el caso del índice ICI, no se encuentran ganancias significativas en ninguno de los dos grupos entre el *pretest* y *postest*. Parece, por tanto, que es un índice más insensible al efecto del entrenamiento, lo cual es consistente con su naturaleza de ‘control de la impulsividad’, variable *a priori* más cercana a temperamento o estilo

típico de respuesta. En cualquier caso, el ANCOVA sobre el *postest* del ICI, aunque no resulta estadísticamente significativo, se aproxima al límite ($p_{(F)} = 0,220 > 0,05$); y el tamaño global del efecto del *diseño cuasi-experimental* sobre el ICI supera la barrera mínima para tomarlo en consideración ($d = 0,24 > 0,20$). Todo ello hace plausible que en una replicación con muestras de mayor tamaño se encuentren resultados estadísticamente favorables al K-8 respecto del currículum TIC tradicional en ‘control de la impulsividad’.

Con respecto a la hipótesis específica H_{4g} , podemos concluir que:

- En comparación con el currículum TIC tradicional, el curso K-8 sí mejora significativamente la aptitud ‘pensamiento computacional’ de los estudiantes, medida a través del TPC; tanto en primer ciclo como en segundo ciclo de la ESO. El tamaño global del efecto del *cuasi-experimento* sobre el ‘pensamiento computacional’ está entre ‘moderado’ y ‘grande’ ($d = 0,62$). Ello es consistente con la diferencia fundamental entre ambos currículum informáticos, a saber, que el TIC tradicional está basado en una perspectiva del estudiante como mero usuario de tecnología digital; frente al K-8, orientado explícitamente a la codigoalfabetización y el desarrollo del pensamiento computacional de los alumnos.
- Adicionalmente, queremos señalar que, segmentando la muestra por sexo, el ANCOVA sobre el *postest* del TPC resulta también estadísticamente muy significativo tanto para los chicos ($F_{(1,244)} = 13,124$; $p_{(F)} = 0,000 < 0,01$; $d_{\text{global}} = 0,59$) como para las chicas ($F_{(1,140)} = 7,879$; $p_{(F)} = 0,006 < 0,01$; $d_{\text{global}} = 0,61$). Así, podríamos afirmar que el K-8, respecto al currículum TIC, mejora el pensamiento computacional con una magnitud muy similar en ambos sexos.

Recapitulando sobre el conjunto de hipótesis específicas que involucran algún tipo de ‘aptitud’, es decir, hipótesis $H_{4a} \dots H_{4g}$, cabe destacar que, en ninguna de ellas, se encontró diferencia significativa alguna en situación *pretest* entre el grupo ‘experimental’ y ‘control’. Ello es evidencia de la relativa equivalencia inicial entre ambos grupos; y apoya la validez del proceso seguido para la conformación global de nuestro *diseño cuasi-experimental* (Tabla 7.6)

Con respecto a las hipótesis específicas $H_{4h} \dots H_{4j}$, podemos concluir que:

- En comparación con el currículum TIC tradicional, el curso K-8 no mejora significativamente el rendimiento académico en ninguna de las tres asignaturas consideradas: Informática, Matemáticas y Lengua. Los rendimientos en ambos grupos son muy similares entre sí, y resistentes a cambios entre la situación *pre* y *post*. Es destacable que, comparando asignaturas, las calificaciones promedio en Informática se sitúan en el entorno del 7, frente a Matemáticas y Lengua, cuyas calificaciones promedio se sitúan entre el 5 y el 6.

Considerando globalmente todas las hipótesis específicas anteriores $H_{4a} \dots H_{4j}$, enmarcadas en la directiva de *adecuación*, podemos afirmar que:

- La única variable aptitudinal en la que se han encontrado mejoras estadísticamente significativas en los estudiantes que siguieron el K-8 frente a los estudiantes que siguieron el currículum TIC tradicional, ha sido el ‘pensamiento computacional’ medido a través de nuestro TPC (sólo se ha aceptado la hipótesis específica H_{4g}). Este único resultado

significativo nos lleva a reafirmar la necesidad que existía de construir un instrumento explícitamente diseñado para medir el ‘pensamiento computacional’ y que fuera, por tanto, sensible a medir los efectos de cursos-programas orientados a desarrollar dicha aptitud. Ello también puede considerarse como una nueva evidencia de la validez discriminante de nuestro TPC, en el sentido de que nuestro instrumento es capaz de discriminar entre sujetos que han seguido un currículum TIC tradicional y sujetos que han seguido un currículum orientado al desarrollo del pensamiento computacional, como el curso K-8. Así se comprueba al calcular la Curva COR de potencia diagnóstica del TPC (*postest*) para discriminar entre ambos grupos de sujetos (Figura 7.18); como puede verse en la Tabla 7.62 el área bajo la curva COR es $A_{COR} = 0,696^{**}$, estadísticamente muy significativa, considerándose una potencia diagnóstica ‘buena’. También en la Figura 7.18 se ilustra el punto de corte diagnóstico más óptimo: una puntuación en el *postest* del TPC $> 18,50$ identifica correctamente un 70% de ‘verdaderos positivos’ (sujetos que han hecho el K-8), frente a sólo un 35% de ‘falsos positivos’.

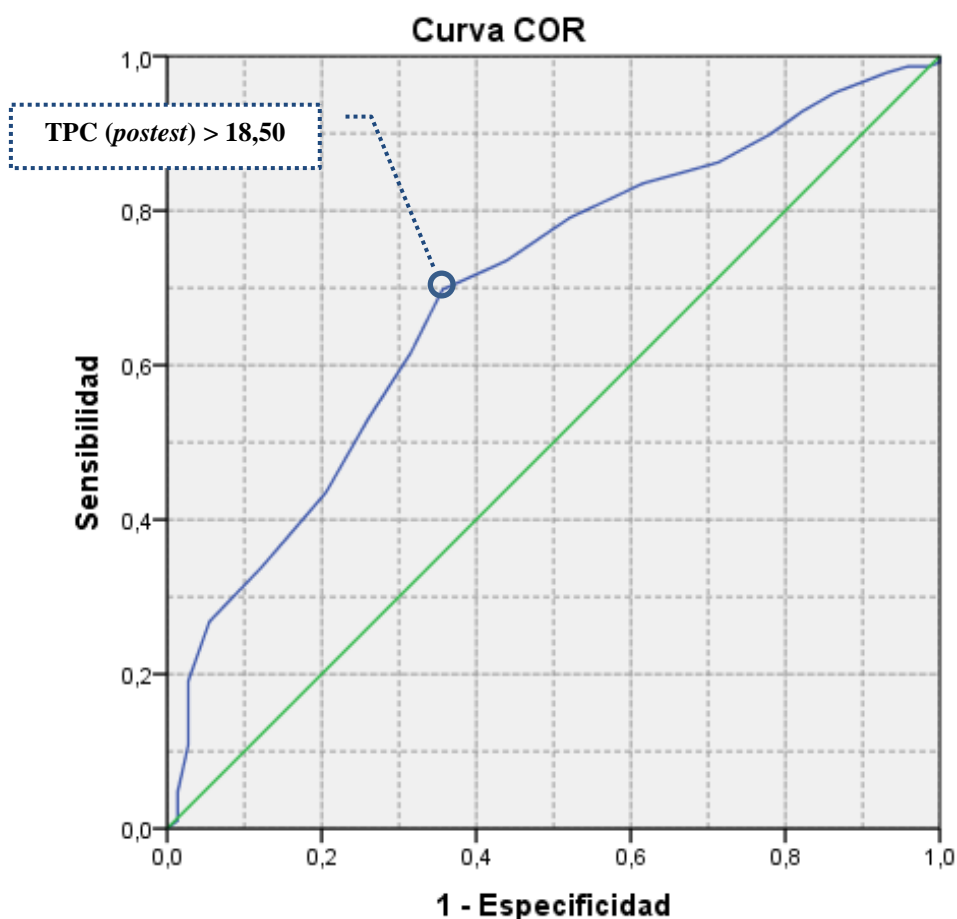


Figura 7.18. Curva COR de potencia diagnóstica del TPC (*postest*) para discriminar entre sujetos que hicieron el K-8 frente al TIC Tradicional

Tabla 7.62. Área bajo la Curva COR de potencia diagnóstica del TPC (*postest*) para discriminar sujetos K-8 vs. TIC

Condición	N válido (por lista)	Área bajo la curva A_{COR}	$p(A_{COR})$
Experimental	377	,696**	,000
Control	73		

** Significativo al nivel $p < 0,01$

- Por otro lado, hay otras dos variables en las que se han encontrado mejoras cercanas al límite de significatividad en los estudiantes que siguieron el K-8 frente a los estudiantes que siguieron el currículum TIC tradicional; y con tamaños globales del efecto del diseño *cuasi-experimental* $d > 0,20$: la ‘resolución de problemas’ y el ‘control de la impulsividad’. Estos resultados nos llevan a proponer replicaciones del *cuasi-experimento* con ambas medidas. Además, son resultados que parecen tener sentido: por un lado, la ‘resolución de problemas’ medida a través del RP30 fue la variable que correlacionó más intensamente en la validación criterial concurrente de nuestro TPC ($r = +0,669^{**}$), y es esperable que ambas medidas se muestren sensibles al mismo tipo de cursos-programas; por otro lado, el ‘control de la impulsividad’ va en la línea de nuestra fundamentación teórica que defiende la codigoalfabetización como antídoto a un uso impulsivo, poco reflexivo, de lo digital.

Con respecto al grupo de hipótesis específicas $H_{4k} \dots H_{4o}$, todas ellas ya en el marco directivo de *viabilidad* y relativas al cuestionario de profesores, podemos concluir que:

- Globalmente, los profesores señalan la *viabilidad* del K-8: más del 50% de los profesores informan de que en los grupos de clase experimentales se han dedicado al menos 8 semanas lectivas al curso K-8 (H_{4k} aceptada); más del 50% de los profesores informan de que en los grupos de clase experimentales se han impartido al menos el 80% de las 20 etapas que componen el curso K-8 (H_{4l} aceptada); los profesores consideran que el grado de aprendizaje de sus estudiantes con el curso K-8 ha sido de al menos 4 sobre 5 (H_{4n} aceptada), y muestran una satisfacción global con respecto al mismo también en un grado de al menos 4 sobre 5 ($H_{4ñ}$ aceptada); y finalmente más del 80% de los profesores (de hecho el 100%) manifiesta su intención de impartir el curso K-8 de nuevo en el próximo curso académico (H_{4o} aceptada).
- La única hipótesis específica rechazada de este grupo es la H_{4m} , dado que sólo el 76,9% de los profesores ($< 80\%$ que fue fijado como criterio de aceptación estadístico) informan de que en los grupos de clase experimentales se hayan realizado tanto etapas ‘*on line*’ como etapas ‘*unplugged*’. Ello enlaza con los 3 puntos débiles principales del K-8 señalados por los profesores en sus respuestas abiertas, a saber: ciertos problemas de temporalización del curso que llevan a prescindir de algunas actividades ‘*unplugged*’ (algunas de ellas son consideradas poco motivadoras y/o excesivamente sencillas para 2º ESO); entorno de aprendizaje ‘*on-line*’ basado en niveles-retos excesivamente repetitivos; y déficit del K-8 para el fomento de la creatividad. Frente a estos 3 puntos débiles pueden contraponerse los 3 puntos fuertes principales del K-8, señalados por los profesores en sus respuestas abiertas: el K-8 permite la autonomía de aprendizaje de los alumnos, atendiendo a distintos ritmos de avance por el curso; es un entorno gamificado y visual que mantiene alta la motivación; y promueve no sólo el pensamiento computacional sino también trabajo en equipo, concentración, tolerancia al error y, transversalmente, competencia en lengua inglesa.

Con respecto al grupo de hipótesis específicas $H_{4p} \dots H_{4u}$, todas ellas en el marco directivo de *viabilidad* y relativas al cuestionario de estudiantes, podemos concluir que:

- Globalmente, los estudiantes señalan la *viabilidad* del K-8: más del 50% de los estudiantes completan al menos el 80% de los 98 niveles-pantallas ‘*on-line*’ que componen el curso K-8

(H_{4p} aceptada); los estudiantes manifiestan una percepción de autoeficacia en ‘coding’ y motivación por continuar aprendiendo ‘coding’ tras el curso K-8 en un grado de al menos 4 sobre 5 (H_{4q} y H_{4r} aceptadas); los estudiantes consideran que su grado de aprendizaje con el curso K-8 ha sido de al menos 4 sobre 5 (H_{4t} aceptada), y su valoración global del mismo también se sitúa en un grado de al menos 4 sobre 5 (H_{4u} aceptada)

- La única hipótesis específica rechazada de este grupo es la H_{4s} , dado que sólo el 78,3% de los estudiantes (< 80% que fue fijado como criterio de aceptación estadístico) manifiesta su intención de continuar aprendiendo ‘coding’ a través de plataformas-tutoriales en Internet tras el curso K-8. Estos resultados son consistentes con los ya encontrados en el estudio exploratorio de ‘La Hora del Código’ (Capítulo 5), en el que vimos que alrededor de un 20% de sujetos (fundamentalmente chicas), incluso teniendo buena percepción de autoeficacia y motivación con el ‘coding’, no desean aprenderlo exclusivamente a través de estos entornos ‘on-line’ autoguiados.
- Adicionalmente, los análisis diferenciados por sexo sobre las variables involucradas en las anteriores hipótesis muestran que no hay diferencias en rendimiento objetivo en el K-8 medido a través del número de ‘niveles-pantallas’ completados; pero sí aparecen diferencias significativas a favor de los chicos en motivación por continuar aprendiendo ‘coding’ tras el curso K-8, y en la valoración global del curso. Estos resultados van también en la línea de los obtenidos en el estudio exploratorio de ‘La Hora del Código’ en el que ya vimos cómo el aprendizaje del ‘coding’, especialmente en estos entornos ‘close-ended’, penalizan diferencialmente la motivación de las chicas, que parecen necesitar otros contextos más creativos, narrativos y cercanos a sus intereses para practicar la lectoescritura de código.

Globalmente, los resultados relativos a las hipótesis específicas dentro de la directiva de *adecuación y viabilidad*, son consistentes con la otra evaluación del curso K-8 de la cual hemos tenido noticia (Kalelioğlu, 2015). En este trabajo, un *diseño pre-experimental de un solo grupo con pretest y postest*, el autor reporta que los estudiantes que siguieron el curso: no mejoraron significativamente sus habilidades de solución de problemas; pero sí mejoraron sus actitudes y motivación hacia la programación informática; y no se encontraron diferencias por sexo en rendimiento objetivo en el K-8 medido a través del número de ‘niveles-pantallas’ completados.

Por último, a modo especulativo y más allá de las hipótesis específicas enunciadas, presentamos un diagrama de dispersión (Figura 7.19) que ilustra la correlación entre las dos medidas de ‘tracking’ que aporta la plataforma de Code.org: ‘niveles-pantallas completados’ (que hemos usado para el contraste de la hipótesis H_{4p}) y ‘número de líneas de código escritas’. Tal y como podría esperarse, la correlación entre ambas medidas es positiva, de intensidad alta y estadísticamente muy significativa ($r = +0,640$; $p_{(r)} = 0,000 < 0,01$). Sin embargo, la relación no es perfectamente lineal. Ello es debido a que un mismo nivel se puede completar con un número variable de líneas de código (p.e. al utilizar un ‘bucle’ de repetición, el programa resultante tiene un menor número de líneas de código, aunque de hecho ejecute las mismas instrucciones). Volviendo a la Figura 7.19, nos preguntamos para estudios futuros qué caracteriza al grupo de sujetos que se apartan notoriamente de la recta de regresión; estos sujetos tienen un estilo ‘ineficiente’ de escritura de código dado que, para superar el mismo número de niveles que sus iguales, necesitan un número de líneas de código mucho mayor.

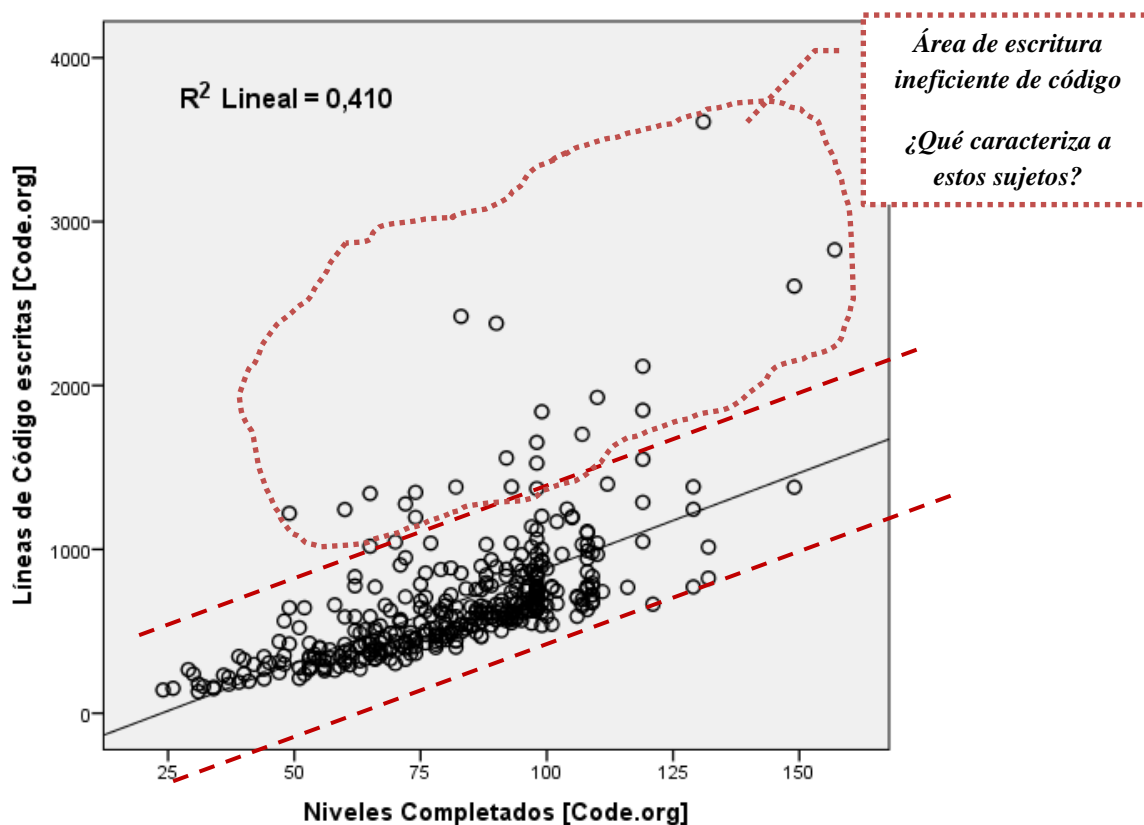


Figura 7.19. Diagrama de dispersión 'Niveles completados' * 'Líneas de código escritas' en Code.org

Finalmente, con respecto a la hipótesis específica H_{4v} , dentro de la directiva de *relevancia*:

- Hay un consenso generalizado por parte de los estudiantes de que el 'coding' es relevante: tras el curso K-8, más del 80% de los estudiantes considera importante el 'coding' para su formación como ciudadano del siglo XXI (H_{4v} aceptada)
- Sin embargo, el análisis cualitativo del contenido de las razones que esgrimen los estudiantes para sostener la afirmación anterior nos devuelve un panorama algo preocupante. Partiendo de la base de que todos los tipos de razones tienen su peso específico, llama poderosamente la atención la baja frecuencia de aparición de razones del tipo 'expresivo-comunicativas' y 'emotivas'. Parece que, tras el curso K-8, los estudiantes tienen claro que el 'coding' les puede resultar útil para encontrar un trabajo, que es una habilidad fundamental para un mundo cada vez más invadido por lo tecnológico-digital, e incluso que puede contribuir a desarrollar sus habilidades mentales; sin embargo, el curso K-8 no parece haber transmitido tan eficazmente a los estudiantes la potencialidad del 'coding' como vehículo de creación, expresión y comunicación de ideas, como una tarea divertida y de disfrute que, además, genera una mayor conciencia crítica con respecto a nuestra relación con las máquinas. Ello es consistente con el contexto de aprendizaje 'close-ended', poco creativo, que propone el curso K-8; y que nos lleva a postular que debe ser complementado por contextos tipo 'open-ended' para que el sujeto sea códigoalfabetizado de manera completa.

Con estas conclusiones parciales, damos por finalizada la evaluación principal sobre el curso 'K-8 Intro to Computer Science'. Ahora damos paso a una evaluación complementaria, de mucha menor profundidad, que confiamos sirva para contextualizar mejor las anteriores conclusiones.

7.2. Evaluación complementaria: el curso ‘K-5 Computer Science Fundamentals (Course 2)’ de Code.org

7.2.1. Antecedentes

El currículum ‘K-5 Computer Science Fundamentals’⁴⁰⁷ es una adaptación del ‘K-8 Intro to Computer Science’, que propone Code.org para sujetos ‘elementary school’ (equivalente a nuestro 1º-5º Primaria). El K-5 está compuesto internamente por cuatro cursos: ‘Course 1’⁴⁰⁸ (diseñado para niños de Primaria pre-lectores), ‘Course 2’⁴⁰⁹, ‘Course 3’⁴¹⁰ y ‘Course 4’⁴¹¹ (éstos últimos diseñados para niños de Primaria ya con habilidades lectoras).

Para la realización del ‘Course 3’ se recomienda haber realizado previamente el ‘Course 2’; y para la realización del ‘Course 4’ se recomienda haber realizado previamente el ‘Course 3’. Por tanto, para la implantación que aquí presentamos y evaluamos, circunscrita a último ciclo de Primaria, se optó por focalizarnos en el ‘Course 2’.

El itinerario curricular del K-5 (‘Course 2’) se estructura de manera análoga al que ya vimos para el K-8: una mezcla de etapas ‘on-line’ y ‘unplugged’, diseñadas para ser cubiertas en una sesión de clase cada etapa, que se van alternando hasta completar el itinerario. En concreto, el K-5 se compone de un total de 19 etapas: 11 etapas ‘on-line’ y 8 etapas ‘unplugged’. Las etapas ‘on-line’ están a su vez internamente compuestas por ‘niveles-pantallas’, a modo de retos-puzles de ‘coding’ que el alumno debe resolver; resultando un total de 151 ‘niveles-pantallas’ en el K-5 (‘Course 2’). En la siguiente Tabla 7.63 se resume la estructura y contenido del mismo (Code.org, 2015d, p. 88-89)

Tabla 7.63. Estructura y contenido del itinerario curricular del K-5 (‘Course 2’)

#	Nombre de la Etapa	Tipo de Etapa	Descripción
1	‘Graph Paper Programming’ (‘Programación con papel cuadrículado’)	‘Unplugged’	Los alumnos escriben un algoritmo (un conjunto de instrucciones) utilizando un conjunto predefinido de órdenes, con el objetivo de conseguir que sus compañeros reproduzcan un dibujo.
2	‘Real-Life Algorithms’ (‘Algoritmos de la vida real: Aviones de papel’)	‘Unplugged’	Esta etapa llama la atención acerca de cómo utilizamos los algoritmos en la vida cotidiana (p.e. instrucciones para construir un avión de papel) También aborda el lugar de los algoritmos dentro del conjunto de las Ciencias de la Computación.
3	‘Maze: Sequence’ (‘El Laberinto: secuencias’)	‘On-line’	Los estudiantes escriben programas (algoritmos destinados a un ordenador) que llevan a un personaje a través de un laberinto. Comprenderán la importancia de secuenciar las instrucciones en los programas que escriben.
4	‘Artist: Sequence’ (‘El Artista: secuencias’)	‘On-line’	Los estudiantes escriben programas para dibujar distintas líneas y formas.

⁴⁰⁷ Más información en: <https://code.org/educate/k5>

⁴⁰⁸ <https://studio.code.org/s/course1>

⁴⁰⁹ <https://studio.code.org/s/course2>

⁴¹⁰ <https://studio.code.org/s/course3>

⁴¹¹ <https://studio.code.org/s/course4>

#	Nombre de la Etapa	Tipo de Etapa	Descripción
5	'Getting Loopy' (‘Volviéndonos bucles’)	'Unplugged'	Esta etapa introduce el concepto computacional de los ‘bucles’ (instrucciones repetidas) a través de una actividad de baile. Los estudiantes aprenderán una coreografía sencilla y serán instruidos para repetirla.
6	'Maze: Loops' (‘El Laberinto: bucles’)	'On-line'	Los estudiantes escriben programas en el entorno ‘Laberinto’, utilizando bucles.
7	'Artist: Loops' (‘El Artista: bucles’)	'On-line'	Los estudiantes escriben programas para dibujar diferentes formas, al tiempo que identifican patrones en su código. Aprenderán sobre el concepto computacional de ‘bucle’ (instrucciones repetidas), que puede ser usado para hacer sus programas más eficientes.
8	'Bee: Loops' (‘La Abeja: bucles’)	'On-line'	Los estudiantes escriben programas en el entorno ‘La Abeja’, utilizando bucles.
9	'Relay programming' (‘Programación por relevos’)	'Unplugged'	Los estudiantes corren una carrera de relevos, en la cual van escribiendo progresivamente un algoritmo que reproduce una imagen en un papel cuadriculado.
10	'Bee: Debugging' (‘La Abeja: depuración’)	'On-line'	A los estudiantes se les presenta un programa ya escrito que contiene algún error para completar el puzle. Los estudiantes deben ‘depurar’ (= arreglar) el programa dado para que funcione correctamente.
11	'Artist: Debugging' (‘El Artista: depuración’)	'On-line'	A los estudiantes se les presenta un programa ya escrito que falla al tratar de crear un diseño geométrico dado. Los estudiantes deben depurar el programa para que funcione correctamente.
12	'Conditionals' (‘Condicionales’)	'Unplugged'	Para aprender qué son las sentencias condicionales, los estudiantes juegan a las cartas y generan reglas como “Si saco una carta roja, yo obtengo un punto” y “Si saco una carta negra, tú obtienes un punto”
13	'Bee: Conditionals' (‘La Abeja: condicionales’)	'On-line'	Los estudiantes escriben programas en el entorno ‘La Abeja’, utilizando sentencias condicionales.
14	'Binary Bracelets' (‘Pulseras binarias’)	'Unplugged'	Los estudiantes hacen pulseras de abalorios a partir de una plantilla de papel, que es una representación en código binario de la inicial de su nombre. Los estudiantes aprenden que un mismo conjunto de datos puede ser representado en más de una manera.
15	'The Big Event' (‘El gran evento’)	'Unplugged'	Se introduce a los estudiantes en el concepto computacional de ‘eventos’, que son las acciones cuyo estado es vigilado constantemente por el ordenador. El profesor pulsará botones sobre un control remoto falso (de juguete), y los estudiantes tendrán que gritar distintas frases dependiendo del botón pulsado.
16	'Flappy' (‘Flappy’)	'On-line'	Aplicando el concepto computacional de ‘eventos’, los estudiantes crearán su propio juego con eventos del tipo “Cuando se clicla el ratón, el pájaro aletea” y “Cuando el pájaro toca el suelo, el juego se acaba”
17	'Play Lab: Create a Story' (‘Laboratorio de Juegos: Crea una historia’)	'On-line'	Los estudiantes emplean todos los conceptos computacionales aprendidos a lo largo del itinerario curricular, para crear una narración digital, personalizada e interactiva.
18	'Your Digital Footprint' (‘Tu huella digital’)	'Unplugged'	Los profesores introducen a los estudiantes acerca de cómo el hecho de subir información personal a la red, genera una huella digital o ‘traza’, que tiene consecuencias.

#	Nombre de la Etapa	Tipo de Etapa	Descripción
19	'Artist: Nested Loops' (‘El Artista: bucles anidados’)	'On-line'	Los estudiantes escriben programas que dibujan bellos patrones, utilizando bucles anidados.

La implantación típica del K-5 (*‘Course 2’*) en el aula, al igual que el K-8, es de 2 etapas semanales; por tanto, el itinerario curricular completo suele llevar alrededor de 10 semanas lectivas (en conjunto, un trimestre escolar). También hay que señalar que el currículum K-5 fue desarrollado por Code.org con posterioridad al K-8, y ya da respuesta a una de las principales críticas a la que fue sometido éste último: el déficit de creatividad. Para ello, el K-5 incluye explícitamente el concepto computacional de ‘control de eventos’ (etapa 15), que es aplicado para la creación de juegos (etapa 16) y de narraciones digitales interactivas (etapa 17).

7.2.2. Hipótesis específicas

En este apartado 7.2 estamos abordando el objetivo general **O₅** [‘Evaluar el programa-curso *‘K-5 (Course 2)’*, desde el marco fijado por nuestras hipótesis directivas de investigación’]. Así pues, las hipótesis específicas, que enunciaremos a continuación, están enmarcadas en las hipótesis directivas que fijamos en el [apartado 2.6](#) de la parte teórica:

- En el marco de la hipótesis directiva de *adecuación*, nos planteamos la siguiente hipótesis específica:
 - **H_{5a}**: La aptitud ‘pensamiento computacional’ de los estudiantes mejora del momento anterior al posterior de realización del K-5 (*‘Course 2’*)
- En el marco de la hipótesis directiva de *viabilidad*, nos planteamos la siguiente hipótesis específica:
 - **H_{5b}**: Más del 50% de los estudiantes completan al menos el 80% de los 151 niveles-pantallas *‘on-line’* que componen el K-5 (*‘Course 2’*)

En este estudio, de menor profundidad, no nos planteamos ninguna hipótesis específica en la directiva de *relevancia*.

7.2.3. Método

Aplicamos metodologías distintas, aunque complementarias entre sí, en función del tipo de hipótesis específica sometida a contraste:

- ✓ La hipótesis específica enunciada en el marco de la directiva de *adecuación* es contrastada a través de un diseño *pre-experimental de un solo grupo con pretest y postest* (Colás y Buendía, 1992); que sólo incluye grupo de tratamiento (no hay grupo de control)
- ✓ La hipótesis específica enunciada en el marco de la directiva de *viabilidad* es contrastada a través de una metodología *descriptivo-cuantitativa*, apoyada en indicadores objetivos de seguimiento (*‘tracking’*) del curso K-5 (Code.org, 2015b)

7.2.3.1. Muestra

Para esta investigación contamos con un total de 51 sujetos, todos ellos alumnos del centro privado Yago School⁴¹² de Sevilla. De ellos, hay 33 alumnos (64,7%) de 5º de Primaria, y 18 alumnos (35,3%) de 6º de Primaria. Hay 23 chicos (45,1%) frente a 28 chicas (54,9%). Todos ellos son sujetos experimentales (no hay sujetos control) que realizan el K-5 (*'Course 2'*) a través de un dispositivo móvil tipo tableta (un *iPad* por cada alumno), a lo largo del 2º trimestre del curso 2014/2015 durante la clase obligatoria de TIC que imparte el centro (2 horas semanales)

El procedimiento de muestreo es no probabilístico e intencional. El centro contacta con este doctorando tras su comunicación en el *'I Simposio Internacional Mobile Learning'* celebrado en Córdoba en marzo de 2014 (Román-González, 2014c)

7.2.3.2. Instrumentos

Para el contraste de la hipótesis específica H_{5a} sobre la aptitud 'pensamiento computacional' utilizamos nuestro Test de Pensamiento Computacional (TPC) (Román-González, 2014i); dado que en su proceso previo de validación, expuesto en el Capítulo 6, se incluye también población escolar de 5º y 6º de Primaria.

Para el contraste de la hipótesis específica H_{5b} se descargan los datos de seguimiento (*'tracking'*) de los estudiantes relativos a su desempeño en el K-5 (*'Course 2'*) ('niveles superados' y 'líneas de código escritas'), que proporciona la plataforma de Code.org.

7.2.3.3. Procedimiento

En la primavera de 2014, tras la comunicación presentada por el doctorando al *'I Simposio Internacional Mobile Learning'* (Román-González, 2014c), el centro privado Yago School se muestra interesado en realizar una implantación piloto de un currículum de *'coding'* en sus grupos de Primaria. Durante el 1º trimestre del curso 2014/2015 se acuerda con la coordinadora TIC de Primaria la implantación piloto del K-5 (*'Course 2'*) de Code.org, a lo largo del 2º trimestre del curso, en todos los grupos de 5º y 6º de Primaria del centro (2 grupos de 5º, más 1 grupo de 6º).

Todos los grupos serán, por tanto, grupos experimentales (no hay grupos de control). La implantación se realiza en el contexto de la clase obligatoria de TIC (2 horas semanales) que oferta el centro; lo cual supone un enriquecimiento curricular en aula ordinaria sobre codigoalfabetización y desarrollo del pensamiento computacional.

Justo al comienzo del 2º trimestre, antes de arrancar con el K-5 (*'Course 2'*), se realizan las medidas psicométricas en situación *pretest* con nuestro TPC en todos los grupos experimentales⁴¹³. Análogamente, justo al finalizar el 2º trimestre, una vez realizado el K-5 (*'Course 2'*) se realizan las medidas psicométricas en situación *posttest* con nuestro TPC, y se descargan las estadísticas de seguimiento (*'tracking'*) con el desempeño en el K-5 de los estudiantes. En la siguiente Figura 7.20

⁴¹² <http://yagoschool.com/>

⁴¹³ Adicionalmente, en estos grupos se tomaron en situación *pretest* medidas con el BFQ-NA. Estos datos fueron utilizados para la validación criterial concurrente del TPC que se presentó en el Capítulo 6 (sub-epígrafe 6.5.2.3.1.1); pero no para la evaluación del K-5 (*'Course 2'*), que ahora nos ocupa.

se diagrama el diseño de investigación seguido: diseño *pre-experimental de un solo grupo con pretest y posttest* (Colás y Buendía, 1992)

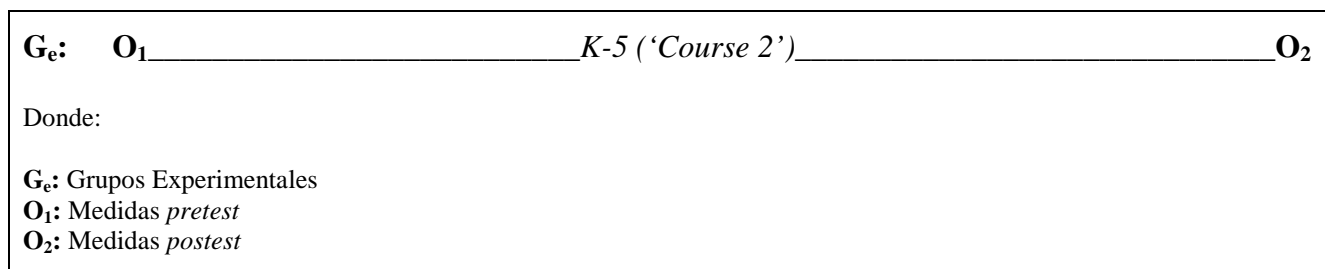


Figura 7.20. Diagrama del diseño pre-experimental seguido para la evaluación del K-5 ('Course 2')

Este tipo de diseño es más débil que el *cuasi-experimental* utilizado para la evaluación del curso K-8, pues no permite asegurar que los cambios producidos entre el *pretest* y el *posttest* sean debidos al K-5 (hay explicaciones alternativas plausibles no controladas en el diseño, como la maduración de los sujetos o el efecto del entrenamiento con las pruebas). Por ello, la hipótesis específica **H_{5a}** ha sido enunciada en términos menos ambiciosos que su análoga de la evaluación K-8.

7.2.4. Resultados

Dando respuesta a la hipótesis específica **H_{5a}**, en la Tabla 7.64 se detalla la prueba *t* de diferencia de medias para muestras emparejadas entre el TPC (*pretest*) y el TPC (*posttest*). Se encuentra una diferencia estadísticamente muy significativa ($t = -6,261$; $p_{(t)} = 0,000 < 0,01$), y de tamaño 'grande' ($d = 0,80$), a favor de la situación *posttest*. Por tanto, **se acepta la hipótesis específica H_{5a}**.

Tabla 7.64. Prueba *t* para muestras relacionadas en el TPC

	N	Media	Desviación estándar	<i>r</i>	$p_{(r)}$	<i>t</i>	$p_{(t)}$	<i>d</i> de Cohen
TPC (<i>pre</i>)	51	13,55	5,423	,581**	,000	-6,261**	,000	0,80
TPC (<i>post</i>)	51	18,00	5,657					

** Significativo al nivel $p < 0,01$

Segmentando el análisis por curso académico, vemos que la diferencia de medias es muy significativa en 5° de Primaria ($p_{(t)} = 0,000 < 0,01$; $d = 0,88$), y sólo significativa en 6° Primaria ($p_{(t)} = 0,020 < 0,05$; $d = 0,70$). Se detalla en la siguiente Tabla 7.65.

Tabla 7.65. Prueba *t* para muestras relacionadas en el TPC, segmentada por curso académico

		N	Media	Desviación estándar	<i>r</i>	$p_{(r)}$	<i>t</i>	$p_{(t)}$	<i>d</i> de Cohen
5° Primaria	TPC (<i>pre</i>)	33	12,79	5,110	,727**	,000	-6,819**	,000	0,88
	TPC (<i>post</i>)	33	17,39	5,379					
6° Primaria	TPC (<i>pre</i>)	18	14,94	5,846	,341	,166	-2,570*	,020	0,70
	TPC (<i>post</i>)	18	19,11	6,135					

* Significativo al nivel $p < 0,05$

** Significativo al nivel $p < 0,01$

Dando respuesta a la hipótesis específica **H_{5b}**, en la Tabla 7.66 se detallan las estadísticas de 'niveles-pantallas' completados por los sujetos experimentales en la plataforma Code.org. Según decíamos algo más arriba, el K-5 ('Course 2') está compuesto por 151 'niveles-pantallas'; por tanto,

el punto de corte para nuestra decisión estadística se sitúa en $0,8 \cdot 151 = 120$ ‘niveles-pantallas’ superados. Tal y como puede observarse en la Tabla 7.66, más del 50% de los estudiantes completan al menos el 80% de los 151 niveles-pantallas ‘on-line’ que componen el K-5 (‘Course 2’) (Mediana = 134 > 120); por tanto, **se acepta la hipótesis específica H_{5b}**. Segmentando por curso académico, comprobamos que la hipótesis se cumple en 6º Primaria, no así en 5º de Primaria. Adicionalmente, comprobamos que hay aproximadamente un 15% de la muestra total que supera los 151 ‘niveles-pantallas’: son sujetos que, una vez completado el 100% de los niveles K-5 (‘Course 2’), hicieron por su cuenta niveles adicionales en alguno de los otros entornos de programación que ofrece la plataforma Code.org.

Tabla 7.66. Estadísticas de ‘niveles-pantallas’ completados en Code.org por los sujetos K-5

		Niveles Completados [Code.org]		
		Muestra Total K-5	5º Primaria	6º Primaria
N	Válido	50	32	18
	Perdidos	1	1	0
Media		120,80	109,28	141,28
Mediana		134,00	115,50	142,00
Moda		144 ^a	152	144
Desviación estándar		35,370	38,607	13,940
Asimetría		-,994	-,464	-,109
Mínimo		33	33	106
Máximo		173	158	173
Percentiles	10	65,10	45,90	124,90
	20	87,00	69,60	134,00
	30	109,50	85,70	137,40
	40	125,20	105,60	139,20
	50	134,00	115,50	142,00
	60	139,20	128,00	144,00
	70	144,00	137,00	144,30
	80	148,80	150,80	147,20
	90	156,50	155,50	164,90

a. Existen múltiples modas. Se muestra el valor más pequeño.

7.2.5. Discusión y conclusiones parciales

Con respecto a la hipótesis H_{5a}, en el marco de la directiva de *adecuación*, podemos concluir que:

- La aptitud ‘pensamiento computacional’ de los estudiantes mejora del momento anterior al posterior de realización del K-5 (‘Course 2’); es decir, se acepta H_{5a}. Ahora bien, dicha mejora no puede atribuirse causalmente al K-5 (‘Course 2’) dado que el diseño no controla posibles explicaciones alternativas, especialmente el efecto del entrenamiento con el TPC.

- Dicho lo anterior, y dado que el tamaño del efecto entre *pretest-postest* del TPC en los sujetos K-5 es ‘grande’ ($d = 0,80$), y de magnitud similar al encontrado en los grupos experimentales de la evaluación K-8 ($d = 0,87$) en donde sí se pudo probar la causalidad de dicho curso en la mejora del pensamiento computacional de los estudiantes de Secundaria; es muy plausible considerar que el K-5 ha ejercido una mejora análoga sobre los estudiantes de Primaria. En cualquier caso, es una hipótesis que deberá ser sometida a contraste mediante el diseño *cuasi-experimental* correspondiente.
- En este punto, consideramos de gran interés comparar los tamaños del efecto *pretest-postest* con el TPC en los 4 tipos de sujetos en que se ha aplicado este par de medidas, a lo largo de esta tesis doctoral; a saber: el grupo de sujetos experimentales K-8; el grupo de sujetos control K-8 (que siguieron el currículum TIC tradicional); el grupo de sujetos experimentales K-5; y el grupo de sujetos que participaron en la validación convergente del TPC con Dr. Scratch (que trabajaron en la plataforma Scratch durante el intervalo *pre-post*). En los 4 tipos, el intervalo de aplicación *pretest-postest* del TPC fue similar (10-12 semanas). En la Tabla 7.67 se muestran los tamaños del efecto, ordenados de mayor a menor.

 Tabla 7.67. Comparación de los tamaños del efecto *pre-post* del TPC, en función del curso-programa seguido

Currículum seguido	Número de sujetos	Etapas educativa de los sujetos	Contexto de aprendizaje del ‘coding’	Tamaño del efecto <i>pretest-postest</i> con el TPC (d de Cohen)
K-8 de Code.org	332	Secundaria	‘Close-ended’	0,87**
K-5 (‘Course 2’) de Code.org	51	Primaria	‘Close-ended’	0,80**
Scratch ⁴¹⁴	71	Secundaria	‘Open-ended’	0,37**
TIC tradicional	54	Secundaria	No se aprende ‘coding’	0,30**

** La prueba t para muestras emparejadas *pretest-postest* resultó significativa al nivel $p_{(t)} < 0,01$

Tal y como puede verse en la tabla anterior (y Figura 7.21) los tamaños del efecto *pretest-postest* con el TPC son mayores tras un aprendizaje del ‘coding’ en contextos ‘close-ended’, que tras un aprendizaje del ‘coding’ en un contexto ‘open-ended’; y mayor en éste último que tras un aprendizaje TIC tradicional (que no incluye el ‘coding’ en sus contenidos). Estos resultados son consistentes, dado que los contextos ‘close-ended’ trabajan sistemáticamente los conceptos computacionales, de manera que sirven de entrenamiento para una prueba de elección múltiple como el TPC. Cabe entonces preguntarse si es que la plataforma Code.org promueve el pensamiento computacional en mayor medida que la plataforma Scratch; o si, más bien, el TPC es una medida de pensamiento computacional más sensible al aprendizaje realizado en contextos como Code.org (‘close-ended’) que al realizado en contextos como Scratch (‘open-ended’). Dada esta posible dependencia del TPC con respecto al tipo de contexto de aprendizaje del ‘coding’, parece aconsejable que, para futuras evaluaciones de programas de códigoalfabetización y desarrollo del pensamiento computacional, se utilicen, junto al TPC, otros instrumentos de medida (p.e. Dr. Scratch o Tareas Bebras).

⁴¹⁴ No se siguió estrictamente ningún currículum estructurado-sistemático con Scratch, aunque sí se aconsejó utilizar la guía ‘Creative Computing’ (Brennan, Balch, & Chung, 2014)

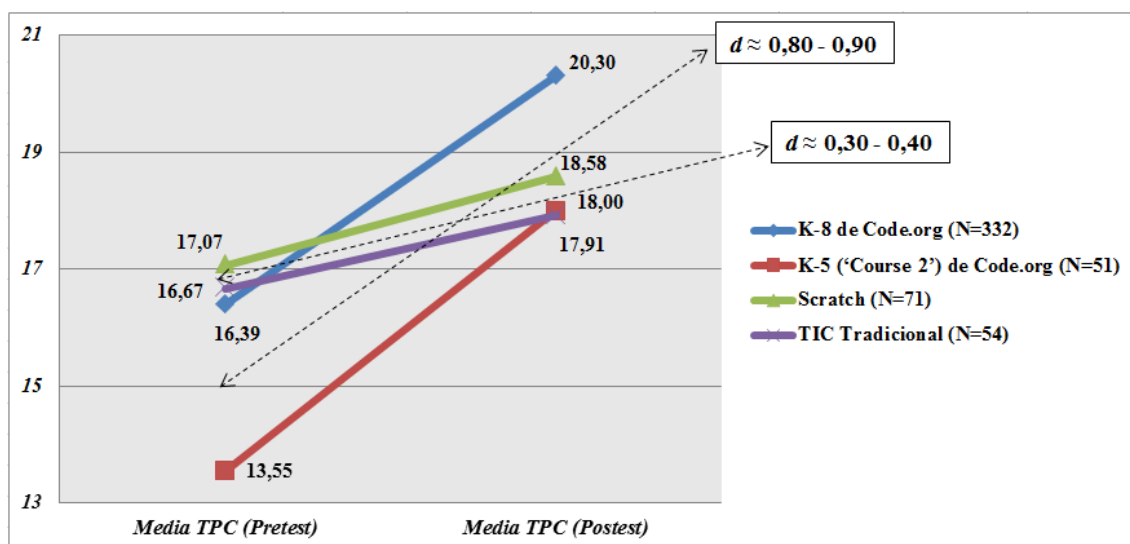


Figura 7.21. Comparación de las medias TPC *pretest-postest*, en función del curso-programa seguido

Con respecto a la hipótesis específica H_{5b} , en el marco de la directiva de *viabilidad*, podemos concluir que:

- La implantación del K-5 ('Course 2') parece globalmente viable en último ciclo de Educación Primaria, dado que, en nuestra evaluación, más del 50% de los estudiantes completan al menos el 80% de los 151 niveles-pantallas 'on-line' que componen el curso (H_{5b} aceptada). Ahora bien, si dicha viabilidad parece garantizada en 6º Primaria, ofrece algunas dudas en 5º de Primaria, en donde la mediana de 'niveles-pantallas' completados no llegó al punto de corte establecido.
- Finalmente, cabe destacar que vuelve a aparecer una cola derecha en la distribución de 'niveles-pantallas' completados, correspondiente aproximadamente a un 15% de sujetos, que traspasan el techo fijado por el K-5 ('Course 2') para adentrarse por su cuenta en otros entornos de programación que ofrece la plataforma Code.org. En este sentido, valoramos muy positivamente las virtudes de este tipo de plataformas para permitir un avance, acaso sin límite, de los sujetos por sucesivos grados de dominio.

Para finalizar este capítulo, pasamos a dar cuenta del tercer sub-estudio, éste centrado en un par de casos de sujetos con 'alta capacidad computacional', que esperamos sirva para cerrar el (amplio) círculo abierto al comienzo de esta tesis.

7.3. Estudio de casos de alta capacidad computacional en el curso 'Computer Programming' de Khan Academy

7.3.1. Antecedentes

El curso 'Computer Programming'⁴¹⁵ (Khan Academy, 2015) y, en particular, su módulo 'Intro to JS: Drawing & Animation'⁴¹⁶, ya fueron descritos en el Capítulo 1 de esta tesis doctoral (epígrafe

⁴¹⁵ <https://www.khanacademy.org/computing/computer-programming>

⁴¹⁶ <https://www.khanacademy.org/computing/computer-programming/programming>

1.2.1) como ejemplos de contextos de codigoalfabetización especialmente indicados para los más capaces. Baste aquí recordar que el módulo *'Intro to JS: Drawing & Animation'* está compuesto por un itinerario formativo perfectamente delimitado en el que se distinguen 3 tipos de elementos: tutoriales de vídeo, tutoriales de 'consola de programación interactiva', y retos de programación.

Los tutoriales de vídeo son piezas audiovisuales alojadas en el canal de Khan Academy en YouTube; mientras que los tutoriales de 'consola de programación interactiva' son simulaciones en el mismo entorno que utilizará el estudiante para sus retos y proyectos de programación, que reproducen cómo ir utilizando los distintos comandos y parámetros del lenguaje de programación textual *'JavaScript & ProcessingJS'*

Precisamente, la principal diferencia del curso *'Computer Programming'* (Khan Academy, 2015) respecto de los anteriormente evaluados K-8 (Code.org, 2015a) y K-5 (Code.org, 2015b) es que éstos últimos funcionan con lenguajes de programación visuales por bloques, mientras que aquél lo hace ya con lenguajes de programación textuales. Ello hace del curso *'Computer Programming'* un entorno de aprendizaje *a priori* indicado para los sujetos computacionalmente más capaces de primer ciclo de la ESO que puedan y quieran avanzar hacia niveles superiores de codigoalfabetización (Severance, 2015).

7.3.2. Objetivos específicos

En este apartado 7.3 estamos abordando el objetivo general **O₆** ['Estudiar casos de sujetos con alta capacidad computacional en el contexto del programa-curso *'Computer Programming'*; desde el marco fijado por nuestras hipótesis directivas de investigación']. Así pues, los objetivos específicos, que enunciaremos a continuación, están enmarcados en las hipótesis directivas que fijamos en el apartado 2.6 de la parte teórica:

- En el marco de la hipótesis directiva de *adecuación*, nos planteamos el siguiente objetivo específico:
 - **O_{6a}**: Describir las medidas psicométricas de los sujetos-casos de 'alta capacidad computacional'
- En el marco de la hipótesis directiva de *viabilidad*, nos planteamos el siguiente objetivo específico:
 - **O_{6b}**: Describir el nivel de desempeño de los sujetos-casos de 'alta capacidad computacional' en el módulo *'Intro to JS: Drawing & Animation'* del curso *'Computer Programming'*
- En el marco de la hipótesis directiva de *relevancia*, nos planteamos el siguiente objetivo específico:
 - **O_{6c}**: Describir los productos de programación realizados por los sujetos-casos de 'alta capacidad computacional' en el módulo *'Intro to JS: Drawing & Animation'* del curso *'Computer Programming'*

7.3.3. Método

Este tercer sub-estudio sigue una metodología de *estudio de casos múltiples* (García Llamas, 2003) o *estudio de caso, múltiple* (Montero & León, 2007); apoyado en medidas de distinto tipo en función del objetivo específico abordado.

7.3.3.1. Muestra

Tal y como ya se justificó en el Capítulo 6 (sub-epígrafe 6.5.2.3.2, sobre la validez discriminante del TPC), los sujetos-casos de ‘alta capacidad computacional’ son extraídos a partir de los grupos de clase que realizaron el curso ‘K-8 Intro to Computer Science’ (Code.org, 2015a) mediante una metodología ‘self-paced’ (‘autoguiada’ por el ritmo de cada sujeto); metodología que permitió emerger y manifestarse a la ‘alta capacidad computacional’.

Más concretamente, los sujetos categorizados como de ‘alta capacidad computacional’ son aquéllos que completaron el curso K-8 a un ritmo notablemente más rápido que sus compañeros de aula, y solicitaron acelerar al curso de nivel superior que nos ocupa, el ‘Computer Programming’ (Khan Academy, 2015). Atendiendo a este criterio, se detectaron 7 sujetos de ‘alta capacidad computacional’: 2 del IES María Blasco y 5 del IES Andreu Sempere, todos ellos varones de 1º Ciclo de la ESO. Finalmente, sólo se obtuvo acceso y seguimiento de la actividad en el curso ‘Computer Programming’ de los dos sujetos del IES María Blasco (dos varones de 1º ESO), que forman la muestra final para este estudio de casos.

7.3.3.2. Instrumentos

Para el objetivo específico O_{6a} nos servimos de las medidas psicométricas aplicadas a los dos sujetos-casos en el marco de la evaluación *cuasi-experimental* del K-8, a saber, el TPC y el RP30, ya descritos anteriormente en esta tesis doctoral (sub-epígrafe 6.5.1.2). Para el objetivo específico O_{6b} se descargan los datos de seguimiento (‘tracking’) aportados por la plataforma Khan Academy, sobre el módulo ‘Intro to JS: Drawing & Animation’ del curso ‘Computer Programming’. Para el objetivo específico O_{6c} se analiza uno de los proyectos realizados por cada uno de los sujetos-casos, que fueron compartidos en la galería pública de código de Khan Academy.

7.3.3.3. Procedimiento

Mediado el 2º trimestre del curso 2014/2015, en el marco de la implantación experimental del curso K-8 en el grupo de 1º ESO del IES María Blasco, que sigue una metodología ‘self-paced’ (los alumnos pueden avanzar por el curso tan rápido como su capacidad y motivación se lo permite, tanto en el aula como en casa); su profesora⁴¹⁷ nos comunica que hay 2 alumnos que han completado los 98 ‘niveles-pantallas’ del K-8 en apenas un par de semanas (cuando el tiempo estimado para ello es de, al menos, ocho semanas). La profesora nos solicita alguna sugerencia para que esos dos alumnos puedan avanzar hacia un currículum de nivel superior, que ya incluya el aprendizaje de algún

⁴¹⁷ Se da la circunstancia de que esta persona, además de ejercer docencia en Secundaria, es profesora en la Universidad de Alicante en el Departamento de Ciencias de la Computación e Inteligencia Artificial. Este particular influyó notablemente en su sensibilidad para promover la alta capacidad computacional en su grupo de clase.

lenguaje de programación textual, y entonces les proponemos cursar el módulo ‘Intro to JS: Drawing & Animation’ del curso ‘Computer Programming’ (Khan Academy, 2015).

Tanto profesora como alumnos aceptan. Creamos entonces un aula virtual asociada al módulo ‘Intro to JS: Drawing & Animation’ en la plataforma Khan Academy, y damos de alta en la misma a los dos sujetos-casos para poder monitorizar su progreso. Durante el resto del 2º trimestre, los dos sujetos-casos fueron avanzando por el módulo según una metodología ‘flipped classroom’: en su casa visualizaban los vídeos y tutoriales, y durante la hora de clase realizaban los retos y proyectos de programación pudiendo solicitar la ayuda de la profesora (sin perjuicio de poder igualmente continuar en su domicilio).

7.3.4. Resultados

Abordando el objetivo O_{6a} , en la Tabla 7.68 se detallan las puntuaciones, tanto directas (P_d) como percentiles⁴¹⁸ (P_c), de los dos sujetos-casos en las pruebas psicométricas que se les aplicaron durante la evaluación *cuasi-experimental* del K-8.

Tabla 7.68. Puntuaciones directas y percentiles de los dos sujetos-casos de alta capacidad computacional en TPC y RP30

	Test de Pensamiento Computacional (TPC)				Test de Resolución de Problemas (RP30)			
	<i>pretest</i>		<i>postest</i>		<i>pretest</i>		<i>postest</i>	
	P_d	P_c	P_d	P_c	P_d	P_c	P_d	P_c
Caso 1	27	100	27	99	42	61	50	76
Caso 2	26	99	28	100	36	44	45	59

Tal y como puede observarse en la tabla anterior, los dos sujetos-casos se sitúan en el extremo superior de la distribución en el TPC ($P_c \geq 99$), tanto en *pretest* como en *postest*. Sin embargo, en el RP30 se sitúan ‘sólo’ en percentiles medio-altos ($44 \leq P_c \leq 76$). Por otro lado, mientras que en el TPC prácticamente ya tocaron ‘techo’ en situación *pretest* (recordemos que el máximo posible en el TPC es $P_d = 28$), alcanzando dicho ‘techo’ el Caso 2 en *postest*; en el RP30, con más margen de mejora, ambos sujetos-casos progresaron del *pre* al *post*.

Abordando el objetivo O_{6b} , en la siguiente Tabla 7.69 se detalla el resumen de datos de seguimiento (‘tracking’) que aporta la plataforma Khan Academy sobre la actividad de aprendizaje de los dos sujetos-casos en el módulo ‘Intro to JS: Drawing & Animation’ del curso ‘Computer Programming’

Tabla 7.69. Estadísticas de seguimiento (‘tracking’) de los dos sujetos-casos en el módulo ‘Intro to JS: Drawing & Animation’ del curso ‘Computer Programming’

	Tiempo utilizado en visualización de videos y tutoriales (en minutos)	Tiempo utilizado en ejercicios y retos (en minutos)	Insignias digitales (‘badges’) ganadas	Puntos conseguidos
Caso 1	169,45	795,01	16	95222
Caso 2	393,41	155,81	13	78005

Tal y como puede observarse en la tabla anterior, aunque ambos casos comparten la etiqueta de ‘sujetos de alta capacidad computacional’ y la metodología ‘flipped classroom’, luego presentan un

⁴¹⁸ Todos los percentiles están referidos a las muestras de 1º ESO en las que se aplicaron el TPC y/o RP30, respectivamente en situación *pretest* y *postest* de la evaluación *cuasi-experimental* del K-8.

estilo distinto de uso de la plataforma de aprendizaje. Así, mientras que el Caso 1 emplea en realizar ejercicios y retos casi cinco veces más de tiempo que en visualizar videos y tutoriales (un estilo diríamos más activo); el Caso 2 presenta un patrón inverso, empleando más del doble de tiempo en videos y tutoriales que en ejercicios y retos (un estilo diríamos más reflexivo).

Adicionalmente, se pueden obtener gráficas detalladas de la actividad y progreso de cada estudiante a lo largo del tiempo. Por ejemplo, en la Figura 7.22 se muestra la gráfica detallada del Caso 1. Tal y como podemos observar, dicha gráfica muestra en el eje de abscisas el periodo temporal analizado, y en el eje de ordenadas el tiempo en minutos empleado en la plataforma cada día de dicho periodo; sobre la gráfica se muestra información detallada sobre videos y tutoriales visionados por el aprendiz, proyectos y retos superados, ‘badges’ y puntos conseguidos.

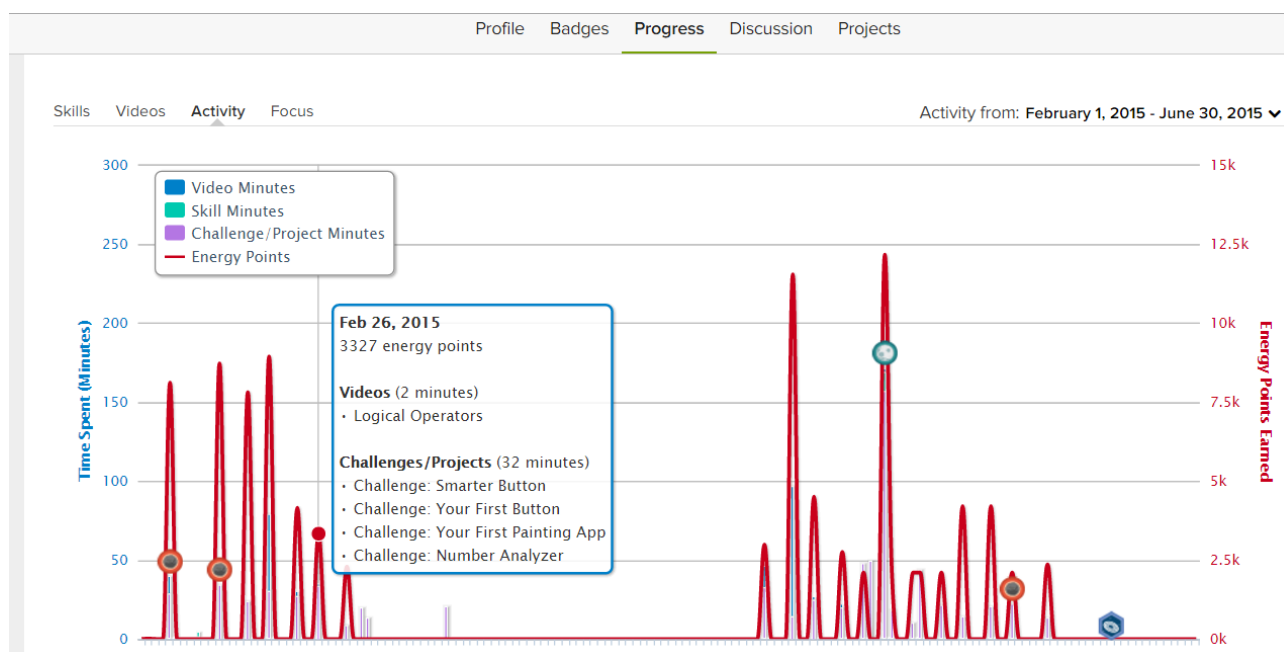


Figura 7.22. Gráfica detallada del progreso del Caso 1 en el módulo ‘Intro to JS: Drawing & Animation’ del curso ‘Computer Programming’

Tal y como puede verse en la anterior Figura 7.22, la gráfica del Caso 1 registra su actividad en la plataforma en un periodo temporal que va desde el 1 de febrero al 30 de junio de 2015. Se detectan claramente dos franjas de actividad: una, a la izquierda, que abarca del 1 de febrero al 15 de marzo (coincidiendo con la segunda mitad del 2º trimestre escolar, en el cual el Caso 1 estuvo realizando el módulo ‘Intro to JS: Drawing & Animation’ mientras sus compañeros ‘normales’ de aula ordinaria acababan el K-8); y otra, a la derecha, que abarca del 1 de mayo al 15 de junio (en la que el Caso 1 siguió avanzando por el módulo de manera independiente al contexto escolar).

En la Figura 7.23, presentamos la gráfica de actividad del Caso 2 correspondiente al mismo periodo temporal (del 1 de febrero al 30 de junio de 2015). Tal y como puede observarse, la actividad del Caso 2 se concentra casi exclusivamente en la segunda mitad del 2º trimestre escolar (a la izquierda de la gráfica); no apareciendo posteriormente casi ningún rastro de actividad del Caso 2 en la plataforma.

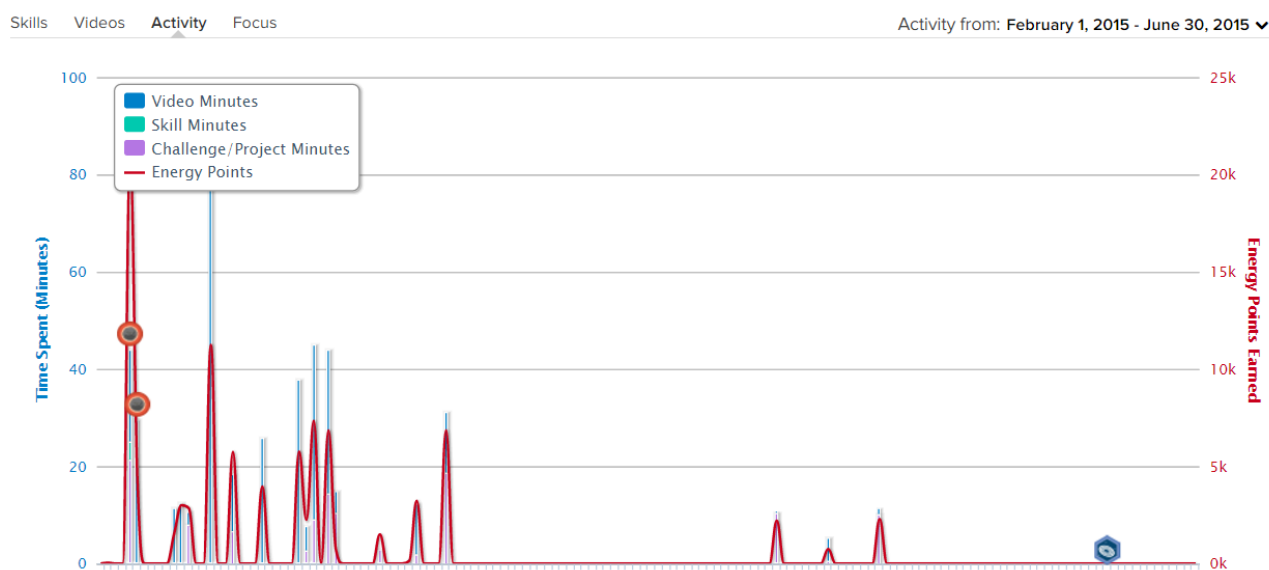


Figura 7.23. Gráfica detallada del progreso del Caso 2 en el módulo 'Intro to JS: Drawing & Animation' del curso 'Computer Programming'

Abordando el objetivo O_{6c} , en la Figura 7.24 se ilustra uno de los productos-proyectos⁴¹⁹ de programación publicados por el Caso 1 en la galería de código de Khan Academy. Tal y como puede observarse en la figura, el Caso 1 ha sido capaz de escribir un programa de 10 líneas de código en lenguaje textual 'JavaScript & ProcessingJS' que incluye: definición inicial de tres variables [posición inicial del centro del círculo en el eje x ('xPos'), posición inicial del centro del círculo en el eje y ('yPos'), y radio inicial del círculo ('e')]; una función ('function') que incluye dichos parámetros-variables, además de los códigos RGB de color del fondo ('background') y de color del círculo ('fill'); y un bucle que repite indefinidamente esta función ('draw' = 'function') con un incremento del radio del círculo ('e+=1') en cada iteración. Tal y como puede intuirse, al ejecutar el programa se genera el efecto animado de que el círculo se va acercando al espectador (recomendamos vivamente visualizar el programa en acción, pinchando sobre enlace al pie de figura)

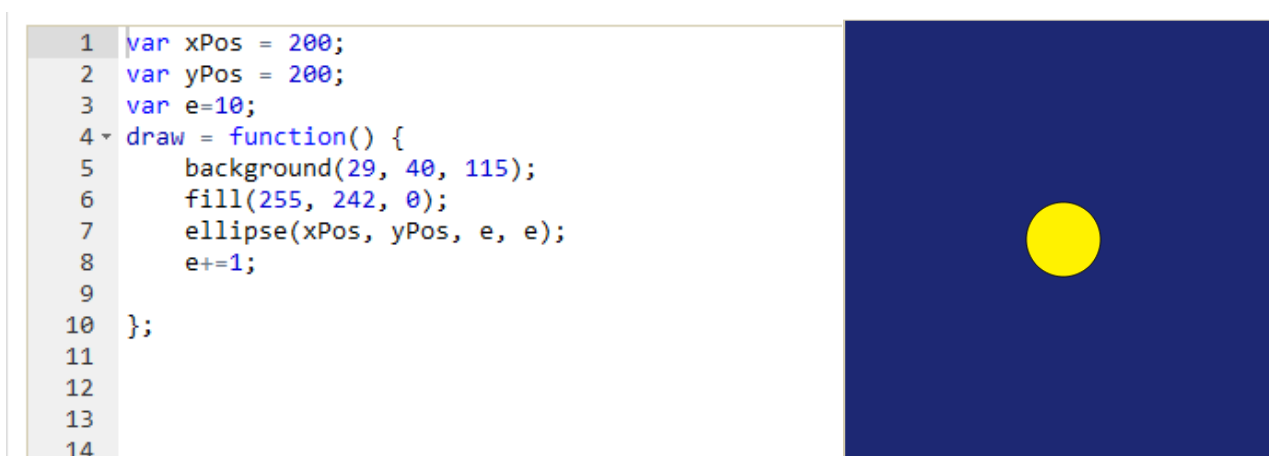


Figura 7.24. Programa escrito por el Caso 1 en el módulo 'Intro to JS: Drawing & Animation' del curso 'Computer Programming' [Recuperado de <https://www.khanacademy.org/computer-programming/spin-off-of-project-shooting-star/6674640557309952>]

⁴¹⁹ Ver en <https://www.khanacademy.org/computer-programming/spin-off-of-project-shooting-star/6674640557309952>

En la Figura 7.25, se ilustra uno de los productos-proyectos⁴²⁰ de programación publicados por el Caso 2 en la galería de código de Khan Academy. Tal y como puede observarse en la figura, el Caso 2 ha sido capaz de escribir un programa de casi 20 líneas de código en lenguaje textual ‘JavaScript & ProcessingJS’ que incluye: definición inicial de dos variables [ángulo en el que comienza el arco (‘wing’), y ángulo en el que finaliza el arco (‘wing2’)]; una función (‘function’) que incluye dichos parámetros-variables, además de los códigos RGB de color del fondo (‘background’), de color de las figura (‘fill’), y de posición y tamaño de texto (‘text’); y un bucle que repite indefinidamente esta función (‘draw’ = ‘function’) con una variación del ángulo en que comienza y finaliza el arco (‘wing--’, ‘wing2--’) en cada iteración. Tal y como puede intuirse, al ejecutar el programa se genera el efecto animado de que el plátano tiene una rueda que le gira encima, a modo de pelador (recomendamos vivamente visualizar el programa en acción, pinchando sobre el enlace al pie de la figura)

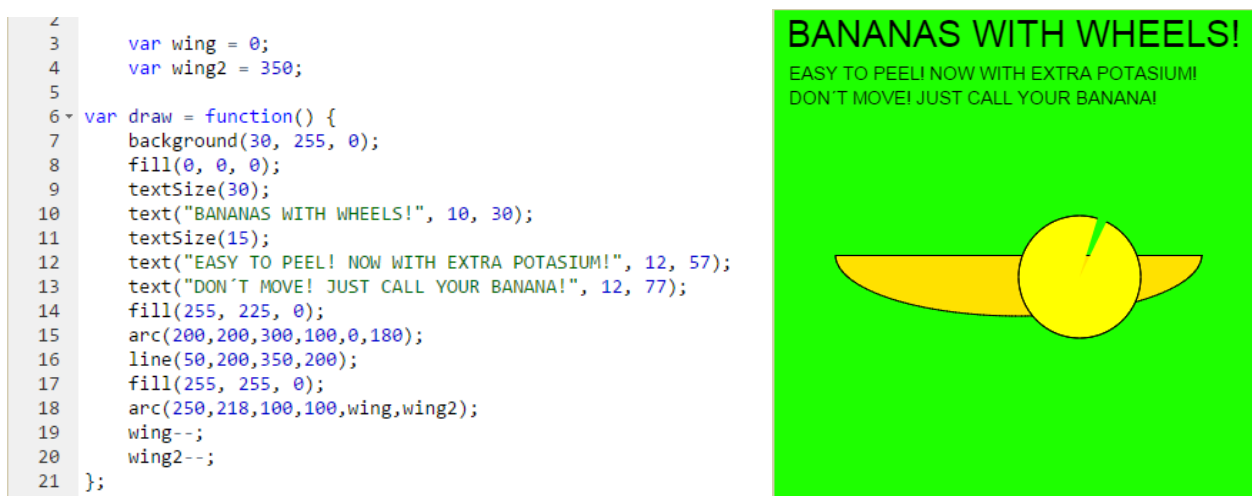


Figura 7.25. Programa escrito por el Caso 2 en el módulo ‘Intro to JS: Drawing & Animation’ del curso ‘Computer Programming’ [Recuperado de <https://www.khanacademy.org/computer-programming/spin-off-of-project-ad-design/5915228372729856>]

7.3.5. Discusión y conclusiones parciales

Antes de entrar a concluir acerca de cada uno de los objetivos específicos del estudio de casos, merece discusión el hecho de que todos los sujetos categorizados como de ‘alta capacidad computacional’ son varones (tanto los 5 del IES Andreu Sempere como los 2 del IES María Blasco que han protagonizado finalmente este estudio de casos). Es decir, una vez implantado el curso K-8 (Code.org, 2015a) en el aula ordinaria (lo cual supone un enriquecimiento curricular en codigoalfabetización y desarrollo del pensamiento computacional para todos los alumnos⁴²¹) y mediante metodología ‘self-paced’ (lo que permite que el talento computacional aflore); entonces emergen 7 sujetos, todos varones, que pueden y quieren acelerar hacia un entorno de programación de nivel superior. Decimos ‘pueden y quieren’ porque dicho desequilibrio de género no es sólo explicable por las posibles diferencias globales en la aptitud ‘pensamiento computacional’ a favor de

⁴²⁰ Ver en <https://www.khanacademy.org/computer-programming/spin-off-of-project-ad-design/5915228372729856>

⁴²¹ Todos los alumnos que cursan la asignatura optativa de Informática

los chicos (que en el estudio de la psicometría general del TPC arrojaron un tamaño ‘pequeño’ $d = 0,22$ en el conjunto de la muestra de 1º Ciclo de la ESO a favor del grupo masculino).

Así, por ejemplo, en el mismo grupo de clase del que hemos extraído nuestros dos sujetos-casos varones, encontramos dos chicas que en el *pretest* del TPC obtuvieron una $P_d = 26$ ($P_c = 99$), que es la misma puntuación que el Caso 2 (ver más arriba [Tabla 7.68](#)). Sin embargo, ninguna de ellas solicitó acelerar al curso de nivel superior ‘*Computer Programming*’ ([Khan Academy, 2015](#)). Estos hechos son consistentes con cierta literatura en el área de las altas capacidades, ya comentada en el Capítulo 1, que informa de cómo la aceleración es un modelo desestimado en ocasiones diferencialmente por las chicas de alta capacidad frente a sus homólogos masculinos ([Crombie, Bouffard-Bouchard, & Schneider, 1992](#)). Ello también es consistente con los resultados encontrados en la evaluación del K-8 que informan de un nivel significativamente menor de motivación en las chicas para seguir aprendiendo ‘*coding*’ tras dicho curso; y consistente con otros estudios que afirman que las chicas de alta capacidad son menos proclives al aprendizaje en entornos ‘*on-line*’ que sus iguales masculinos ([Yang et al., 2011](#)).

Ahora sí, con respecto al objetivo específico O_{6a} , en el marco de la directiva de *adecuación*, podemos concluir que:

- Cuando descendemos a nivel de sujeto, específicamente a 2 casos etiquetados como de ‘alta capacidad computacional’, se comprueba que el TPC tiene una buena potencia diagnóstica para detectar dichos casos (conclusión a la que ya llegamos tras los resultados del [sub-epígrafe 6.5.2.3.2](#), sobre la validez discriminante del TPC). No ocurre así con otro instrumento ya conocido como el RP30 que, aunque globalmente arroja una alta correlación positiva con el TPC (recordemos, $r = + 0,669$), al descender a la escala de sujeto no llega a detectar la ‘alta capacidad computacional’. Estos resultados van en la línea de las afirmaciones de Siegle (2004) en lo relativo a que el talento digital puede no ser detectado por pruebas diagnósticas habituales de inteligencia y/o resolución de problemas. Todo ello es una nueva evidencia de la necesidad de contar con un instrumento como el TPC, que sirva para la medida específica del pensamiento computacional. Y es que el pensamiento computacional no deja de ser una operacionalización de un constructo más amplio, como la inteligencia digital, de la que se ha venido apuntando su relativa independencia respecto de otras inteligencias ‘clásicas’, como la lógico-matemática o la lingüística ([Battro & Denham, 2007](#))

Con respecto al objetivo específico O_{6b} , en el marco de la directiva de *viabilidad*, podemos concluir que:

- Tanto el Caso 1 como el Caso 2 presentan actividad y progreso en la plataforma Khan Academy a lo largo del módulo ‘*Intro to JS: Drawing & Animation*’ del curso ‘*Computer Programming*’. El Caso 1 registra algo más de 16 horas de trabajo en la plataforma, mientras que el Caso 2 registra casi 10 horas.
- Sin embargo, aunque ambos sujetos-casos comparten etiqueta de ‘alta capacidad computacional’ y metodología ‘*flipped classroom*’ para la realización del módulo ‘*Intro to JS: Drawing & Animation*’; su estilo de aprendizaje y uso de la plataforma es completamente

distinto. El Caso 1 presenta un estilo más activo e independiente: emplea la mayor parte del tiempo en retos y ejercicios, y sigue con el curso más allá de la temporalización escolar. El Caso 2 presenta un estilo más reflexivo y dependiente: emplea la mayor parte del tiempo en videos y tutoriales, y una vez finalizada la temporalización escolar relacionada con el curso, no prosigue con el mismo por su cuenta. Ello es evidencia de:

- La heterogeneidad interna del grupo de ‘alta capacidad computacional’ en lo que a estilo de aprendizaje ‘*on-line*’ se refiere.
- La versatilidad de plataformas como Khan Academy para adaptarse a los estilos individuales, posibilitando que el aprendizaje rompa las barreras del espacio-tiempo escolar. Por fin, el sujeto de alta capacidad no se encuentra limitado a la estructuración de tiempos y contenidos fijados por el profesor y/o por el currículo oficial.

Con respecto al objetivo específico **O_{6c}**, en el marco de la directiva de *relevancia*, podemos concluir que:

- Ambos sujetos-casos, de 12 años de edad (1º ESO), son capaces de escribir (y compartir) un programa informático en lenguaje textual ‘*JavaScript & ProcessingJS*’ utilizando conceptos computacionales avanzados como las funciones con parámetros-variables insertas en bucles indefinidos.
- Con ello, ambos sujetos-casos superan de largo el techo que fijamos en el diseño del TPC para estas edades (lectoescritura con lenguajes visuales ‘por bloques’, llegando hasta el concepto computacional de ‘funciones simples’); y se alinean con los estándares de codigoalfabetización fijados aproximadamente para final de 3º ESO tanto por el currículum británico ([UK Department of Education, 2013](#)) como por el de la Comunidad de Madrid ([DECRETO 48/2015](#)), que ya incluyen los lenguajes textuales y las funciones con parámetros-variables. Dicho de otra manera, nuestros dos sujetos-casos han acelerado aproximadamente el equivalente a dos cursos.

CAPÍTULO 8

CONCLUSIONES, IMPLICACIONES, LIMITACIONES Y SUGERENCIAS

8.1. Conclusiones

Al final de cada uno de los tres estudios empíricos de esta tesis doctoral, ya se discutieron los correspondientes resultados y se procedió a enunciar unas conclusiones parciales, muy detalladas, referidas a los objetivos específicos respectivos de cada estudio. Así:

- En el [apartado 5.5](#) se enuncian las conclusiones parciales relativas a los objetivos específicos abordados en el primer estudio (O_{1*} y O_{2*})
- En el [apartado 6.6](#) se enuncian las conclusiones parciales relativas a los objetivos específicos abordados en el segundo estudio (O_{3*})
- Con respecto al tercer estudio, compuesto a su vez por tres sub-estudios:
 - En el [epígrafe 7.1.5](#) se enuncian las conclusiones parciales relativas a los objetivos específicos, formulados como hipótesis específicas, abordados en el primer sub-estudio (H_{4*})
 - En el [epígrafe 7.2.5](#) se enuncian las conclusiones parciales relativas a los objetivos específicos, formulados como hipótesis específicas, abordados en el segundo sub-estudio (H_{5*})
 - En el [epígrafe 7.3.5](#) se enuncian las conclusiones parciales relativas a los objetivos específicos abordados en el tercer sub-estudio (O_{6*})

De manera análoga, a continuación vamos a proceder a enunciar las conclusiones finales, de carácter necesariamente sintético, que estarán referidas a cada uno de los seis objetivos generales de esta tesis doctoral. Para facilitar su posterior comunicación y defensa, se procederá a enunciar las conclusiones finales numerándolas de manera correlativa.

8.1.1. Con respecto al primer objetivo general

Objetivo 1 (O_1): Describir el impacto del evento ‘La Hora del Código’ (a partir de ahora, siguiendo la abreviatura inglesa habitual, ‘HoC’) en España a lo largo de sus dos primeras ediciones: HoC-2013 y HoC-2014.

Primera: Se evidencia un fortísimo crecimiento del evento ‘La Hora del Código’ en España, con un número de centros educativos registrados en el mismo que se multiplica por cinco entre 2013 y 2014 (y por quince entre 2013 y 2015) en nuestro país.

Segunda: Aceptando ‘La Hora del Código’ como un evento indicador de los procesos de codigoalfabetización en España, se puede afirmar que la codigoalfabetización irrumpe en España en 2013 por la zona Este de la península de la mano de centros privados y concertados multietapa, con especial incidencia en Secundaria; posteriormente, en 2014 la codigoalfabetización se extiende hacia el Centro y Noroeste de la península, con la incorporación del sector público y la etapa de Primaria.

8.1.2. Con respecto al segundo objetivo general

Objetivo 2 (O₂): Explorar evidencias empíricas en dicho evento ‘HoC’ que den apoyo y fundamento a nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.

Tercera: Los profesores responsables de ‘La Hora del Código’ en España consideran que: el ‘coding’ es *adecuado* para el desarrollo de un amplio abanico de aptitudes y habilidades (principalmente razonamiento lógico y resolución de problemas, pero también aptitudes espaciales, atencionales, de motivación e iniciativa, creatividad o trabajo en equipo); que el ‘coding’ es *adecuado* para todo tipo de alumnos, independientemente de su edad, género o capacidad; y que el ‘coding’ es *adecuado* para atender a la diversidad del alumnado dado la flexibilidad y autonomía que ofrecen sus recursos de aprendizaje.

Cuarta: Los profesores responsables de ‘La Hora del Código’ en España están de acuerdo en señalar la *viabilidad* del aprendizaje del ‘coding’: con los lenguajes visuales ‘por flechas’ para Educación Infantil y comienzos de Educación Primaria; con los lenguajes visuales ‘por bloques’ para finales de Educación Primaria y comienzos de Educación Secundaria; y con los lenguajes ‘textuales’ para finales de Educación Secundaria, Bachillerato y Formación Profesional.

Quinta: Los profesores responsables de ‘La Hora del Código’ en España declaran que es *viable* introducir a los estudiantes en el ‘coding’ a través de una actividad de 1 hora, consiguiendo: despertar la curiosidad del alumnado para proseguir con un aprendizaje del ‘coding’ más profundo; mostrar a la comunidad educativa la accesibilidad de la programación informática para todo tipo de alumnos; y desmitificar la dificultad extrema del ‘coding’. Según estos mismos profesores, ello es posible dadas algunas características de los tutoriales ‘on-line’ que ofrece ‘La Hora del Código’: tutoriales gamificados, accesibles y adaptables, basados en su mayoría en lenguajes visuales de programación, que ofrecen *feed-back* inmediato al alumno y mantienen alto su nivel de motivación y percepción de autoeficacia subsiguiente; así como le permiten publicar y compartir los proyectos de código realizados.

Sexta: Los profesores responsables de ‘La Hora del Código’ en España informan unánimemente de un excelente caldo de cultivo en la comunidad educativa para celebrar el evento: muchos apoyos y casi ninguna resistencia. Hay una asunción colectiva y mayoritaria de padres, madres, estudiantes, equipos directivos y profesorado, de la importancia de la programación informática o ‘coding’ como habilidad digital clave en el siglo XXI. Dado este ambiente de receptividad al evento, y la posibilidad de escalar fácilmente los recursos de aprendizaje del ‘coding’, es *viable* organizar ‘La Hora del Código’ a nivel de todo un centro educativo.

Séptima: Los profesores responsables de ‘La Hora del Código’ en España están de acuerdo en señalar la *relevancia* del aprendizaje del ‘coding’ para el futuro de sus estudiantes, como ciudadanos y como profesionales digitales del siglo XXI.

Octava: Los profesores responsables de ‘La Hora del Código’ en España señalan cierta casuística y anecdotario entre sus estudiantes durante la celebración del evento; siendo las más recurrentes: estudiantes que muestran un talento especial en tareas de ‘coding’ y que no necesariamente destacan en otras áreas curriculares; sujetos o grupos habitualmente disruptivos que muestran atención sostenida y perseverancia en tareas de ‘coding’; chicas inicialmente con prejuicios negativos alrededor del ‘coding’ que se ven confrontadas ante la posibilidad real de iniciar su codigoalfabetización; y la buena acogida de los estudiantes de la metodología ‘pair programming’.

Novena: Algo más de la mitad de los estudiantes participantes en ‘La Hora del Código’ (53,0%) declara haber oído hablar de la programación informática o ‘coding’ con anterioridad al evento; y algo menos de la mitad (43,0%) declara tener experiencia previa en ‘coding’ con anterioridad al mismo. Los porcentajes se van incrementando consistentemente a lo largo de las distintas etapas educativas.

Décima: Los lenguajes de programación con los que un mayor porcentaje de estudiantes declara haber tenido experiencia previa a ‘La Hora del Código’ son: HTML (14,02%), Scratch (12,03%), JavaScript (9,63%), CSS (4,45%), Python (3,61%), Lightbot (2,77%) y Blockly (2,17%). Se observa una clara trayectoria evolutiva en el uso de los lenguajes: los lenguajes visuales encuentran su mayor uso en Primaria (Lightbot, ‘visual por flechas’) y en Secundaria (Scratch y Blockly, ‘visuales por bloques’); los ‘pseudolenguajes’ textuales de etiquetas y estilo (HTML y CSS) tienen su mayor uso en el Bachillerato; y los lenguajes textuales ‘puros’ (JavaScript y Python) encuentran su mayor uso en la Formación Profesional.

Undécima: Los estudiantes participantes en ‘La Hora del Código’ perciben que el ‘coding’ contribuye a desarrollar sus aptitudes y habilidades. Esta percepción se organiza alrededor de tres componentes principales, relativamente independientes entre sí: 1) un componente principal o ‘*el acto de programar en sí*’ que aglutina dos subcomponentes: uno cognitivo (‘pensamiento lógico’ y ‘resolución de problemas’) y otro motivacional-atencional (‘persistencia-perseverancia’ y ‘capacidad de atención’); 2) un componente de menor peso, relativo a la carga creativa del ‘*entorno-tarea de programación*’ (‘close-ended’ vs. ‘open-ended’); 3) y un último componente, relativo al carga colaborativa de la ‘*metodología de programación*’ utilizada en el aula (‘single programming’ vs. ‘pair programming’).

Duodécima: Globalmente, el evento ‘La Hora del Código’ logra su objetivo de desmitificar la dificultad del ‘coding’ (un 70,2% de los participantes declara que se les da bien o muy bien, tras la experiencia); y de motivar a los estudiantes a proseguir con un aprendizaje más profundo del ‘coding’ tras el evento (un 60,2% de los participantes dice estar bastante o muy motivado para ello). Además, 8 de cada 10 estudiantes declaran su determinación a seguir aprendiendo ‘coding’ a través de entornos autoguiados ‘on-line’. Así pues, la mayoría de los estudiantes participantes en ‘La Hora del Código’ entran en un círculo motivacional virtuoso en ‘coding’ tras el evento: buena percepción de autoeficacia, buena motivación, y buena determinación a la acción.

Decimotercera: Las variables que mejor explican la motivación por proseguir con el aprendizaje del ‘coding’ tras ‘La Hora del Código’ son: la autoeficacia percibida en ‘coding’ tras la actividad, la percepción de que el ‘coding’ contribuye a desarrollar la creatividad y la resolución de problemas; y, en sentido inverso, la etapa educativa del estudiante. En síntesis, para promover una alta motivación de los estudiantes en continuar aprendiendo ‘coding’ es crítico plantear tareas que sean capaces de acometer con éxito y que les planteen problemas significativos que puedan abordarse desde un plano creativo; ello parece estar sucediendo más adecuadamente en Primaria que en Secundaria.

Decimocuarta: Las variables que mejor explican la determinación a seguir aprendiendo ‘coding’ tras ‘La Hora del Código’ a través de entornos autoguiados ‘on-line’ son, además de la autoeficacia y la motivación: la experiencia previa en ‘coding’ del estudiante y su percepción de que éste contribuye al desarrollo del pensamiento lógico. En síntesis, el perfil más específico del estudiante que se declara proclive al aprendizaje del ‘coding on-line’ autoguiado, es el de un sujeto que ya ha tenido contacto con la programación informática y que es especialmente sensible a sus aspectos formales-lógicos.

Decimoquinta: Existe una correlación significativa entre tener experiencia previa en ‘coding’ y el rendimiento académico informado por los estudiantes participantes en ‘La Hora del Código’; especialmente en las asignaturas de Tecnología/Informática y Matemáticas. Además, dicha correlación es más fuerte en el colectivo masculino. Así, el tener contacto con la programación informática podría estar funcionando como catalizador del buen rendimiento académico, especialmente en las asignaturas más afines y con un efecto acelerador especialmente intenso entre los chicos (las chicas tienden a rendir alto en cualquier caso).

Decimosexta: Las plataformas de aprendizaje del ‘coding’ más conocidas por los estudiantes participantes en ‘La Hora del Código’ son: Code.org y Scratch, con una alta penetración que se sitúa en el orden del 30-40% del total de estudiantes; Light-Bot, App Inventor, Khan Academy, Codecademy, Tynker, Blockly y CodeHS, con una penetración moderada en el orden del 5-10%; y c) el resto de plataformas-tutoriales, con una baja penetración que no llega al 1% sobre el total de estudiantes.

Decimoséptima: Una amplia mayoría de los estudiantes participantes en ‘La Hora del Código’ afirma la relevancia del ‘coding’, tanto para su futuro personal (87,6%) como profesional (82,1%). Las razones que aportan para sustentar dicha información son consistentes con los mensajes lanzados desde el evento, y giran alrededor de aspectos sociológicos (‘el coding como respuesta a un mundo-sociedad tecnológico y digital), utilitarios (‘el coding como herramienta para conseguir beneficios personales y profesionales’), y de desarrollo (‘el coding como potenciador de una mente lógica y solucionadora de problemas). Es preocupante la escasa frecuencia de aparición de razones de tipo expresivo-comunicativas (‘el coding como vía de expresión a través de la creación de objetos digitales’), emotivas (‘el coding como fuente de diversión y satisfacción’), o críticas (‘el coding como elemento de empoderamiento y alfabetización crítica en la sociedad digital; programar para evitar ser programado’).

Decimoctava: Globalmente, parece que los tutoriales de Code.org, en el contexto de ‘La Hora del Código’, ofrecen ventajas respecto a los de Scratch. Este hecho es consistente con el diseño de los

tutoriales de Code.org, muy orientados a la solución progresiva de puzzles breves de programación, en un entorno gamificado y con *feed-back* inmediato que consigue, en sólo 1 hora, elevar la sensación de competencia y motivación a corto plazo del aprendiz; mientras que Scratch se orienta a la creación de proyectos abiertos a través del código, exigiendo procesos de aprendizaje más largos (y, al principio del proceso, bastante dirigidos por el docente).

Decimonovena: Para finalizar las conclusiones relativas a ‘La Hora del Código’, desde una perspectiva de género, se evidencia una brecha progresiva en lo referente al ‘*coding*’: es leve en Educación Primaria, emerge a partir de Educación Secundaria, y se agranda aún más en Bachillerato; hasta el punto de que, en el nivel máximo de codigoalfabetización o lectoescritura con lenguajes textuales ‘puros’, el factor de género es 2,5-3 veces superior en chicos. Esta brecha de género tiene, al menos, tres aspectos: 1) las chicas están menos expuestas que los chicos a la idea del ‘*coding*’, y se atreven menos a probarlo; 2) las chicas se sienten menos autoeficaces que los chicos tras la actividad introductoria de ‘*coding*’; 3) y las chicas se sienten menos determinadas que los chicos a proseguir el aprendizaje del ‘*coding*’; especialmente si éste se realiza en entornos autoguiados ‘*on-line*’. En síntesis, es necesario trabajar por la equidad de género en lo que a la codigoalfabetización se refiere; lo retomamos en el apartado de implicaciones.

8.1.3. Con respecto al tercer objetivo general

Objetivo 3 (O₃): Diseñar y validar un instrumento que mida el nivel de desarrollo del pensamiento computacional, o ‘Test de Pensamiento Computacional’ (TPC), en población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO).

Vigésima: Globalmente, se ha conseguido diseñar y validar un test de 28 ítems de longitud, el ‘Test de Pensamiento Computacional’ (TPC), que mide el nivel de desarrollo del pensamiento computacional, fundamentalmente en población escolar española de primer ciclo de la Educación Secundaria Obligatoria (1º ESO – 2º ESO), con extensiones hacia tercer ciclo de Primaria y segundo ciclo de Secundaria. Para ello se han seguido las indicaciones metodológicas de Buffum *et al.* (2015) para la construcción de instrumentos de medida en el área de las Ciencias de la Computación en edades ‘*middle school*’, indicaciones a su vez alineadas con los estándares para la medición psicoeducativa de la AERA, la APA, y la NCME (AERA, APA, & NCME, 2014).

Vigesimoprimera: El TPC presenta una adecuada validez de contenido. Los conceptos computacionales que constituyen el dominio del TPC: están alineados con los estándares de aprendizaje fijados por la CSTA para la educación en Ciencias de la Computación en edades ‘*middle school*’ (CSTA, 2011); son consistentes con los conceptos computacionales medidos por otros instrumentos de similares características recientemente diseñados en Alemania (Mühling, Ruf, & Hubwieser, 2015) o Estados Unidos (Weintrop & Wilensky, 2015a); y han sido validados a través del juicio de una veintena de expertos de nuestro país.

Vigesimosegunda: El TPC presenta una adecuada validez aparente. Se ha conseguido virtualizar-digitalizar el TPC de manera que éste sea accesible desde cualquier dispositivo electrónico fijo o móvil con conexión a Internet. Así, el TPC presenta una adecuada legibilidad y navegabilidad, siendo los sujetos de la población objetivo capaces de: acceder al TPC a partir del enlace URL que se

les proporciona; leer y comprender autónomamente las instrucciones y ejemplos; navegar a lo largo del TPC; y emitir las correspondientes respuestas, enviándolas a su conclusión.

Vigesimotercera: Se ha conseguido aplicar el TPC sobre una muestra representativa a nivel nacional y de tamaño suficiente. En concreto, se ha acumulado una muestra de 1.251 sujetos, de los cuales 735 (58,8%) corresponden al foco central del instrumento (1º y 2º de la ESO). Los sujetos proceden de un total de 24 centros educativos distintos, ubicados en 7 provincias diferentes.

Vigesimocuarta: El TPC presenta, en general, unos índices psicométricos adecuados. Así, las puntuaciones totales en el TPC: se distribuyen con un ajuste notable a la curva normal y de manera simétrica alrededor de la media, que se sitúa muy cerca del punto medio de la escala; presentan una buena variabilidad, lo que permite la construcción de baremos adecuados para las poblaciones de referencia; se incrementan de manera significativa según avanzamos en ciclo educativo y curso académico. Todo ello apoya la afirmación de que el TPC es una prueba de competencia cognitiva.

Vigesimoquinta: Globalmente, el TPC está compuesto por ítems adecuados. Así, los 28 ítems del TPC arrojan, de valor promedio: un índice de dificultad en el intervalo 0,40–0,60 (dificultad adecuada para la población objetivo); una correlación ítem-total $> 0,30$ (adecuada homogeneidad interna); y un índice de discriminación $> 0,30$. El análisis en detalle de cada uno de los 28 ítems del TPC en los indicadores anteriores, revela tres claros candidatos a ser eliminados en futuras versiones mejoradas del instrumento: ítems 12, 15 y 23.

Vigesimosexta: El TPC presenta una adecuada fiabilidad. Su fiabilidad como consistencia interna es buena ($\alpha \approx 0,80$), incrementándose cuando el TPC es aplicado sobre dispositivos móviles tipo tableta. (p.e. $\alpha \approx 0,84$ en aplicaciones sobre tableta en primer ciclo de la ESO). Su fiabilidad como como estabilidad temporal es igualmente aceptable ($\rho_{xx} \approx 0,70$).

Vigesimoséptima: El TPC presenta una adecuada validez criterial concurrente. Así, el TPC correlaciona concurrentemente con medidas estandarizadas de razonamiento lógico, aptitud espacial y resolución de problemas; esta terna de correlaciones es consistente con las recientes formulaciones del constructo ‘pensamiento computacional’ de Ambrosio *et al.* (2014), que lo relacionan con los factores *Gf* (*Fluid reasoning*), *Gv* (*Visual processing*), y *Gsm* (*Short-term memory*) del modelo de inteligencia Cattell-Horn-Carroll (CHC) (McGrew, 2009). Adicionalmente, la dimensión de personalidad que correlaciona más intensamente con el TPC, es la denominada ‘Apertura’ (dimensión vinculada con aspectos intelectuales); ello es nueva evidencia de la naturaleza fundamentalmente cognitiva de la medida del TPC.

Vigesimoctava: El modelo de regresión de las aptitudes mentales primarias sobre el TPC explica un 27,0% de su varianza; mientras que el modelo de regresión de las dimensiones de personalidad sobre el TPC (construido de manera independiente al anterior) explica un 24,1% de su varianza. Ello nos lleva a concluir: que el TPC es una medida fundamentalmente cognitiva, pero también sensible a las dimensiones de personalidad (consistentemente con algunas definiciones del pensamiento computacional que incorporan aspectos actitudinales); y que el TPC tiene un amplio porcentaje de varianza específica, no explicable por otras medidas estandarizadas de aptitud o personalidad, que nos lleva a concluir sobre la plausible especificidad del pensamiento computacional como constructo psicopedagógico con entidad propia.

Vigesimonovena: El TPC presenta una adecuada validez criterial predictiva. En concreto, el TPC se muestra válido: para predecir el rendimiento académico en la asignatura de Informática, siempre que el currículum seguido en ésta sea de codigoalfabetización y desarrollo del pensamiento computacional; y para predecir el nivel de desempeño objetivo de los sujetos en plataformas de aprendizaje del ‘coding’ como Code.org, en especial si los sujetos siguen una metodología autoguiada ‘self-paced’ (a su propio ritmo). En este sentido, concluimos que la metodología ‘self-paced’ tiende a equiparar en mayor medida *potencial computacional* a *rendimiento computacional*, que la metodología ‘teacher-paced’ (ritmo homogéneo fijado por el profesor).

Trigésima: El TPC presenta una adecuada validez discriminante sobre la ‘alta capacidad computacional’, definida ésta como propia de los sujetos que solicitan acelerar a entornos de codigoalfabetización de nivel superior (concretamente en primer ciclo de la ESO, acelerar desde entornos con lenguajes de programación visuales por bloques hacia entornos con lenguajes textuales). Puntuaciones ≥ 25 en situación *pretest* con el TPC en primer ciclo de la ESO son especialmente discriminantes de sujetos con ‘alta capacidad computacional’.

Trigesimoprimera: El TPC presenta una adecuada validez convergente con otras medidas alternativas del pensamiento computacional. En concreto, el TPC correlaciona con Tareas Bebras que implican pensamiento lógico-condicional y pensamiento algorítmico; no así con Tareas Bebras que implican pensamiento paralelo (ello consistente con la ausencia de ítems de ‘paralelización’ en el TPC). Igualmente, el TPC correlaciona con Dr. Scratch, tanto de manera predictiva como concurrente. En conjunto, los valores de validez convergente son algo inferiores de lo deseable ($r \approx 0,5 < 0,7$); pero explicables dada la distinta orientación de los instrumentos: TPC (medida aptitudinal-diagnóstica), Tareas Bebras (medida competencial, de transferencia), Dr. Scratch (medida formativa-iterativa de proyectos de ‘coding’). Se concluye la bondad y necesidad del uso complementario de las tres medidas.

Trigesimosegunda: El TPC presenta una validez factorial adecuada, aunque no concluyente. Se encuentra una estructura factorial mixta, que combina un factor común (representado por la subescala ‘Común-Principal’), con 3 factores específicos (representados por las subescalas ‘Secuencias y Bucles’, ‘Funciones’ y ‘Condicionales’) relativamente independientes entre sí (correlaciones $r \approx +0,3$ entre las subescalas específicas). Dicha combinación compone la versión reducida TPC-R (20 ítems), que posee varias ventajas psicométricas respecto del TPC (28 ítems): menor longitud y menor tiempo de aplicación; fiabilidad similar; cuatro puntuaciones derivadas de las cuatro sub-escalas; eliminación de los ítems 12, 15 y 23 (detectados como débiles en el análisis de ítems); y mejor comportamiento con respecto a los criterios de validez concurrente (p.e. con el TPC-R emerge la correlación con la aptitud perceptivo-atencional y se difumina la correlación no esperada con la dimensión ‘Extraversión’ de personalidad).

Trigesimotercera: Para finalizar las conclusiones relativas al TPC, y desde una perspectiva de género; el rendimiento en el TPC de los chicos es superior al de las chicas ($d = 0,31$). Ello es consistente con la carga lógica y, sobre todo, espacial del TPC. Dicho rendimiento diferencial en el TPC a favor del colectivo masculino se incrementa a medida que avanzamos en ciclo educativo, y sólo es estadísticamente significativo de primer ciclo de la ESO en adelante; por tanto, se concluye una brecha de género progresiva en lo que a nivel de desarrollo del pensamiento computacional se

refiere. Esta brecha constituye una potencial amenaza a la equidad de género en lo relativo a la medida del emergente constructo del pensamiento computacional. Lo retomamos en el apartado de implicaciones.

8.1.4. Con respecto al cuarto objetivo general

Objetivo 4 (O₄): Evaluar el programa-curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a), desde el marco fijado por nuestras hipótesis directivas de investigación, ya enunciadas en el apartado 2.6 de esta tesis.

Trigesimocuarta: En comparación con el currículum TIC tradicional, el curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) mejora significativamente el pensamiento computacional de los estudiantes, medido a través del TPC; esta mejora se produce tanto en primer como en segundo ciclo de la ESO, y tanto en chicos como en chicas. Ello constituye una evidencia adicional de la validez discriminante del TPC, para diferenciar entre sujetos que han o no han seguido un curso-programa específico de codigoalfabetización y desarrollo del pensamiento computacional.

Trigesimoquinta: En comparación con el currículum TIC tradicional, el curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) no mejora significativamente ninguna de las otras variables aptitudinales de los estudiantes. Aunque sí se encuentran indicios que señalan la plausibilidad de que, en futuras replicaciones con muestras de mayor tamaño, se puedan producir mejoras atribuibles al curso K-8 en razonamiento lógico, resolución de problemas y control de la impulsividad.

Trigesimosexta: Así, globalmente, se concluye que el K-8 presenta una *adecuación* restringida: produce mejoras específicas en pensamiento computacional, pero no en otras variables aptitudinales clásicas.

Trigesimoséptima: Globalmente, los profesores señalan la *viabilidad*, en términos de implantación, del curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) en aulas ordinarias de 1° y 2° ciclo de la ESO (asignatura optativa de Informática). Así, en más del 50% de aulas de implantación: se dedican 8 o más semanas lectivas al K-8, y se imparten el 80% o más de las etapas que componen el mismo. Las estadísticas de desempeño objetivo (*‘tracking’*) de los estudiantes en el curso confirman dicha *viabilidad*, ya que más del 50% de los estudiantes completan al menos el 80% de los niveles-pantallas *‘on-line’* que componen el curso K-8.

Trigesimoctava: Globalmente, los profesores señalan la *viabilidad*, en términos de satisfacción, del curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a): hay unanimidad entre los profesores sobre impartir de nuevo el K-8 el siguiente curso académico; valoran en alto grado el aprendizaje realizado por los estudiantes y manifiestan una alta satisfacción con el curso. Los estudiantes confirman esta *viabilidad*, en términos de satisfacción, dado que, al terminar el K-8, también manifiestan alto grado en: percepción de autoeficacia en *‘coding’*, motivación por continuar aprendiendo *‘coding’*, y satisfacción global.

Trigesimonovena: Globalmente, los estudiantes afirman la relevancia del aprendizaje del *‘coding’* tras el curso ‘*K-8 Intro to Computer Science*’ (Code.org, 2015a) para su formación como ciudadanos

del siglo XXI. Sin embargo, las razones que manifiestan los estudiantes para sustentar dicha afirmación son de corte fundamentalmente utilitarista, sociológico y de desarrollo; y en mucha menor medida esgrimen razones expresivo-comunicativas, emocionales o críticas. Se concluye que los contextos ‘close-ended’ de aprendizaje del ‘coding’, como el K-8, son limitados en cuanto a la perspectiva final que generan en los sujetos sobre la *relevancia* de la codigoalfabetización en la sociedad actual; siendo necesario, por tanto, su complemento con otros contextos ‘open-ended’ o ‘hardware-driven’.

Cuadragésima: Para finalizar con las conclusiones relativas al curso ‘K-8 Intro to Computer Science’ (Code.org, 2015a), y desde una perspectiva de género; se concluye la ausencia de diferencias entre chicos y chicas en desempeño objetivo (‘tracking’) durante el curso. Sin embargo, aparecen niveles de motivación y satisfacción global significativamente menores entre las chicas tras su finalización; lo que nos lleva a concluir sobre la necesidad de diversificar los entornos de aprendizaje del ‘coding’ para cubrir los gustos y necesidades de ambos géneros. Lo retomamos en el apartado de implicaciones.

8.1.5. Con respecto al quinto objetivo general

Objetivo 5 (O₅): Evaluar el programa-curso ‘K-5 Computer Science Fundamentals (Course 2)’ (Code.org, 2015b), desde el marco fijado por nuestras hipótesis directivas de investigación, ya enunciadas en el [apartado 2.6](#) de esta tesis.

Cuadragésimoprimera: El pensamiento computacional de los estudiantes de tercer ciclo de Educación Primaria, medido a través del TPC, mejora significativamente entre el inicio y el final del curso ‘K-5 Computer Science Fundamentals (Course 2)’ (Code.org, 2015b). Aunque dicha mejora no es atribuible causalmente al curso K-5 dadas las debilidades del diseño *pre-experimental* utilizado, la magnitud del tamaño del efecto *pre-post* ($d \approx 0,80$; grande y similar al encontrado con el curso K-8) nos lleva a concluir que dicha relación causal es plausible; algo que debe ser replicado y que nos permitirá afirmar con seguridad la *adecuación* del K-5.

Cuadragésimosegunda: Al comparar la magnitud de los tamaños del efecto *pre-post* en pensamiento computacional, medido a través del TPC, entre los distintos cursos-programas, se puede concluir que: las mejoras en pensamiento computacional son mayores tras aprendizajes del ‘coding’ en contextos estructurados ‘close-ended’ (cursos K-8 o K-5), que tras lo propio en contextos abiertos ‘open-ended’ (Scratch); y mayores en estos últimos que tras el seguimiento de un currículum TIC tradicional (que no incluye ‘coding’). Esta relativa dependencia de la sensibilidad del TPC para detectar mejoras en pensamiento computacional en función del tipo de contexto de aprendizaje del ‘coding’ seguido (‘close-ended’ vs. ‘open-ended’) nos lleva a concluir la necesidad de su uso complementario con otros instrumentos de medida, como las Tareas Bebras o Dr. Scratch, para futuras evaluaciones de programas de codigoalfabetización y desarrollo del pensamiento computacional.

Cuadragésimotercera: Globalmente, se concluye la *viabilidad*, en términos de implantación, del curso ‘K-5 Computer Science Fundamentals (Course 2)’ (Code.org, 2015b) en tercer ciclo de

Educación Primaria, ya que más del 50% de los estudiantes completan al menos el 80% de los niveles-pantallas ‘*on-line*’ que componen el curso.

8.1.6. Con respecto al sexto objetivo general

Objetivo 6 (O₆): Estudiar casos de sujetos con alta capacidad computacional en el contexto del programa-curso ‘*Computer Programming*’ (Khan Academy, 2015); desde el marco fijado por nuestras hipótesis directivas de investigación, ya enunciadas en el apartado 2.6 de esta tesis.

Cuadragésimocuarta: Los casos de alta capacidad computacional son detectados *adecuadamente* a través del TPC; no así por otros instrumentos de medida sobre constructos próximos, como el test de resolución de problemas RP30. Ello nos lleva a concluir sobre la relativa independencia del talento computacional respecto de otros talentos cercanos; y, por tanto, sobre la necesidad de contar con instrumentos de medida específicos para el pensamiento computacional.

Cuadragésimoquinta: En las aulas de primer ciclo de la ESO que abordan la codigoalfabetización desde una metodología ‘*self-paced*’, emergen sujetos de alta capacidad computacional que pueden y quieren acelerar hacia contextos ‘*on-line*’ de codigoalfabetización superior; desde entornos con lenguajes visuales por bloques (Code.org) hacia entornos con lenguajes textuales (Khan Academy), y mediante un esquema ‘*flipped classroom*’. Ahora bien, se evidencia heterogeneidad interna entre los sujetos de alta capacidad computacional en lo que a estilo de uso-aprendizaje de la plataforma Khan Academy se refiere (estilo activo \approx pocos videos-tutoriales, muchos retos-ejercicios *vs.* estilo reflexivo \approx muchos videos-tutoriales, pocos retos-ejercicios)

Cuadragésimosexta: Sujetos de alta capacidad computacional de 1º ESO, que aceleraron hacia contextos ‘*on-line*’ de codigoalfabetización superior (Khan Academy), son capaces de escribir programas informáticos en lenguajes textuales (JavaScript & Processing JS) que involucran conceptos computacionales complejos como funciones con parámetros-variables insertas en bucles indefinidos. Ello supone, de hecho, una aceleración de dos cursos académicos si tomamos de referencia los recientes currículums en computación de Reino Unido y de la Comunidad de Madrid.

Cuadragésimoséptima: Para finalizar las conclusiones respecto del sexto objetivo general, y desde una perspectiva de género; se evidencian resistencias entre el colectivo femenino de mayor capacidad computacional de primer ciclo de la ESO para acelerar hacia contextos de codigoalfabetización superior. Lo retomamos en el apartado de implicaciones.

8.2. Respuesta al problema general de investigación

Ahora ya estamos en condiciones de dar respuesta a nuestro problema general de investigación, formulado allá por el Capítulo 2 de esta tesis doctoral (Figura 2.12): ¿es posible (*adecuado* y *viabile*) y deseable (*relevante*) introducir la codigoalfabetización en el sistema educativo español (Educación Primaria y Secundaria)? Dadas las anteriores conclusiones, **la respuesta es ‘sí’**. Ello supondría un enriquecimiento curricular del sistema en una triple dimensión: *contenido*, *proceso* y *producto* de aprendizaje. Ahora bien, con las siguientes matizaciones:

- ✓ Introducir la codigoalfabetización en el sistema educativo español (Educación Primaria y Secundaria) es *adecuado*. Pero se trata de una adecuación restringida (no indiscriminada): la codigoalfabetización mejora aptitudes específicas como el pensamiento computacional, no así otras aptitudes próximas de mayor tradición en la medición psicopedagógica.
- ✓ Introducir la codigoalfabetización en el sistema educativo español (Educación Primaria y Secundaria) es *viable*, tanto en términos de implantación como de satisfacción. Sin embargo, esta viabilidad corre el riesgo de quedar descompensada en términos de género, favoreciendo al colectivo masculino.
- ✓ Introducir la codigoalfabetización en el sistema educativo español (Educación Primaria y Secundaria) es *relevante*. Sin embargo, esta relevancia corre el riesgo de quedar restringida a los aspectos más utilitarios y sociológicamente deterministas.

Las matizaciones anteriores, con los riesgos que se advierten, nos llevan a formular el siguiente apartado de implicaciones; que cristaliza en una propuesta-modelo integral de codigoalfabetización y desarrollo del pensamiento computacional desde una perspectiva holística, equitativa y diversa.

8.3. Implicaciones

8.3.1. En el marco de la directiva de *adecuación*

Dado que la inteligencia digital, en general, y el pensamiento computacional, en particular, parecen tener especificidad propia y ser relativamente independientes de otras inteligencias y pensamientos afines; es necesario seguir avanzando en una mejor definición de ambos constructos, así como en sus instrumentos propios de medición.

Al volver sobre nuestra definición operativa de trabajo del pensamiento computacional, o '*la capacidad de formular y solucionar problemas apoyándose en los conceptos fundamentales de la computación...*', parece evidente que, en el desarrollo del TPC, nos hemos centrado en un tipo particular de problemas: problemas '*close-ended*' con carga lógica y espacial. Este sesgo, que parece favorecer a los chicos frente a las chicas, corre el riesgo de alimentar y mantener los estereotipos de género en cuanto a la aptitud para la programación informática se refiere. En otras palabras, se nos plantea el siguiente interrogante, al cual es fundamental dar respuesta ahora que encaramos los primeros pasos en la definición-medición del emergente constructo del pensamiento computacional: ¿medimos lo que existe, o existe lo que medimos? ¿El pensamiento computacional es una aptitud fundamentalmente masculina, o lo es debido a la forma (acaso sesgada) en que la estamos midiendo? Ello nos impulsa a trabajar en sucesivas versiones del instrumento TPC que impliquen la proyección del pensamiento computacional sobre un amplio abanico, más equitativo, de problemas: lógicos y viso-espaciales, pero también narrativos o musicales.

Igualmente, dadas las limitaciones inherentes al TPC (medida aptitudinal-diagnóstica, excesivamente centrada en conceptos computacionales proyectados sobre problemas lógicos y viso-espaciales, y con cierto sesgo de género), recomendamos que la medida del pensamiento computacional y la evaluación de los nuevos currículos de codigoalfabetización en Primaria y Secundaria, se apoyen en un conjunto de instrumentos complementarios entre sí, a saber: el propio TPC, Tareas Bebras

(enfoque competencial y de transferencia del pensamiento computacional a problemas cotidianos y significativos), y Dr. Scratch (enfoque iterativo-formativo sobre proyectos abiertos en Scratch).

8.3.2. En el marco de la directiva de *viabilidad*

La introducción de la codigoalfabetización en el sistema educativo (Educación Primaria y Secundaria) es viable, tanto en términos de implantación como de satisfacción. Pero si dicha implantación no se realiza de manera diversa (sobre temas diversos, en contextos diversos, y con ritmos diversos), se corre el riesgo de que la codigoalfabetización quede sesgada a favor de unos cuantos, predominantemente del colectivo masculino.

Así pues, la lectoescritura con código informático, y el pensamiento computacional subyacente, debe: proyectarse sobre todo tipo de problemas⁴²² (simulaciones científicas computacionales, narrativas computacionales, música computacional, diseños geométricos computacionales, etcétera); mediante todo tipo de contextos de aprendizaje (problemas cerrados-determinados ‘*close-ended*’, problemas o proyectos abiertos-indeterminados ‘*open-ended*’, y conexiones del código con el mundo físico ‘*hardware-driven*’ que vayan más allá de la pantalla e incluyan el componente cinestésico); y atendiendo todos los ritmos de aprendizaje, para que *todos* y *cada uno* de los estudiantes avancen hacia las mayores cotas de codigoalfabetización posibles que su aptitud y motivación les permitan. Por tanto, apostamos por ir avanzando hacia una codigoalfabetización que siga los principios generales de equidad, excelencia y atención a la diversidad.

8.3.3. En el marco de la directiva de *relevancia*

Finalmente, debemos asegurar que los sujetos que se codigoalfabeticen en Primaria y Secundaria interioricen durante el proceso que la lectoescritura de código no sólo les será útil para encontrar un mejor trabajo o para controlar los artefactos digitales de los cuales (casi) inevitablemente estarán rodeados; sino también para expresarse, para comunicarse, para divertirse, para ser ciudadanos críticos en el ecosistema digital. Diríamos:

“Pensar computacionalmente no para pensar como ordenadores, sino precisamente para poder seguir pensando (y expresando) como humanos en un mundo repleto de máquinas”

Todas estas implicaciones se sintetizan en la siguiente propuesta-modelo integral de codigoalfabetización y desarrollo del pensamiento computacional en Primaria y Secundaria, desde una perspectiva holística, equitativa y diversa (Figura 8.1).

⁴²² De lo contrario estaríamos limitando la codigoalfabetización a temas y contextos muy parciales, alimentando de nuevo los estereotipos de género asociados al área de las Ciencias de la Computación. Haciendo una analogía con la alfabetización analógica tradicional, es como si pretendiéramos abordar ésta solicitando a los sujetos que escribieran textos exclusivamente sobre temas de interés predominantemente masculino (p.e. imaginemos que la alfabetización analógica tradicional se basara exclusivamente en redacciones y dictados sobre rugby y béisbol) y luego nos sorprendiéramos por la mayor aptitud y motivación de los chicos hacia la escritura. Así lo denuncian ya en nuestro país algunas asociaciones en defensa de una codigoalfabetización equitativa como ‘Mujeres Tech’ [<https://mujerestech.com/>]

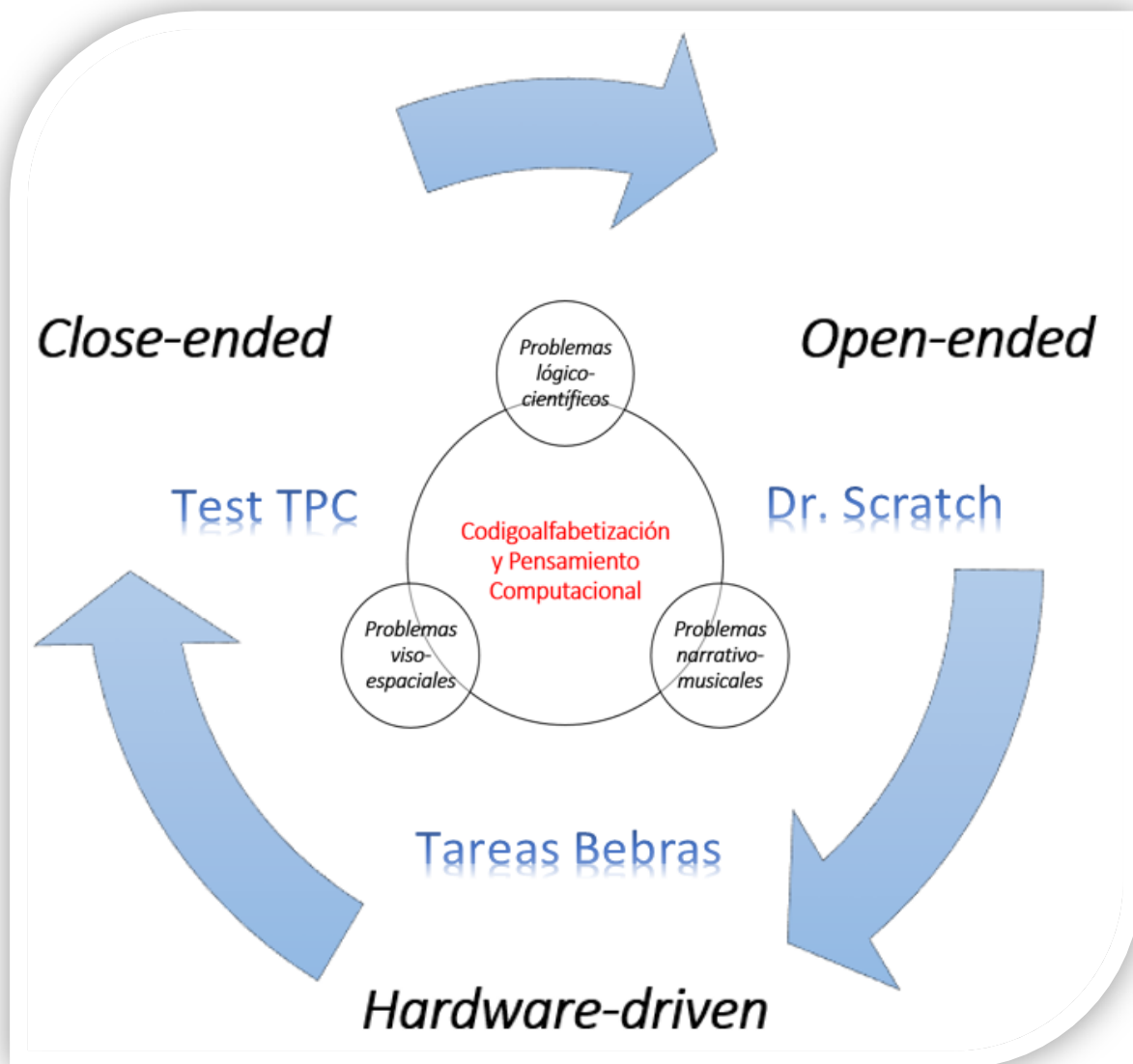


Figura 8.1. Modelo integral de codigoalfabetización y desarrollo del pensamiento computacional en Educación Primaria y Secundaria

8.4. Limitaciones y sugerencias

En conjunto, consideramos que los tres estudios empíricos que aporta esta tesis doctoral gozan de una alta validez externa: se acumulan muestras de gran tamaño y representativas a nivel nacional; y tanto la aplicación de instrumentos como la implantación de los programas se realizan en contextos naturales de aula ordinaria. En ese sentido, se estima que las conclusiones son fuertemente susceptibles de extrapolación.

Por el contrario, y en aras de preservar dicha alta validez externa, se ha sacrificado cierto control de variables extrañas tanto en la aplicación de instrumentos (p.e. el TPC fue aplicado en aula ordinaria con la supervisión del profesor correspondiente, pero sin la presencia de un administrador externo) como en la implantación de los programas (p.e. el K-8 fue implantado en múltiples IES, sin control estricto de algunas variables como ‘años de experiencia del profesor’ u ‘hora del día’ en que se

acometía el mismo). Todo ello constituye una limitación a la validez interna de nuestros estudios empíricos; y aconseja la replicación de partes concretas de los mismos a través de diseños más controlados.

Igualmente, la orientación general de nuestros estudios empíricos ha sido de *'datos masivos'*: acumular un gran número de aplicaciones del TPC para asegurar el estudio global de la fiabilidad y la validez del instrumento; y extender la implantación del curso K-8 a múltiples centros educativos con objeto de acumular grupos experimentales y de control de tamaño suficiente para la comparación entre ambas condiciones. Por tanto, nos hemos focalizado, respectivamente, en el nivel de *'constructo'* y en el nivel de *'grupo'*; y, sólo ligeramente, nos hemos acercado al nivel de *'ítem'* y al nivel de *'sujeto'*. Prestigiosos investigadores, tanto del ámbito de la medición de variables intelectuales-cognitivas (Haier, 2014) como del ámbito de la evaluación de programas educativos (Slavin, 2008), han señalado recientemente la necesidad de complementar ambos niveles de análisis para no incurrir en insuficiencias y sesgos. Queda pendiente para futuras investigaciones.

En lo relativo a nuestro primer estudio, descriptivo-exploratorio sobre *'La Hora del Código'* (HoC) en España; éste parte de la premisa de que el evento HoC es el más representativo para estudiar el fenómeno global de la *codigoalfabetización* en nuestro país. Debemos reconocer que hay otros eventos, como la *'CodeWeek'*⁴²³ promovida por la Comisión Europea, que podrían haber gozado de dicha consideración. Igualmente, dado que nuestra investigación sobre el evento *'HoC'* está limitada a sólo dos ediciones consecutivas (2013 y 2014), sugerimos extenderla sobre un periodo acumulado total de, al menos, cinco años; que permita hacer una aproximación verdaderamente longitudinal al fenómeno.

En lo relativo a nuestro segundo estudio, ya hemos venido señalando algunas limitaciones inherentes al diseño y validación del *'Test de Pensamiento Computacional'* (TPC): esencialmente, que es un instrumento casi exclusivamente orientado a medir *'conceptos computacionales'* (Brennan & Resnick, 2012), a partir de la resolución de un tipo concreto de problemas (lógicos y visoespaciales), y focalizado en edades *'middle school'*. En ese sentido se sugiere trabajar en futuras versiones del TPC que incluyan también *'prácticas computacionales'* y *'perspectivas computacionales'* (Brennan & Resnick, 2012), todo ello proyectado sobre un abanico más amplio de problemas (narrativos, musicales, etc...), y con extensiones hacia edades *'elementary school'* y *'high school'*. En cualquier caso, y mientras las anteriores sugerencias se van acometiendo, consideramos que el actual estado del TPC es ya suficientemente fiable y válido como para permitir su traducción y adaptación a otras lenguas⁴²⁴ y culturas; lo cual haga posible realizar, próximamente, estudios comparados de medición del pensamiento computacional entre escolares de distintos países.

En lo relativo a nuestro tercer estudio, la evaluación de programas acometida se ha focalizado sobre contextos de aprendizaje esencialmente *'close-ended'*, como son los cursos K-8 y K-5 de Code.org. Queda pendiente mucha investigación evaluativa sobre programas basados en contextos *'open-ended'* y *'hardware-driven'*; y especialmente sobre la combinación efectiva de todos ellos. Así pues, sugerimos emprender líneas de investigación que se interroguen sobre cuál o cuáles son las

⁴²³ <http://codeweek.eu/>

⁴²⁴ Durante la estancia internacional de investigación del doctorando en la Universidad de Sussex (verano de 2015) se han iniciado los trabajos de traducción y adaptación del TPC a lengua inglesa.

secuencias y combinaciones curriculares de mayor calidad para la codigoalfabetización y desarrollo del pensamiento computacional de nuestros escolares (p.e. una combinación posible en primer ciclo de la ESO es: primer trimestre escolar *'close-ended'* con Code.org, segundo trimestre escolar *'open-ended'* con Scratch, y tercer trimestre escolar *'hardware-driven'* con Arduino; ¿sería ésta la mejor secuenciación posible?)

En síntesis, creemos que esta tesis doctoral delimita un campo de juego, amplio y global, para la futura investigación en codigoalfabetización y desarrollo del pensamiento computacional. En conjunto, consideramos que, con esta tesis, se abren más preguntas de las que se cierran; esperamos que ello sea un buen indicador sobre su valor.

REFERENCIAS

- Adams, A., & Mowers, H. (10 de octubre de 2013). *7 Apps for Teaching Children Coding Skills* [Mensaje en un blog] Recuperado de <http://www.edutopia.org/blog/7-apps-teaching-children-coding-anna-adam>
- Adams, C. M., & Cross, T. L. (2000). Distance learning opportunities for academically gifted students. *Journal of Secondary Gifted Education*, 11(2), 88-96.
- Adams, N. B. (2004). *Digital intelligence fostered by technology*. *Journal of Technology Studies*, 30(2), 93-97.
- AERA, APA, & NCME (2014). *Standards for Educational and Psychological Testing*. Washington, DC: AERA.
- Agencia EFE (27 de marzo de 2013). Nick D'Aloisio, el adolescente de 17 años que ha creado una app de 23 millones. *La Vanguardia.com*. Recuperado de <http://www.lavanguardia.com/tecnologia/20130327/54369594772/nick-d-aloisio-adolescente-app-47-millones.html>
- Agencia EFE (08 de octubre de 2015). Más de 15 millones de usuarios en España utilizan «smartphones». *ABC.es*. Recuperado de <http://www.abc.es/tecnologia/moviles-telefonía/20150929/abci-usuarios-smartphones-espana-whatsapp-facebook-201509291429.html>
- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835.
- Akpinar, Y., & Aslan, Ü. (2015). Supporting Children's Learning of Probability Through Video Game Programming. *Journal of Educational Computing Research*, 53(2), 228-259.
- Allan, P. (20 de noviembre de 2014). These Are the Highest Paying Programming Skills to Have on Your Resume. *LifeHacker.com*. Recuperado de <http://lifelifehacker.com/these-are-the-highest-paying-programming-skills-to-have-1661389920>
- Alliance for Excellent Education (2012). *Sal Khan Video Lecture on Digital Learning* [Archivo de vídeo] Recuperado de <https://youtu.be/00tSs2xEpZY>
- Amabile, T. M., & Pillemer, J. (2012). *Perspectives on the Social Psychology of Creativity*. *The Journal of Creative Behavior*, 46(1), 3-15. doi: <http://dx.doi.org/10.1002/jocb.001>
- Ambrosio, A. P., Xavier, C., & Georges, F. (2014). Digital ink for cognitive assessment of computational thinking. *Frontiers in Education Conference (FIE), 2014 IEEE*, 1-7. doi: <http://dx.doi.org/10.1109/FIE.2014.7044237>
- Barbaranelli, C., Caprara, G. V., Rabasca, A., & Pastorelli, C. (2003). A questionnaire for measuring the big five in late childhood. *Personality and Individual Differences*, 34(4), 645-664.

- Barbaranelli, C., Caprara, G. V., & Rabasca, A. (2013). *BFQ-NA Cuestionario “Big Five” de Personalidad para Niños y Adolescentes* [Manual técnico]. Madrid: TEA Ediciones.
- Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, 2(1), 48-54
- Basawapatna, A., Koh, K. H., Repenning, A., Webb, D. C., & Marshall, K. S. (2011). Recognizing computational thinking patterns. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 245-250. doi: <http://dx.doi.org/10.1145/1953163.1953241>
- Basogain, X., Olabe, M. A., & Olabe, J.C. (2015). Pensamiento Computacional a través de la programación: paradigma de aprendizaje. *RED, Revista de Educación a Distancia*, 46. Recuperado de <http://www.um.es/ead/red/46/Basogain.pdf>
- Battista, M. T., & Clements, D. H. (1988). A case for a Logo-based elementary school geometry curriculum. *Arithmetic Teacher*, 36(3), 11-17.
- Battro, A. M. (2009a). Digital intelligence: The evolution of a new human capacity. *Scientific Insights into the Evolution of the Universe and of Life - Pontifical Academy of Sciences*, 20, 539-549.
- Battro, A. M. (2009b). Multiple intelligences and constructionism in the digital era. En J. Chen, S. Moran & H. Gardner (Eds.), *Multiple Intelligences around the World* (pp. 261-268). San Francisco, CA: Jossey-Bass.
- Battro, A. M., & Denham, P. J. (2007). *Hacia una inteligencia digital*. Buenos Aires: Academia Nacional de Educación.
- BBC News (09 de diciembre de 2014). President Obama writes Frozen code with school group. *BBC.com*. Recuperado de <http://www.bbc.com/news/technology-30393234>
- Bell, T., Witten, I.H., & Fellows, M. (2015). *CS Unplugged. An enrichment and extension programme for primary-aged students* [Documento en línea] Recuperado de http://csunplugged.org/wp-content/uploads/2015/03/CSUnplugged_OS_2015_v3.1.pdf
- Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., & Torelli, M. (2015). How challenging are Bebras tasks?: An IRT analysis based on the performance of italian students. *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, 27-32. doi: <http://dx.doi.org/10.1145/2729094.2742603>
- Belshaw, D. (22 de enero de 2014a). *Mozilla Open Badges and Web Literacy Map* [Presentación en línea] Recuperado de <http://www.slideshare.net/dajbelshaw/open-badges-web-literacy-map-101>
- Belshaw, D. (06 de enero de 2014b). *Going Beyond ‘Learning to Code’: Why 2014 is the Year of Web Literacy* [Mensaje en un blog] Recuperado de <http://dmlcentral.net/blog/doug-belshaw/going-beyond-learning-code-why-2014-year-web-literacy>

- Benbow, C. (1991). Mathematically talented children: can acceleration meet their educational needs? En N. Colangelo y G. A. Davis, *Handbook of Gifted Education* (pp. 154-165). Massachusetts: Allyn and Bacon.
- Bender, W., & Urrea, C. (2015). Visualizing Learning in Open-Ended Problem Solving in the Arts. *RED, Revista de Educación a Distancia*, 46. Recuperado de http://www.um.es/ead/red/46/bender_urrea.pdf
- Bender, W., Urrea, C., & Zapata-Ros, M. (2015). Pensamiento Computacional [Número monográfico]. *RED, Revista de Educación a Distancia*, 46. Recuperado de <http://www.um.es/ead/red/46/>
- Bennett, B., Kern, J., Gudenrath, A., & McIntosh, Ph. (2012). *The Flipped Class. What Does a Good One Look Like?* [Documento en línea]. Recuperado de <http://www.thedailyriff.com/articles/the-flipped-class-what-does-a-good-one-look-like-692.php>
- Bergmann, J., & Sams, A. (2012). *Flip your Classroom: Reach every student in every Class every day*. Washington, DC: ISTE.
- Besnoy, K. D., Dantzler, J. A., & Siders, J. A. (2012). Creating a digital ecosystem for the gifted education classroom. *Journal of Advanced Academics*, 23(4), 305-325. doi: <http://dx.doi.org/10.1177/1932202X12461005>
- Bitzer, J., Schrettl, W., & Schroder, P. J. H. (2007). Intrinsic motivation in open source software development. *Journal of Comparative Economics*, 35(1), 160-169. doi: <http://dx.doi.org/10.1016/j.jce.2006.10.001>
- Blair, R. (2011). Online learning for gifted students from the parents' perspectives. *Gifted Child Today*, 34(3), 28-30.
- Bloom, B. (1972). *Taxonomía de los objetivos de la educación. La clasificación de las metas educacionales*. Buenos Aires: El Ateneo.
- Boe, B., Hill, C., Len, M., Dreschler, G., Conrad, P., & Franklin, D. (2013). Hairball: Lint-inspired static analysis of Scratch projects. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 215-220. doi: <http://dx.doi.org/10.1145/2445196.2445265>
- Bohemia, E., & Ghassan, A. (2012). Globally networked collaborative learning in industrial design. *American Journal of Distance Education*, 26(2), 110-125.
- Bort, J. (18 de enero de 2016). Two Valley teen programmers have started a movement where kids teach other kids how to code. *BusinessInsider.com*. Recuperado de <http://www.businessinsider.com/teens-launch-kid-programming-movement-2016-01>
- Brennan, K., Balch, C., & Chung, M. (2014). *Creative Computing* [Documento en línea] Recuperado de <http://scratched.gse.harvard.edu/guide/>

- Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. *Proceedings of the 2012 Annual Meeting of the American Educational Research Association* (Vancouver: Canada) Recuperado de <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- Buffum, P. S., Lobene, E. V., Frankosky, M. H., Boyer, K. E., Wiebe, E. N., & Lester, J. C. (2015). A practical guide to developing and validating computer science knowledge assessments with application to middle school. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 622-627. doi: <http://dx.doi.org/10.1145/2676723.2677295>
- Bullard, B. (2005). Catching the tail of the comet: Technology in the classroom. *Understanding our Gifted*, 17(4), 9-12.
- Burke, Q. (2012). The markings of a new pencil: Introducing programming-as-writing in the middle school classroom. *The Journal of Media Literacy Education*, 4(2), 121–135.
- Burke, Q., & Kafai, Y. (2010). Programming & storytelling: opportunities for learning about coding & composition. *Proceedings of the 9th International Conference on Interaction Design and Children* (pp. 348–351). Barcelona, Spain: ACM. doi: <http://dx.doi.org/10.1145/1810543.1810611>
- Burker, J. (2015). Going from Bits to Atoms: Programming in Turtle Blocks JS and Personal Fabrication in Youth Maker Projects. *RED, Revista de Educación a Distancia*, 46. Recuperado de <http://www.um.es/ead/red/46/burker.pdf>
- Burnett, M. M., Beckwith, L., Wiedenbeck, S., Fleming, S. D., Cao, J., Park, T. H., ... Rector, K. (2011). Gender pluralism in problem-solving software. *Interacting with Computers*, 23(5), 450-460.
- Burns, A. M. (2006). Integrating technology into your elementary music classroom. *General Music Today*, 20(1), 6-10.
- Cadwalladr, C. (02 de marzo de 2014). The hour of code: why we should get with the program... and try coding. *TheGuardian.com*. Recuperado de <http://www.theguardian.com/technology/2014/mar/02/hour-of-code-get-with-program-try-coding>
- Calao, L. A., Moreno-León, J., Correa, H. E., & Robles, G. (2015). Developing mathematical thinking with Scratch: An experiment with 6th grade students. In *Design for Teaching and Learning in a Networked World, 10th European Conference on Technology Enhanced Learning, EC-TEL 2015* (pp. 17-27). Toledo, Spain: Springer. doi: http://dx.doi.org/10.1007/978-3-319-24258-3_2
- Campbell, P. F. (1987). Measuring distance: Children's use of number and unit. *Final report submitted to the National Institute of Mental Health Under the ADAMHA Small Grant Award Program. Grant No. MSMA, 1, R03*.

- Carlson, K. D., & Herdman, A. O. (2012). Understanding the impact of convergent validity on research results. *Organizational Research Methods*, 15(1), 17-32.
- Carr, N. G. (2011). *Superficiales: ¿Qué está haciendo Internet con nuestras mentes?* Barcelona: Taurus.
- Cartelli, A., Dagiene, V., & Futschek, G. (2012). Bebras contest and digital competence assessment: Analysis of frameworks. En A. Cartelli (Ed.), *Current Trends and Future Practices for Digital Literacy and Competence* (pp. 35-46). Hershey, PA: IGI Global.
- CAS, Computing at School (2015). *CAS Barefoot* [Página web] Recuperado de <http://barefootcas.org.uk/>
- Caselles Pérez, J. F. (1997). Pedagogía diferencial, sexo y educación: El sexismo en el sistema educativo mixto y la coeducación como alternativa. En C. Jiménez Fernández (Coord.), *Lecturas de Pedagogía Diferencial* (pp. 135-170). Madrid: Dykinson.
- Cattell, R. B. (1963). Theory of fluid and crystallized intelligence: a critical experiment. *Journal of Educational Psychology*, 54, 1-22.
- Chamorro-Premuzic, T., & Furnham, A. (2008). Personality, intelligence and approaches to learning as predictors of academic performance. *Personality and Individual Differences*, 44(7), 1596-1603. doi: <http://dx.doi.org/10.1016/j.paid.2008.01.003>
- Chan, Y., Hui, D., Dickinson, A. R. et al. (2010). Engineering outreach: A successful initiative with gifted students in science and technology in Hong Kong. *IEEE Transactions on Education*, 53(1), 158-171.
- Cheng, P. (1993). Metacognition and giftedness: The state of the relationship. *Gifted Child Quarterly*, 37(3), 105-12.
- Chilcott, L. (2015). *CodeGirl* [Documental de vídeo] Recuperado de <http://www.codegirlmovie.com/>
- Clark, L. (2005). Gifted and growing. *Educational Leadership*, 63(3), 56-60.
- Clements, D. H. (1985). Research on Logo in education: Is the turtle slow but steady, or not even in the race? *Computers in the Schools*, 2(2-3), 55-71.
- Clements, D. H. (1986). Effects of Logo and CAI environments on cognition and creativity. *Journal of Educational Psychology*, 78(4), 309-318. doi: <http://dx.doi.org/10.1037/0022-0663.78.4.309>
- Clements, D. H., & Battista, M. T. (1989). Learning of geometric concepts in a Logo environment. *Journal for Research in Mathematics Education*, 20(5), 450-467.
- Clements, D. H., & Gullo, D. F. (1984). Effects of computer programming on young children's cognition. *Journal of Educational Psychology*, 76(6), 1051-1058. doi: <http://dx.doi.org/10.1037/0022-0663.76.6.1051>

- Clements, D. H., & Meredith, J. S. (1993). Research on Logo: Effects and efficacy. *Journal of Computing in Childhood Education*, 4(4), 263-290.
- Clements, D. H., & Merriman, S. (1988). Componential developments in Logo programming and environments. En R. E. Mayer (Ed.), *Teaching and learning computer programming* (pp. 13-54). Hillsdale, NJ: Erlbaum.
- Clements, D. H., & Sarama, J. (1997). Research on Logo: A decade of progress. *Computers in the Schools*, 14(1-2), 9-46.
- Clements, D. H., & Swaminathan, S. (1995). Technology and School Change New Lamps for Old? *Childhood Education*, 71(5), 275-281. doi: <http://dx.doi.org/10.1080/00094056.1995.10522619>
- Crombie, G., Bouffard-Bouchard, T., & Schneider, B.H. (1992). Gifted Programs: Gender Differences in Referral and Enrollment. *Gifted Child Quarterly*, 36(4), 213-214.
- Code.org (2013). *What Most Schools Don't Teach* [Archivo de vídeo] Recuperado de <https://youtu.be/nKIu9yen5nc>
- Code.org (2014). *Pair Programming* [Archivo de vídeo] Recuperado de <https://youtu.be/vgkahOzFH2Q>
- Code.org (2015a). *K-8 Intro to Computer Science* [Curso en línea] Recuperado de <https://studio.code.org/s/20-hour>
- Code.org (2015b). *K-5 Computer Science Fundamentals (Course 2)* [Curso en línea] Recuperado de <https://studio.code.org/s/course2>
- Code.org (2015c). *The Hour of Code* [Página web] Recuperado de <https://hourofcode.com>
- Code.org (2015d). *Instructor Handbook: Code Studio Lesson Plans for Courses One, Two, and Three* [Documento en línea] Recuperado de <https://code.org/curriculum/docs/k-5/complete.pdf>
- Cohen, L., Manion, L., & Morrison, K. (2007). *Research Methods in Education*. New York: Routledge.
- Colás, P., & Buendía, L. (1992). *Investigación Educativa*. Sevilla: Ediciones Alfar.
- Colás-Bravo, P., González-Ramírez, T., & de Pablos-Pons, J. (2013). Juventud y redes sociales: Motivaciones y usos preferentes. *Comunicar: Revista Científica De Comunicación y Educación*, 20(40), 15-23. doi: <http://dx.doi.org/10.3916/C40-2013-02-01>
- Compañ-Rosique, P., Satorre-Cuerda, R., Llorens-Largo, F., & Molina-Carmona, R. (2015). Enseñando a programar: un camino directo para desarrollar el pensamiento computacional. *RED, Revista de Educación a Distancia*, 46. Recuperado de http://www.um.es/ead/red/46/faraon_et_al.pdf

- Connolly, T. (2010). Review of outliers: The story of success. *Journal of Behavioral Decision Making*, 23(3), 331-333. doi: <http://dx.doi.org/10.1002/bdm.693>
- Cooper, S., Dann, W., & Pausch, R. (2000). Alice: A 3-D tool for introductory programming concepts. *Journal of Computing Sciences in Colleges*, 15(5), 107-116.
- Cope, E. W., & Suppes, P. (2002). Gifted students' individual differences in distance-learning computer-based calculus and linear algebra. *Instructional Science*, 30(2), 79-110.
- Coxon, S. V. (2012). Innovative allies: Spatial and creative abilities. *Gifted Child Today*, 35(4), 277-284.
- Crawley, D. (3 de junio de 2014). *12 games that teach kids to code — and are even fun, too* [Mensaje en un blog] Recuperado de <http://venturebeat.com/2014/06/03/12-games-that-teach-kids-to-code/view-all/>
- Cronin, A. (14 de julio de 2014). *Teach Coding in the Classroom: Resources from ISTE '14* [Mensaje en un blog] Recuperado de <http://www.edutopia.org/blog/teach-coding-classroom-resources-iste-14>
- CSTA (2011). *K–12 Computer Science Standards* [Documento en línea]. Recuperado de http://csta.acm.org/Curriculum/sub/CurrFiles/CSTA_K-12_CSS.pdf
- CSTA & ISTE (2015). *Operational Definition of Computational Thinking for K–12 Education* [Documento en línea]. Recuperado de <http://csta.acm.org/Curriculum/sub/CurrFiles/CompThinkingFlyer.pdf>
- Curzon, P., & McOwan, P. (2015). *The Magic of Computer Science* [Documento en línea] Recuperado de <http://www.cs4fn.org/magic/downloads/cs4fnmagicbook3.pdf>
- Cuthbertson, A. (2 de septiembre de 2014). Coding in the Classroom: Computational Thinking Will Allow Children to ‘Change the World’. *InternationalBusinessTimes.com*. Recuperado de <http://www.ibtimes.co.uk/coding-classroom-computational-thinking-will-allow-children-change-world-1463493>
- Dagiene, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. En R. T. Mittermeir & M. M. Sysło (Eds.), *Informatics education-supporting computational thinking* (pp. 19-30). Berlín: Springer.
- Dagiene, V., & Stupuriene, G. (2014). Informatics education based on solving attractive tasks through a contest. *Proceedings of KEYCIT 2014–Key Competencies in Informatics and ICT*, 51-62.
- Daily, S. B., Leonard, A. E., Jörg, S., Babu, S., & Gundersen, K. (2014). Dancing Alice: Exploring embodied pedagogical strategies for learning computational thinking. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 91-96.

- Dans, E. (28 de septiembre de 2013). *Programación, niños y escuelas: el reto del momento* [Mensaje en un blog]. Recuperado de <http://www.enriquedans.com/2013/09/programacion-ninos-y-escuelas-el-reto-del-momento.html>
- Dans, E. (23 de mayo de 2015a). *Educación y tecnología: aterrizando conceptos* [Mensaje en un blog] Recuperado de <http://www.enriquedans.com/2015/05/educacion-y-tecnologia-aterrizando-conceptos.html>
- Dans, E. (17 de septiembre de 2015b). *Facebook y el supuesto botón “no me gusta”* [Mensaje en un blog] Recuperado de <http://www.enriquedans.com/2015/09/facebook-y-el-supuesto-boton-no-me-gusta.html>
- Dans, E. (27 de enero de 2016). *No envíes a un humano a hacer el trabajo de una máquina* [Mensaje en un blog] Recuperado de <http://www.enriquedans.com/2016/01/no-envies-a-un-humano-a-hacer-el-trabajo-de-una-maquina.html>
- De Wet, C. F. (2006). Beyond presentations: Using PowerPoint as an effective instructional tool. *Gifted Child Today*, 29(4), 29-39.
- DECRETO 48/2015 (2015). Decreto 48/2015, de 14 de mayo, del Consejo de Gobierno, por el que se establece para la Comunidad de Madrid el currículo de la Educación Secundaria Obligatoria. *Boletín Oficial de la Comunidad de Madrid*, 20/05/2015. Recuperado de http://www.bocm.es/boletin/CM_Orden_BOCM/2015/05/20/BOCM-20150520-1.PDF
- Denner, J., Werner, L., & Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1), 240-249. doi: <http://dx.doi.org/10.1016/j.compedu.2011.08.006>
- DiBello, L. V., & Stout, W. (2007). Guest editors' introduction and overview: IRT-Based cognitive diagnostic models and related methods. *Journal of Educational Measurement*, 44(4), 285-291.
- Dickes, A. C., & Sengupta, P. (2013). Learning natural selection in 4th grade with multi-agent-based computational models. *Research in Science Education*, 43(3), 921-953.
- Díez, A., Santiago, R., & Climent, J. (2015). *Eduapps* [Página web] Recuperado de <http://www.eduapps.es/>
- Dorling, M. (2015). *CAS Computing Progression Pathways KS1 (Y1) to KS3 (Y9) by topic* [Documento en línea] Recuperado de <http://community.computingschool.org.uk/resources/1692>
- Dove, M. K., & Zitkovich, J. A. (2003). Technology driven group investigations for gifted elementary students. *Information Technology in Childhood Education Annual*, 15, 223-241.
- Dredge, S. (27 de enero de 2014). Tablets in schools: coding, creativity and the importance of teachers. *TheGuardian.com*. Recuperado de

<http://www.theguardian.com/technology/2014/jan/27/tablets-schools-coding-kids-education-ipad>

- Dubinski, K. (04 de diciembre de 2014). An Hour of Code, a link to future. *Lfpres.com*. Recuperado de <http://www.lfpres.com/2014/12/04/an-hour-of-code-a-link-to-future>
- Eckstein, M. (2009). Enrichment 2.0 gifted and talented education for the 21st century. *Gifted Child Today*, 32(1), 59-63.
- EdTech Team (17 de noviembre de 2014). *Teaching coding in class: 17 apps to try* [Mensaje en un blog] Recuperado de <http://www.educatorstechnology.com/2014/11/teaching-coding-in-class-17-apps-to-try.html>
- Elder, J. (26 de junio de 2015). Bangalore Girls Who Code Pitch Their Waste Killer App in San Francisco. *TheWallStreetJournal.com*. Recuperado de <http://blogs.wsj.com/indiarealtime/2015/06/26/bangalore-girls-who-code-pitch-their-waste-killer-app-in-san-francisco/>
- Ellis, P. D. (2010). *The essential guide to effect sizes: Statistical power, meta-analysis, and the interpretation of research results*. Cambridge, MA: Cambridge University Press.
- Epstein, R., Schmidt, S. M., & Warfel, R. (2008). Measuring and training creativity competencies: Validation of a new test. *Creativity Research Journal*, 20(1), 7-12.
- Eristi, B. (2012). To learn from teachers at school, ideal teacher or E-learning applications from the perspectives of gifted students. *Turkish Online Journal of Distance Education*, 13(4), 153-166.
- Espeso, P. (16 de noviembre de 2015). CS Unplugged: aprendiendo programación en el aula... ¡sin ordenadores! *Educación 3.0*. [Artículo en línea] Recuperado de <http://www.educaciontrespuntocero.com/formacion/libros/cs-unplugged-aprendiendo-programacion-en-el-aula-sin-ordenadores/30076.html>
- European Commission (2014). *E-Skills for Jobs in Europe: measuring progress and moving ahead* [Informe técnico] Recuperado de <http://eskills-monitor2013.eu/results/>
- European Schoolnet (2015). *Computing our future. Computer programming and coding: priorities, school curricula and initiatives across Europe* [Informe técnico]. Recuperado de http://www.eun.org/c/document_library/get_file?uuid=3596b121-941c-4296-a760-0f4e4795d6fa&groupId=43887
- Fessakis, G., Gouli, E., & Mavroudi, E. (2013). Problem solving by 5–6 years old kindergarten children in a computer programming environment: A case study. *Computers & Education*, 63, 87-97.
- Festinger, L. (1957). *A theory of cognitive dissonance*. Stanford, CA: Stanford University Press.
- Fincher, S., & Petre, M. (2004). *Computer science education research*. London: Taylor & Francis.

- Finkel, E. (2012). *Flipping the Script in K12* [Documento en línea] Recuperado de <http://www.districtadministration.com/article/flipping-script-k12>
- Freeman, J. (1990). Educating gifted children. En N. Entwistle (ed.), *Handbook of Educational Ideas and Practices* (pp. 939-947). London: Routledge.
- Frenkel, K. A. (1990). Women and computing. *Communications of the ACM*, 33(11), 34-46. doi: <http://dx.doi.org/10.1145/92755.92756>
- Fundación Telefónica (2014). *La Sociedad de la Información en España 2014* [Informe técnico] Recuperado de http://www.fundaciontelefonica.com/arte_cultura/publicaciones-listado/pagina-item-publicaciones/?itempubli=323
- FUNED (2014). *Desarrollo avanzado de aplicaciones en iOS7 (Héroe de iOS)* [Curso en línea] Recuperado de <http://www.fundacion.uned.es/actividad/idactividad/5766>
- FUNED (2015). *Desarrollo Rápido de Aplicaciones Android en el Aula de ESO, Bachillerato y FP* [Curso en línea] Recuperado de https://formacionpermanente.uned.es/tp_actividad/idactividad/8018
- Gadanidis, G., Hughes, J., & Cordy, M. (2011). Mathematics for gifted students in an Arts and Technology rich setting. *Journal for the Education of the Gifted*, 34(3), 397-433.
- Gagné, F. (1991). Toward a differentiated model of giftedness and talent. En N. Colangelo & G. A. Davis, *Handbook of Gifted Education* (pp. 65-80). Massachusetts: Allyn and Bacon.
- Gagné, F. (2004). Transforming gifts into talents: The DMGT as a developmental theory. *High Ability Studies*, 15(2), 119-147.
- Gagné, F. (2015). De los genes al talento: La perspectiva DMGT/CMTD. *Revista de Educación*, 368, 12-39. <http://dx.doi.org/10.4438/1988-592X-RE-2015-368-289>
- García, D. D., Ding, W., Cohen, J., Ericson, B., Gray, J., & Reed, D. (2015). One-day activities for K-12 face-to-face outreach. *Proceedings of the 46th ACM Technical Symposium on Computer Science Education*, 520-521. doi: <http://dx.doi.org/10.1145/2676723.2677337>
- García Hoz, V. (1991). El principio de complementariedad en la investigación pedagógica y en la educación personalizada. *Anales de la Real Academia de Ciencias Morales y Políticas*, 68, 299-318.
- García Llamas, J. L. (2003). *Métodos de Investigación en Educación. Volumen II: Investigación Cualitativa y Evaluativa*. Madrid: UNED.
- Gardner, H. (1983). *Frames of Mind: The Theory of Multiple Intelligences*. New York: Basic Books.
- Gardner, H. (1995). *Inteligencias múltiples. La teoría en la práctica*. Barcelona: Paidós.
- Gardner, H. (1999). *Intelligence Reframed*. New York: Basic Books.

- Gardner, H., & Davis, K. (2014). *La generación APP: Cómo los jóvenes gestionan su identidad, su privacidad y su imaginación en el mundo digital*. Barcelona: Paidós.
- Genishi, C., McCollum, P., & Strand, E. (1985). Research currents: The interactional richness of children's computer use. *Language Arts*, 62(5), 526-532.
- Gentry, J. (2008). E-publishing's impact on learning in an inclusive sixth grade social studies classroom. *Journal of Interactive Learning Research*, 19(3), 455-467.
- Gentry, J., Fowler, T., & Nichols, B. (2007). Textbook preferences: The possibilities of individualized learning in social studies with an individualized textbook. *Journal of Interactive Learning Research*, 18(4), 493-510.
- Gil Pascual, J. A. (2003). *Métodos de Investigación en Educación. Volumen III: Análisis Multivariante*. Madrid: UNED.
- Gilbert-Macmillan, K. (2000). Computer-based distance learning for gifted students: The EPGY experience. *Understanding our Gifted*, 12(3), 17-20.
- Gombachika, H. S., & Khangamwa, G. (2013). ICT readiness and acceptance among TEVT students in University of Malawi. *Campus-Wide Information Systems*, 30(1), 35-43. doi: <http://dx.doi.org/10.1108/10650741311288805>
- Goodwin, B., & Miller, K. (2013). Research Says Evidence on Flipped Classrooms Is Still Coming In. *Technology-Rich Learning*, 70(6), 78-80. Recuperado de <http://www.ascd.org/publications/educational-leadership/mar13/vol70/num06/Evidence-on-Flipped-Classrooms-Is-Still-Coming-In.aspx>
- Google for Education (2015). *Exploring Computational Thinking* [Página web]. Recuperado de <https://www.google.com/edu/resources/programs/exploring-computational-thinking/>
- Gorman, H., & Bourne, L. E. (1983). Learning to think by learning LOGO: Rule learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 21(3), 165-167.
- Gould, J. C., Staff, L. K., & Theiss, H. M. (2012). The right fit for Henry. *Educational Leadership*, 69(5), 71-73.
- Gouws, L. A., Bradshaw, K., & Wentworth, P. (2013a). Computational thinking in educational activities: An evaluation of the educational game light-bot. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 10-15.
- Gouws, L., Bradshaw, K., & Wentworth, P. (2013b). First year student performance in a test for computational thinking. *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, 271-277.
- Graczyńska, E. (2010). ALICE as a tool for programming at schools. *Natural Science*, 2(2), 124-129.
- Grimes, D., & Warschauer, M. (2008). Learning with laptops: A multi-method case study. *Journal of Educational Computing Research*, 38(3), 305-332.

- Grover, S. (2011). Robotics and engineering for middle and high school students to develop computational thinking. *Annual Meeting of the American Educational Research Association, New Orleans (LA)*
- Grover, S., & Pea, R. (2013a). Computational thinking in K–12. A review of the state of the field. *Educational Researcher*, 42(1), 38-43.
- Grover, S., & Pea, R. (2013b). Using a discourse-intensive pedagogy and Android's App Inventor for introducing computational concepts to middle school students. *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, 723-728. doi: <http://dx.doi.org/10.1145/2445196.2445404>
- Guilford, J. P. (1967). *La naturaleza de la inteligencia humana*. Barcelona: Paidós.
- Gwet, K. L. (2014). *Handbook of inter-rater reliability: The definitive guide to measuring the extent of agreement among raters*. Gaithersburg, MD: Advanced Analytics, LLC.
- Haier, R. J. (2014). Increased intelligence is a myth (so far). *Frontiers in Systems Neuroscience*, 8, 34. doi: <http://dx.doi.org/10.3389%2Ffnsys.2014.00034>
- Hamada, R. M. (1987). The relationship between learning Logo and proficiency in mathematics. *Dissertation Abstracts International*, 47, 2510-A.
- Hamdan, N, McKnight, P., McKnight, K., & Arfstrom, K. M. (2013). *A White Paper Based on the Literature Review of Flipped Learning* [Informe técnico]. Recuperado de http://www.flippedlearning.org/cms/lib07/VA01923112/Centricity/Domain/41/WhitePaper_FlippedLearning.pdf
- Han, B. C. (2014). *En el enjambre*. Barcelona: Herder.
- Han, B. C. (2015). *La salvación de lo bello*. Barcelona: Herder.
- Hatch, M. (2014). *The Maker Movement Manifesto*. New York: McGraw Hill Education
- Hauser-Reynolds, R. (2015). *CODE: Debugging the gender gap* [Documental de video] Recuperado de <http://www.codedocumentary.com/>
- Hawke, L. (10 de diciembre de 2014). Hour of Code: Young Australian students explore computer science, as numbers of IT graduates shrink. *ABC.net.au*. Recuperado de <http://mobile.abc.net.au/news/2014-12-09/hour-of-code-australian-students-explore-computer-science/5954282>
- Henderson, P. B., Cortina, T. J., & Wing, J. M. (2007). Computational thinking. *ACM SIGCSE Bulletin*, 39(1), 195-196.
- Hernández Jorge, C. M., & Borges del Rosal, A. (2005). Un programa de aprendizaje autorregulado para personas de altas capacidades mediante herramientas telemáticas. *Revista Electrónica de Investigación Psicoeducativa*, 3(3), 233-252.

- Hidalgo, M. (13 de febrero de 2013). Un aula en la web para «alumnos del siglo XXI». *DiarioVasco.com*. Recuperado de <http://www.diariovasco.com/20130217/local/aula-para-alumnos-siglo-201302131008.html>
- Himanen, P. (2004). *La ética del hacker y el espíritu de la era de la información*. Barcelona: Destino.
- Hines, S. N. (1983). Computer programming abilities of five-year-old children. *Educational Computer*, 3(4), 10-12.
- Holmboe, C., McIver, L., & George, C. (2001). Research agenda for computer science education. *Proceedings of the 13th Workshop of the Psychology of Programming Interest Group*, 207-233.
- Hoover, A. K., & Stanley, K. O. (2009). Exploiting functional relationships in musical composition. *Connection Science*, 21(2-3), 227-251. doi: <http://dx.doi.org/10.1080/09540090902733871>
- Howland, K., & Good, J. (2015). Learning to communicate computationally with flip: A bi-modal programming language for game creation. *Computers & Education*, 80, 224-240. doi: <http://dx.doi.org/10.1016/j.compedu.2014.08.014>
- Huang, L., Cui, Y., Zhang, D., & Wu, S. (2011). Impact of noise structure and network topology on tracking speed of neural networks. *Neural Networks*, 24(10), 1110-1119. doi: <http://dx.doi.org/10.1016/j.neunet.2011.05.018>
- Hubwieser, P., & Mühlhng, A. (2014). Playing PISA with Bebras. *Proceedings of the 9th Workshop in Primary and Secondary Computing Education*, 128-129. doi: <http://dx.doi.org/10.1145/2670757.2670759>
- Hubwieser, P., & Mühlhng, A. (2015). Investigating the psychometric structure of Bebras contest: Towards measuring computational thinking skills. *International Conference on Learning and Teaching in Computing and Engineering (LaTiCE)*, 62-69. doi: <http://dx.doi.org/10.1109/LaTiCE.2015.19>
- Hyde, J. S., & Linn, M. C. (1988). Gender differences in verbal ability: A meta-analysis. *Psychological Bulletin*, 104(1), 53.
- INTEF (20 de octubre de 2014). MOOC “Las preguntas de PISA y TIMSS: una práctica para el aula”, inscripción abierta [Mensaje en un blog]. Recuperado de <http://blog.educalab.es/intef/2014/10/20/mooc-las-preguntas-de-pisa-y-timss-una-practica-para-el-aula-inscripcion-abierta/>
- Jašková, L., & Kováčová, N. (2015). Bebras contest for blind pupils. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 92-95. doi: <http://dx.doi.org/10.1145/2818314.2818324>

- Jee, C. (18 de mayo de 2015). Coding languages: which ones pay most in Europe? *TechWorld.com*. Recuperado de <http://www.techworld.com/picture-gallery/careers/coding-languages-which-ones-pay-most-in-europe-3611877/>
- Jenkins, H. (2010). Transmedia storytelling and entertainment: An annotated syllabus. *Continuum: Journal of Media & Cultural Studies*, 24(6), 943-958.
- Jiménez Cano, R. (26 de abril de 2013). Google no puede con la App Store. *El País.com*. Recuperado de http://tecnologia.elpais.com/tecnologia/2013/04/26/actualidad/1366971919_151325.html
- Jiménez Fernández, C. (1997). Educación de los más dotados. *Revista de Investigación Educativa*, 15(2), 217-234.
- Jiménez Fernández, C. (2010). *Diagnóstico y educación de los más capaces*. Madrid: Pearson Educación.
- Jiménez Fernández, C. (02 de octubre de 2014). *El Desarrollo del Talento: Educación y Alta Capacidad* [Lección inaugural: UNED] Recuperado de https://canal.uned.es/uploads/material/Video/21562/LECCI__N__INAUGURAL14.PDF
- Jiménez Fernández, C., & García Perales, R. (2013). Los alumnos más capaces en España. Normativa e incidencia en el diagnóstico y la educación. *Revista Española de Orientación y Psicopedagogía (REOP)*, 24(1), 7-24.
- Jiménez Fernández, C., & González Galán, M. A. (2010). *Pedagogía Diferencial y Atención a la Diversidad*. Madrid: UNED.
- Johnson-Gentile, K., Clements, D. H., & Battista, M. T. (1994). Effects of computer and noncomputer environments on students' conceptualizations of geometric motions. *Journal of Educational Computing Research*, 11(2), 121-140.
- Jonassen, D., Howland, J., Marra, R.M., & Crismond, D. (2008). *Meaningful learning with technology (3rd ed.)*. Boston, MA: Prentice Hall.
- Kafai, Y., & Burke, Q. (2013). Computer programming goes back to school. *Phi Delta Kappan*, 95(1), 61-65.
- Kafai, Y., & Burke, Q. (2014). *Connected Code: Why Children Need to Learn Programming*. Cambridge, MA: MIT Press.
- Kafai, Y., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. New York: Routledge.
- Kafai, Y., & Vasudevan, V. (2015). Constructionist gaming beyond the screen: Middle school students' crafting and computing of touchpads, board games, and controllers. *Proceedings of the Workshop in Primary and Secondary Computing Education WiPSCE 2015*, 49-54. doi: <http://dx.doi.org/10.1145/2818314.2818334>

- Kahn, K., Sendova, E., Sacristán, A. I., & Noss, R. (2011). Young students exploring cardinality by constructing infinite processes. *Technology, Knowledge and Learning*, 16(1), 3-34.
- Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org. *Computers in Human Behavior*, 52, 200-210. doi:<http://dx.doi.org/10.1016/j.chb.2015.05.047>
- Kalelioğlu, F., Gülbahar, Y., & Madran, O. (2015). A snapshot of the first implementation of Bebras international informatics contest in Turkey. En A. Brodник & J. Vahrenhold (Eds.), *Informatics in schools. Curricula, competences, and competitions* (pp. 131-140). Berna: Springer.
- Kazakoff, E., & Bers, M. (2012). Programming in a robotics context in the kindergarten classroom: The impact on sequencing skills. *Journal of Educational Multimedia and Hypermedia*, 21(4), 371-391.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers & Education*, 73, 26-39. doi: <http://dx.doi.org/10.1016/j.compedu.2013.12.010>
- Kelleher, C., & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83-137.
- Kelleher, C., Pausch, R., & Kiesler, S. (2007). Storytelling Alice motivates middle school girls to learn computer programming. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 1455-1464.
- Khan, S. (2011). *Usemos el vídeo para reinventar la educación* [Archivo de vídeo] Recuperado de https://www.ted.com/talks/salman_khan_let_s_use_video_to_reinvent_education?language=es
- Khan Academy (2015). *Computer Programming* [Curso en línea] Recuperado de <https://www.khanacademy.org/computing/computer-programming>
- Khorram, Y. (21 de enero de 2014). She's a beauty and a geek: Supermodel is a coder. *CNN.com*. Recuperado de <http://edition.cnn.com/2014/01/20/tech/web/lyndsey-scott-model-coder/index.html>
- Kim, M. (2011). The relationship between thinking style differences and career choices for high-achieving students. *Roeper Review: A Journal on Gifted Education*, 33(4), 252-262. doi: <http://dx.doi.org/10.1080/02783193.2011.603113>
- Kim, S., Chung, K., & Yu, H. (2013). Enhancing digital fluency through a training program for creative problem solving using computer programming. *The Journal of Creative Behavior*, 47(3), 171-199. doi: <http://dx.doi.org/10.1002/jocb.30>
- Klawe, M., Whitney, T., & Simard, C. (2009). Women in computing---take 2. *Communications of the ACM*, 52(2), 68-76. doi: <http://dx.doi.org/10.1145/1461928.1461947>

- Kline, P. (1994). *An easy guide to factor analysis*. London: Routledge.
- Koh, K. H., Basawapatna, A., Bennett, V., & Repenning, A. (2010). Towards the automatic recognition of computational thinking for adaptive visual language learning. *Visual Languages and Human-Centric Computing (VL/HCC), 2010 IEEE Symposium*, 59-66.
- Koivunen, N. (2009). Collective expertise: Ways of organizing expert work in collective settings. *Journal of Management & Organization*, 15(2), 258-276. doi: <http://dx.doi.org/10.1017/S1833367200002820>
- Lai, C. S., & Lai, M. H. (2012). Using Computer Programming to Enhance Science Learning for 5th Graders in Taipei. In *Computer, Consumer and Control (IS3C), 2012 International Symposium on* (pp. 146–148). Taichung, Taiwan: IEEE. doi: <http://dx.doi.org/10.1109/IS3C.2012.45>
- Lanier, J. (2011). *Contra el rebaño digital*. Barcelona: Debate.
- Lankshear, C., & Knobel, M. (2008). *Nuevos alfabetismos. Su práctica cotidiana y el aprendizaje en el aula*. Madrid: Morata.
- Larson, R. C., & Murray, M. E. (2008). Open educational resources for blended learning in high schools: Overcoming impediments in developing countries. *Journal of Asynchronous Learning Networks*, 12(1), 85-103.
- Layton, L. (14 de enero de 2014). Successful 'Hour of Code' computer tutorials prompts effort to change school policies. *WashingtonPost.com*. Recuperado de https://www.washingtonpost.com/local/education/successful-hour-of-code-computer-tutorials-prompts-effort-to-change-school-policies/2014/01/14/f224f112-7d5f-11e3-95c6-0a7aa80874bc_story.html
- Lee, G., Lin, Y., & Lin, J. (2014). Assessment of computational thinking skill among high school and vocational school students in Taiwan. *World Conference on Educational Multimedia, Hypermedia and Telecommunications*, 173-180.
- Lee, Y. (2010). Developing computer programming concepts and skills via technology-enriched language-art projects: A case study. *Journal of Educational Multimedia and Hypermedia*, 19(3), 307-326.
- Lee, Y. (2011). Scratch: Multimedia programming environment for young gifted learners. *Gifted Child Today*, 34(2), 26-31.
- Lehrer, R., & DeBernard, A. (1987). Language of learning and language of computing: The perceptual-language model. *Journal of Educational Psychology*, 79(1), 41-48. doi: <http://dx.doi.org/10.1037/0022-0663.79.1.41>
- Lehrer, R., Lee, M., & Jeong, A. (1999). Reflective teaching of Logo. *The Journal of the Learning Sciences*, 8(2), 245-289.

- Leonard, A. E., Dsouza, N., Babu, S. V., Daily, S. B., Jörg, S., Waddell, C., ... , & Boggs, K (2015). Embodying and programming a constellation of multimodal literacy practices: Computational thinking, creative movement, biology, & virtual environment interactions. *Journal of Language and Literacy Education* 11(2), 64-93.
- Levy, S. (1984). *Hackers: Heroes of the Computer Revolution*. New York: Doubleday
- Lewis, C. M., & Shah, N. (2012). Building upon and enriching grade four mathematics standards with programming curriculum. *Proceedings of the 43rd ACM technical symposium on Computer Science Education* (pp. 57–62). Raleigh, NC, USA: ACM. doi: <http://dx.doi.org/10.1145/2157136.2157156>
- Lin, J. M., & Liu, S. (2012). An investigation into parent-child collaboration in learning computer programming. *Journal of Educational Technology & Society*, 15(1), 162-173.
- Linn, M. C., & Petersen, A. C. (1985). Emergence and characterization of sex differences in spatial ability: A meta-analysis. *Child Development*, 56(6), 1479-1498.
- LOCE (2002). Ley Orgánica 10/2002, de 23 de diciembre, de Calidad de la Educación. *Boletín Oficial del Estado*, 24/12/2002.
- LOE (2006). Ley Orgánica 2/2006, de 3 de mayo, de Educación. *Boletín Oficial del Estado*, 04/05/2006.
- LOGSE (1990). Ley 1/1990, de 3 de octubre, de Ordenación General del Sistema Educativo. *Boletín Oficial del Estado*, 04/10/1990.
- LOMCE (2013). Ley 8/2013, de 9 de diciembre, para la Mejora de la Calidad Educativa. *Boletín Oficial del Estado*, 10/12/2013
- Lye, S. Y., & Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior*, 41, 51-61. doi: <http://dx.doi.org/10.1016/j.chb.2014.09.012>
- Madrid, I. (15 de junio de 2015). Two high schoolers create Tampon Run video game to take the taboos out of menstruation. *PRI.org*. Recuperado de <http://www.pri.org/stories/2015-06-15/two-high-schoolers-create-tampon-run-video-game-take-taboos-out-menstruation>
- Maloney, J., Peppler, K., Kafai, Y., Resnick, M., & Rusk, N. (2008). Programming by choice: Urban youth learning programming with scratch. *ACM SIGCSE Bulletin*, 40(1), 367-371.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16. doi: <http://dx.doi.org/10.1145/1868358.1868363>
- Mann, E. L. (2006). Creativity: The essence of mathematics. *Journal for the Education of the Gifted*, 30(2), 236-260.
- Manovich, L. (2013). *El software toma el mando*. Barcelona: UOC Press.

- Marais, C., & Bradshaw, K. (2015). Problem-solving ability of first year CS students: A case study and intervention. *Proceedings of Southern African Computer Lecturers' Association*, 154-160.
- Massachusetts Institute of Technology (2015). *MIT – App Inventor* [Curso en línea] Recuperado de <http://appinventor.mit.edu/explore/>
- Mayer, R. E. (2004). Should there be a three-strikes rule against pure discovery learning? *American Psychologist*, 59(1), 14-19.
- McAllister, A. (1993). Representing the programming process: Goal structures and action sequences in LOGO programming. *Annual Meeting of the American Educational Research Association*. Atlanta (April 12-16, 1993). Recuperado de <http://files.eric.ed.gov/fulltext/ED364195.pdf>
- McGraw-Hill Higher Education (13 de junio de 2013). *Flipping the Classroom Panel Discussion* [Archivo de vídeo] Recuperado de <https://youtu.be/3T8jzCJza0>
- McGrew, K. S. (2009). CHC theory and the human cognitive abilities project: Standing on the shoulders of the giants of psychometric intelligence research. *Intelligence*, 37(1), 1-10. doi: <http://dx.doi.org/10.1016/j.intell.2008.08.004>
- McLester, S. (2012). Sally Reis and Joe Renzulli: Models for Education Reform, part 2. *District Administration*, 48(5), 68-72.
- McLuhan, M. (1994). *Understanding media: The extensions of man*. Cambridge, MA: MIT Press.
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. M. (2010). Learning computer science concepts with Scratch. *Proceedings of the Sixth International Workshop on Computing Education Research*, 69-76. doi: <http://dx.doi.org/10.1145/1839594.1839607>
- Microsoft Research (2015). *Touch Develop* [Curso en línea] Recuperado de <https://www.touchdevelop.com/>
- Miller, L. D., Soh, L., Chiriacescu, V., Ingraham, E., Shell, D. F., ..., & Hazley, M. P. (2013). Improving learning of computational thinking using creative thinking exercises in CS-1 computer science courses. *Frontiers in Education Conference, 2013 IEEE*, 1426-1432.
- Miller, L. D., Soh, L., Chiriacescu, V., Ingraham, E., Shell, D. F., & Hazley, M. P. (2014). Integrating computational and creative thinking to improve learning and performance in CS1. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 475-480.
- Miller, P. (2009). Learning with a missing sense: What can we learn from the interaction of a deaf child with a turtle? *American Annals of the Deaf*, 154(1), 71-82.
- Mills, K. A. (2010). A review of the 'digital turn' in the new literacy studies. *Review of Educational Research*, 80(2), 246-271.

- Miriada X (2015). *Emprendimiento y App Inventor* [Curso en línea] Recuperado de <https://miriadax.net/web/emprendimiento-y-app-inventor1>
- MIT Media Lab (2015). *Scratch* [Plataforma en línea] Recuperado de <https://scratch.mit.edu/>
- Montero, I. & León, O. G. (2007). A guide for naming research studies in psychology. *International Journal of Clinical and Health Psychology*, 7(3), 847-862.
- Moreno-León, J., & Robles, G. (2014). Automatic detection of bad programming habits in Scratch: A preliminary study. *Frontiers in Education Conference (FIE), 2014 IEEE*, 1-4. doi: <http://dx.doi.org/10.1109/FIE.2014.7044055>
- Moreno-León, J., & Robles, G. (2015). Analyze your Scratch projects with Dr. Scratch and assess your computational thinking skills. *Scratch Conference*, 12-15.
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: análisis automático de proyectos Scratch para evaluar y fomentar el Pensamiento Computacional. *RED, Revista de Educación a Distancia*, 46. Recuperado de http://www.um.es/ead/red/46/moreno_robles.pdf
- Morozov, E. (2015). *La locura del solucionismo tecnológico*. Madrid: Clave Intelectual
- Morrison, B. B., & DiSalvo, B. (2014). Khan Academy gamifies computer science. *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, 39-44. doi: <http://dx.doi.org/10.1145/2538862.2538946>
- Morrison, N. (6 de julio de 2014). The primary getting kids into binary. *TheTelegraph.co.uk*. Recuperado de <http://www.theguardian.com/technology/2014/jan/27/tablets-schools-coding-kids-education-ipad>
- Mozilla Webmaker (2015). *Estándares de alfabetización web* [Documento en línea] Recuperado de <https://teach.mozilla.org/activities/web-literacy/>
- Mühling, A., Ruf, A., & Hubwieser, P. (2015). Design and first results of a psychometric test for measuring basic programming abilities. *Proceedings of the Workshop in Primary and Secondary Computing Education*, 2-10. doi: <http://dx.doi.org/10.1145/2818314.2818320>
- Muñiz, J. (1992). *Teoría clásica de los tests*. Madrid: Pirámide.
- Muñoz-Merino, P. J., Valiente, J. A. R., & Kloos, C. D. (2013). Inferring higher level learning information from low level data for the khan academy platform. *Proceedings of the Third International Conference on Learning Analytics and Knowledge*, 112-116. doi: <http://dx.doi.org/10.1145/2460296.2460318>
- National Science Foundation (2014). *Computing Education for the 21st Century* [Informe técnico] Recuperado de <http://www.nsf.gov/pubs/2012/nsf12527/nsf12527.htm>
- National Science Foundation (2015). *CS Principles* [Curso en línea] Recuperado de <http://apcsprinciples.org/>

- Navas Ara, M. J. (2001). *Métodos, diseños y técnicas de investigación psicológica*. Madrid: UNED.
- Negroponete, N. (1995). *Ser digital*. Barcelona: Ediciones B.
- NFER & Nesta (2015). *Flipped Learning Research Report* [Informe de Investigación] Recuperado de https://www.nesta.org.uk/sites/default/files/flipped_learning_-_research_report.pdf
- Ng, W. (2012). Can we teach digital natives digital literacy? *Computers & Education*, 59(3), 1065-1078.
- Nikolova, O., & Taylor, G. (2003). The impact of a language learning task on instructional outcomes in two student populations: *High-ability and average-ability students*. *Journal of Secondary Gifted Education*, 14(4), 205-217.
- Noer, M. (11 de febrero de 2012). One man, one computer, 10 million students: How Khan Academy is reinventing education. *Forbes.com*. Recuperado de <http://www.prisim.com/wp-content/uploads/2013/12/One-Man-One-Computer-10-Million-Students-How-Khan-Academy-Is-Reinventing-Education-Forbes.pdf>
- O'Brien, B., Friedman-Nimz, R., Lacey, J., & Denson, D. (2005). From bits and bytes to C++ and web sites: What is computer talent made of? *Gifted Child Today*, 28(3), 56-64.
- O'Reilly, T. (23 de febrero de 2006). Qué es Web 2.0. Patrones del diseño y modelos del negocio para la siguiente generación del software. *Telos.com*. En http://elchinorarodemadera.com/wp-content/uploads/2013/03/Web2punto0_by_Tim_OReilly.pdf
- Olabe, J. C., Basogain, X., Olabe, M. A., Maíz, I., & Castaño, C. (2014). Solving math and science problems in the real world with a computational mind. *Journal of New Approaches in Educational Research (NAER Journal)*, 3(2), 75-82.
- Olive, J. (1991). LOGO programming and geometric understanding: An in-depth study. *Journal for Research in Mathematics Education*, 22(2), 90-111.
- Olszewski-Kubilius, P. (2001). Interview with Joyce VanTassel-Baska. *Journal of Secondary Gifted Education*, 12(2), 57-61.
- Olszewski-Kubilius, P., & Lee, S. (2004). Gifted adolescents' talent development through distance learning. *Journal for the Education of the Gifted*, 28(1), 7-35.
- Olszewski-Kubilius, P., & Lee, S. (2005). The role of participation in In-School and Outside-of-School activities in the talent development of gifted students. *Journal of Secondary Gifted Education*, 15(3), 107-123.
- Olszewski-Kubilius, P., Subotnik, R. F., & Worrell, F. C. (2015). Re-pensando las altas capacidades: Una aproximación evolutiva. *Revista de Educación*, 368, 40-65. doi: <http://dx.doi.org/10.4438/1988-592X-RE-2015-368-297>
- ORDEN 1493/2015 (2015). Orden 1493/2015, de 22 de mayo, de la Consejería de Educación, Juventud y Deporte, por la que se regula la evaluación y la promoción de los alumnos con

necesidad específica de apoyo educativo, que cursen segundo ciclo de Educación Infantil, Educación Primaria y Enseñanza Básica Obligatoria, así como la flexibilización de la duración de las enseñanzas de los alumnos con altas capacidades intelectuales en la Comunidad de Madrid. *Boletín Oficial de la Comunidad de Madrid*, 15/06/2015

- Palumbo, D. B. (1990). Programming language/problem-solving research: A review of relevant issues. *Review of educational research*, 60(1), 65-89.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Papert, S. (1994). *The children's machine: Rethinking school in the age of the computer*. New York: Basic Books.
- Papert, S., Watt, D., diSessa, A., & Weir, S. (1979). *Final report of the Brookline Logo Project: Project summary and data analysis (Logo Memo 53)*. Cambridge, MA: MIT Logo Group.
- Parke, B.N. (1989). *Gifted Students in Regular Classrooms*. Massachusetts: Allyn and Bacon.
- Pedersen, S., & Liu, M. (2002). Effects of modeling expert cognitive strategies during problem-based learning. *Journal of Educational Computing Research*, 26(4), 353-380. doi: <http://dx.doi.org/10.2190/6NL3-HMED-J8HE-GD4T>
- Pendarvis, E. D., Howley, A., & Howley, C.B. (1990). *The abilities of gifted children*. Nueva Jersey: Prentice-Hall.
- Pérez Juste, R. (2000). La evaluación de programas educativos: Conceptos básicos, planteamientos generales y problemática. *Revista de Investigación Educativa*, 18(2), 261-287.
- Pérez San-José, P. (2010). *Estudio sobre seguridad y privacidad en el uso de los servicios móviles por los menores españoles*. Madrid: INTECO-Orange. Recuperado de http://acercadeorange.orange.es/UpImages/files/2204/estudio_intecoorange_menores_y_moviles_31e23c99b05123c622f224dfa.pdf
- Piaget, J. (1952). *The origins of intelligence in children*. New York: International Universities Press.
- Plummer, J. D., Wasko, K. D., & Slagle, C. (2011). Children learning to explain daily celestial motion: Understanding astronomy across moving frames of reference. *International Journal of Science Education*, 33(14), 1963-1992.
- Pozniak, H. (21 de abril de 2014). Code to crack the job market. *TheTelegraph.co.uk*. Recuperado de <http://www.telegraph.co.uk/education/educationopinion/10767496/Code-to-crack-the-job-market.html>
- Prensky, M. (2001). Digital natives, digital immigrants. *On the Horizon*, 9(5), 1-6.
- Prensky, M. (13 de enero de 2008). *Programming Is the New Literacy* [Mensaje en un blog]. Recuperado de <http://www.edutopia.org/programming-the-new-literacy>

- Prensky, M. (2009). H. sapiens digital: From digital immigrants and digital natives to digital wisdom. *Innovate: Journal of Online Education*, 5(3).
- Prensky, M. (2011). *Enseñar a nativos digitales*. Madrid: Ediciones SM.
- Prince, K. R., Litovsky, A. R., & Friedman-Wheeler, D. (2012). Internet-mediated research: Beware of bots. *The Behavior Therapist*, 35(5), 85-88.
- PuroMarketing (s.f. 2014). En el mundo ya hay casi tantos teléfonos móviles como personas. *Puromarketing.com*. Recuperado de <http://www.puromarketing.com/12/19020/mundo-casi-tantos-telefonos-moviles-como-personas.html>
- Quartararo, J. (2002). Continental mathematics league. *Understanding our Gifted*, 14(4), 15-16.
- R. D. 943/2003 (2003). Real Decreto 943/2003, de 18 de julio, por el que se regulan las condiciones para flexibilizar la duración de los diversos niveles y etapas del sistema educativo para los alumnos superdotados intelectualmente. *Boletín Oficial del Estado*, 31/07/2003.
- RAE (2015). *Aplicación*. Recuperado el 14 de octubre de 2015 de <http://www.rae.es/>
- Rasch, G. (1993). *Probabilistic models for some intelligence and attainment tests*. Chicago: Chicago Press.
- Ratcliff, C. C., & Anderson, S. E. (2011). Reviving the turtle: Exploring the use of Logo with students with mild disabilities. *Computers in the Schools*, 28(3), 241-255.
- Raymond, E. S. (2001). *Cómo convertirse en hacker* [Documento en línea] Recuperado de <http://biblioweb.sindominio.net/telematica/hacker-como.html>
- Raymond, E. S. (2003). *The Art of Unix Programming*. Reading, MA: Addison-Wesley
- Reid, P. T., & Roberts, S. K. (2006). Gaining options: A mathematics program for potentially talented at-risk adolescent girls. *Merrill-Palmer Quarterly*, 52(2), 288-304. doi: <http://dx.doi.org/10.1353/mpq.2006.0019>
- Reis, S. M., & Renzulli, J. S. (2003). Research related to the Schoolwide Enrichment Triad Model. *Gifted Education International*, 18(1), 15-40.
- Renzulli, J. S. (1977). *The Enrichment Triad Model: A guide for developing defensible program for the gifted and talented*. Mansfield: Creative Learning Press.
- Renzulli, J. S. (1978). What makes giftedness? Reexamining a definition. *Phi Delta Kappan*, 78(3), 180-184.
- Renzulli, J. S. (Ed.). (1988). *Technical report of research studies related to the enrichment triad/revolving door model (3rd ed.)*. Storrs, CT: University of Connecticut.
- Renzulli, J. S. (2005). The three-ring conception of giftedness: A developmental model for promoting creative productivity. En R. J. Sternberg, & J. Davidson (Eds.), *Conceptions of giftedness* (2nd ed., pp. 217-245). Boston, MA: Cambridge University Press.

- Renzulli, J.S., & Gaesser, A. H. (2015). Un sistema multicriterial para la identificación del alumnado de alto rendimiento y de alta capacidad creativo-productiva. *Revista de Educación*, 368, 96-131. doi: <http://dx.doi.org/10.4438/1988-592X-RE-2015-368-290>
- Renzulli, J. S., & Reis, S. M. (1991). The Schoolwide Enrichment Model: A Comprehensive Plan for the Development of Creative Productivity. En N. Colangelo & G. A. Davis, *Handbook of Gifted Education* (pp. 111-141). Massachusetts: Allyn and Bacon.
- Resig, J. (14 de agosto de 2012). *Redefining the Introduction to Computer Science*. [Mensaje en un blog]. Recuperado de <http://ejohn.org/blog/introducing-khan-cs/>
- Resnick, M. (2013a). *Let's Teach Kids to Code* [Archivo de vídeo]. Recuperado de http://www.ted.com/talks/mitch_resnick_let_s_teach_kids_to_code.html
- Resnick, M. (08 de mayo de 2013b). *Learn to Code, Code to Learn* [Mensaje en un blog] Recuperado de <https://www.edsurge.com/news/2013-05-08-learn-to-code-code-to-learn>
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., et al. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11), 60-67. doi: <http://dx.doi.org/10.1145/1592761.1592779>
- Ritchel, M. (10 de mayo de 2014). Reading, Writing, Arithmetic, and Lately, Coding. *NYtimes.com*. Recuperado de <http://www.nytimes.com/2014/05/11/us/reading-writing-arithmetic-and-lately-coding.html>
- Robertson, J. (2012). Making games in the classroom: Benefits and gender concerns. *Computers & Education*, 59(2), 385-398. doi: <http://dx.doi.org/10.1016/j.compedu.2011.12.020>
- Robinson, M. A., Gilley, W. F., & Uhlig, G. E. (1988). The effects of guided discovery Logo on SAT performance of first grade students. *Education*, 109(2), 226-230.
- Rodríguez, S., Cabanach, R. G., Valle, A., Núñez, J. C., & González-Pineda, J. A. (2004). Diferencias en el uso de self-handicapping y pesimismo defensivo y sus relaciones con las metas de logro, la autoestima y las estrategias de autorregulación. *Psicothema*, 16(4), 625-631.
- Román-González, M. (2013a). Programación de aplicaciones para dispositivos móviles ('apps') como enriquecimiento curricular en sujetos de alta capacidad. Una revisión sistemática. *Trabajo Fin de Máster (TFM)*. Madrid: UNED. Recuperado de <https://db.tt/nx0XNmQ5>
- Román-González, M. (2013b). Programación de aplicaciones para dispositivos móviles ('apps') como enriquecimiento curricular en sujetos de alta capacidad. *Actas del XVI Congreso Nacional / II Internacional, Modelos de Investigación Educativa de AIDIPE* (Alicante, 4-6 de septiembre de 2013). Recuperado de <https://db.tt/hpY0bDFd>
- Román-González, M. (26 de noviembre de 2013c). 'Flipped Classroom': una oportunidad para profundizar en el EEES [Mensaje en un blog] Recuperado de

<http://www.theflippedclassroom.es/flipped-classroom-una-oportunidad-para-profundizar-en-el-eees/>

- Román-González, M. (2013d). *CSEW: Cuestionario para profesores/as (previo)* [Cuestionario en línea] Recuperado de <http://goo.gl/xCRO26>
- Román-González, M. (2013e). *CSEW: Cuestionario para profesores/as (post)* [Cuestionario en línea] Recuperado de <https://goo.gl/RRV9E5>
- Román-González, M. (2013f). *CSEW: Cuestionario para estudiantes* [Cuestionario en línea] Recuperado de <https://goo.gl/c3j5xq>
- Román-González, M. (2014a). Aprender a programar ‘apps’ como enriquecimiento curricular en alumnado de alta capacidad. *Bordón. Revista de Pedagogía*, 66(4), 135-155. doi: <http://dx.doi.org/10.13042/Bordon.2014.66401>
- Román-González, M. (2014b). Aprender a programar aplicaciones para dispositivos móviles (‘apps’) como enriquecimiento curricular en sujetos de alta capacidad: una revisión sistemática. *VI Congreso Iberoamericano de Pedagogía* (Chile, 23-25 septiembre de 2014).
- Román-González, M. (2014c). Codigoalfabetización y aprendizaje móvil. *I Simposio Internacional Mobile Learning*. Córdoba (21-23 de marzo de 2014). Recuperado de <https://db.tt/fgSMz2vL>
- Román-González, M. (2014d). *Test de Pensamiento Computacional – Versión 1.0* [Documento en línea]. Recuperado de <https://db.tt/66FJPURK>
- Román-González, M. (2014e). *Validación de expertos – 1ª Parte* [Cuestionario en línea]. Recuperado de <http://goo.gl/6p1gcR>
- Román-González, M. (2014f). *Validación de expertos – 2ª Parte* [Cuestionario en línea]. Recuperado de <http://goo.gl/JSrHSD>
- Román-González, M. (2014g). *Validación de expertos – 3ª Parte* [Cuestionario en línea]. Recuperado de <http://goo.gl/xCZIU8>
- Román-González, M. (2014h). *Test de Pensamiento Computacional – Versión 2.0* [Documento en línea]. Recuperado de <https://db.tt/6hg2seLu>
- Román-González, M. (2014i). *Test de Pensamiento Computacional* [Test en línea] Recuperado de <http://goo.gl/IYEKMB>
- Román-González, M. (2015a). Computational Thinking Test: Design Guidelines and Content Validation. *7th annual International Conference on Education and New Learning Technologies* (Barcelona, 6-8 de julio de 2015) Recuperado de <http://library.iated.org/view/ROMANGONZALEZ2015COM>
- Román-González, M. (2015b). Test de Pensamiento Computacional: principios de diseño, validación de contenido y análisis de ítems. En M. A. Murga Menoyo, & M. P. Trillo Miravalles

- (Coord.), *Perspectivas y avances de la investigación* (pp. 279-302). Madrid: UNED. Recuperado de <http://e-uned.es/product/product.php?prdctID=561>
- Román-González, M. (2015c). *Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org* [Cuestionario en línea] Recuperado de <https://goo.gl/Ovswgg>
- Román-González, M. (2015d). *Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org* [Cuestionario en línea] Recuperado de <https://goo.gl/6X9rcv>
- Román-González, M., Pérez-González, J.C., & Jiménez-Fernández, C. (2015). Test de Pensamiento Computacional: diseño y psicometría general. *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)* (Madrid, 14-16 de octubre de 2015). Recuperado de http://www.dmami.upm.es/dmami/documentos/liti/Actas_CINAIC_2015.pdf
- Rotigel, J. V., & Fello, S. (2004). Mathematically gifted students: How can we meet their needs? *Gifted Child Today*, 27(4), 46-51.
- Royal Society. (2012). *Shut down or restart: The way forward for computing in UK schools* [Informe técnico] Recuperado de <http://royalsociety.org/education/policy/computing-in-schools/report/>
- Ruipérez-Valiente, J. A., Muñoz-Merino, P. J., Leony, D., & Kloos, C. D. (2015). ALAS-KA: A learning analytics extension for better understanding the learning process in the Khan Academy platform. *Computers in Human Behavior*, 47, 139-148. doi: <http://dx.doi.org/10.1016/j.chb.2014.07.002>
- Rushkoff, D. (2010). *Program or be programmed*. New York: OR Books.
- Rushkoff, D. (13 de noviembre de 2012). *Code Literacy: A 21st-Century Requirement* [Mensaje en un blog]. Recuperado de <http://www.edutopia.org/blog/code-literacy-21st-century-requirement-douglas-rushkoff>
- Sacristán, A. (2013). *Sociedad del Conocimiento, Tecnología y Educación*. Madrid: Morata.
- Sadin, E. (2015). *La vie algorithmique: Critique de la raison numérique*. París: Éditions L'Échappée.
- Salaman, W. (2008). Reflections on progress in musical education. *British Journal of Music Education*, 25(3), 237-243. doi: <http://dx.doi.org/10.1017/S0265051708008073>
- Sanjanaashree, P., Kumar, M. A., & Soman, K. (2014). Language learning for visual and auditory learners using scratch toolkit. *Proceedings of the Computer Communication and Informatics (ICCCI), 2014 International Conference on* (pp. 1-5). Coimbatore, India: IEEE. doi: <http://dx.doi.org/10.1109/ICCCI.2014.6921765>
- Santiago, R., Trbaldo, S., Kamijo, M., & Fernández, A. (2015). *Mobile learning: nuevas realidades en el aula*. Barcelona: Digital-Text.
- Seisdedos, N. (2002). *RP-30 Resolución de Problemas* [Manual técnico] Madrid: TEA Ediciones.

- Seiter, L., & Foreman, B. (2013). Modeling the learning progressions of computational thinking of primary grade students. *Proceedings of the Ninth Annual International ACM Conference on International Computing Education Research*, 59-66. doi: <http://dx.doi.org/10.1145/2493394.2493403>
- Severance, C. (2015). Khan Academy and computer science. *Computer*, 48(1), 14-15.
- Shah, S. (12 de septiembre de 2014). Hour of Code: Build The Next DevOps Generation. *Informationweek.com*. Recuperado de <http://www.informationweek.com/strategic-cio/it-strategy/hour-of-code-build-the-next-devops-generation/a/d-id/1317967>
- Shaklee, B., & Landrum, M. (2000). Traveling the information highway in search of evidence. *Understanding our Gifted*, 12(3), 14-16.
- Shell, D. F., Hazley, M. P., Soh, L., Dee Miller, L., Chiriacescu, V., & Ingraham, E. (2014). Improving learning of computational thinking using computational creativity exercises in a college CS1 computer science course for engineers. *Frontiers in Education Conference (FIE), 2014 IEEE*, 1-7.
- Shivers, C. (2012). Shaking up expectations: The OCLS shake it! App. *Computers in Libraries*, 32(2), 14-17.
- Siegle, D. (2003). Mentors on the net: Extending learning through telementoring. *Gifted Child Today*, 26(4), 51-54.
- Siegle, D. (2004). Identifying students with gifts and talents in technology. *Gifted Child Today*, 27(4), 30-33.
- Siegle, D. (2005). Technology: An introduction to using spreadsheets to increase the sophistication of student projects. *Gifted Child Today*, 28(4), 50-55.
- Siegle, D. (2009). Developing student programming and problem-solving skills with visual basic. *Gifted Child Today*, 32(4), 24-29.
- Siegle, D. (2011). Technology: Presentations in the cloud with a twist. *Gifted Child Today*, 34(4), 54-58.
- Siegle, D. (2013). iPads: Intuitive technology for 21st-century students. *Gifted Child Today*, 36(2), 146-150.
- Siegle, D., & Powell, T. (2004). Exploring teacher biases when nominating students for gifted programs. *Gifted Child Quarterly*, 48(1), 21-29.
- Siemens, G. (2005). Conectivismo: Una teoría de la enseñanza para la era digital. *International Journal of Instructional Technology and Distance Learning*, 2(10).
- Siemens, G. (2010). Conectivismo: Una teoría de aprendizaje para la era digital. En R. Aparici (Coord.), *Conectados en el Ciberespacio* (pp. 77-90). Madrid: UNED.

- Silió, E. (7 de marzo de 2013). Aprender a programar como se aprende a leer. *El País.com*. Recuperado de http://sociedad.elpais.com/sociedad/2013/03/07/actualidad/1362689630_904553.html
- Slavin, R. E. (2008). Perspectives on evidence-based research in education—What works? Issues in synthesizing educational program evaluations. *Educational Researcher*, 37(1), 5-14.
- Smith, D. C., Cypher, A., & Tesler, L. (2000). Novice programming comes of age. *Communications of the ACM*, 43(3), 75–81.
- Soe, L., & Yakura, E. K. (2008). What's wrong with the pipeline? Assumptions about gender and culture in IT work. *Women's Studies*, 37(3), 176-201.
- Somyürek, S., & Coskun, B. K. (2013). Digital competence: Is it an innate talent of the new generation or an ability that must be developed? *British Journal of Educational Technology*, 44(5), 163-166. doi: <http://dx.doi.org/10.1111/bjet.12044>
- SRI Education (2014). *Research on the Use of Khan Academy in Schools* [Informe técnico] Recuperado de https://www.sri.com/sites/default/files/publications/2014-03-07_implementation_briefing.pdf
- St. Cyr, S. (2004). Can distance learning meet the needs of gifted elementary math students? *Gifted Child Today*, 27(2), 42-51.
- Statz, J. A. (1974). The development of computer programming concepts and problem-solving abilities among ten-year-olds learning Logo. *Dissertation Abstracts International*, 34, 5418B-5419B.
- Stefik, A., & Siebert, S. (2013). An Empirical Investigation into Programming Language Syntax. *ACM Transactions on Computing Education (TOCE)*, 13(4), 1-40. doi: <http://dx.doi.org/10.1145/2534973>
- Sternberg, R. (1991). Giftedness according to the Triarchic Theory of Human Intelligence. En N. Colangelo y Davis, G. A., *Handbook of Gifted Education* (pp. 45-54). Massachusetts: Allyn and Bacon.
- Suarez, Th. (2011). *A 12-year-old app developer* [Archivo de Vídeo]. Recuperado de http://www.ted.com/talks/thomas_suarez_a_12_year_old_app_developer
- Suárez Riveiro, J. M., & Fernández Suárez, A. P. (2005). Escalas de evaluación de las estrategias motivacionales de los estudiantes. *Anales de psicología*, 21(1), 116-128.
- Sullivan, F., & Lin, X. (2012). The ideal science student: Exploring the relationship of students' perceptions to their problem solving activity in a robotics context. *Journal of Interactive Learning Research*, 23(3), 273-308.

- Sun, C., Lin, H., & Hong Ho, C. (2006). Sharing tips with strangers: Exploiting gift culture in computer gaming. *CyberPsychology & Behavior*, 9(5), 560-570. doi: <http://dx.doi.org/10.1089/cpb.2006.9.560>
- Suppes, P., Holland, P. W., Hu, Y., & Vu, M. (2013). Effectiveness of an individualized computer-driven online math K-5 course in eight California title I elementary schools. *Educational Assessment*, 18(3), 162-181. doi: <http://dx.doi.org/10.1080/10627197.2013.814516>
- Tannenbaum, A. J. (1986). Giftedness: a psychosocial approach. En R. Sternberg y J. E. Davidson (eds.), *Conceptions of giftedness* (pp. 21-52). Cambridge: Cambridge University Press.
- Tannenbaum, A. J. (1991). The social psychology of giftedness. En N. Colangelo y G. A. Davis, *Handbook of Gifted Education* (pp. 27-44). Massachusetts: Allyn and Bacon.
- TEA Ediciones (2007). *PMA – Aptitudes Mentales Primarias* [Manual técnico]. Madrid: TEA Ediciones.
- TEA Ediciones (2015a). *Batería de Aptitudes Mentales Primarias (PMA)* [Test en línea] Disponible en <http://www.e-teaediciones.com/>
- TEA Ediciones (2015b). *Resolución de Problemas (RP30)* [Test en línea] Disponible en <http://www.e-teaediciones.com/>
- TEA Ediciones (2015c). *Formas Idénticas – Revisadas (FI-R)* [Test en línea] Disponible en <http://www.e-teaediciones.com/>
- TEA Ediciones (2015d). *Cuestionario “Big Five” de Personalidad para Niños y Adolescentes (BFQ-NA)* [Test en línea] Disponible en <http://www.e-teaediciones.com/>
- TEDx Talks (12 de febrero de 2013). *Hackschooling makes me happy* [Archivo de vídeo] Recuperado de <https://www.youtube.com/watch?v=h11u3vtcpaY>
- Tew, A. E., & Guzdial, M. (2010). Developing a validated assessment of fundamental CS1 concepts. *Proceedings of the 41st ACM Technical Symposium on Computer Science Education*, 97-101.
- Tew, A. E., & Guzdial, M. (2011). The FCS1: A language independent assessment of CS1 knowledge. *Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, 111-116.
- THNKR (2013a). *14-Year-Old Prodigy Programmer Dreams In Code* [Archivo de vídeo] Recuperado de https://youtu.be/DBXZWB_dNsw
- THNKR (2013b). *Is Prodigy Programmer The Next Steve Jobs?* [Archivo de vídeo] Recuperado de <https://youtu.be/XWny9lZ9fM8>
- Thompson, C. (15 de julio de 2011). How Khan Academy is changing the rules of education. *Wired Digital Magazine*. Recuperado de http://resources.rosettastone.com/CDN/us/pdfs/K-12/Wired_KhanAcademy.pdf

- Thomson, D. L. (2010). Beyond classroom walls: Teachers' and students' perspectives on how online learning can meet the needs of gifted students. *Journal of Advanced Academics*, 21(4), 662-712.
- Thurstone, L. L. (1938a). *Primary mental abilities*. Chicago: University of Chicago Press.
- Thurstone, L. (1938b). *The perceptual factor*. *Psychometrika*, 3(1), 1-17.
- Torrance, E.P. (1986). Teaching creative and gifted learners. En M. C. Wittrock (ed.), *Handbook of Research on Teaching* (pp. 630-647). London: McMillan.
- Tourón, J. (2008). La enseñanza a distancia: posibilidades para la atención individualizada de los alumnos de alta capacidad en la escuela y la familia. *Revista Española de Pedagogía*, 240, 297-314.
- Tourón, J. (2010). El desarrollo del talento y la promoción de la excelencia: Exigencias de un sistema educativo mejor. *Bordón. Revista De Pedagogía*, 62(3), 133-149.
- Tourón, J. (08 de febrero de 2013a). *Khan Academy. ¿Una revolución a coste cero!* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2013/02/khanacademy-una-revolucion-coste-cero.html>
- Tourón, J. (15 de febrero de 2013b). *Khan Academy para profesores. ¿Quién dijo que no es posible diferenciar?* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2013/02/khanacademy-para-profesores-quien-dijo.html>
- Tourón, J. (16 de abril de 2013c). *Dispositivos móviles en la clase: ¿modernidad o eficiencia?* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2013/04/dispositivos-moviles-en-la-clase.html>
- Tourón, J. (22 de abril de 2013d). *Enseñando con tablets: ¿moda u oportunidad?* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2013/04/ensenando-con-tablets-moda-u-oportunidad.html>
- Tourón, J. (28 de junio de 2013e). *The Flipped Classroom: ¿no has 'flipado' aún? (1)* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2013/06/the-flipped-classroom-no-has-flipado.html>
- Tourón, J. (01 de julio de 2013f). *Qué es y qué no es la Flipped Classroom: aclarando conceptos (2)* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2013/07/que-es-y-que-no-es-la-flipped-classroom.html>
- Tourón, J. (14 de septiembre de 2015). *¿Qué dicen los profesores sobre el Flipped Classroom?* [Mensaje en un blog] Recuperado de <http://www.javiertouron.es/2015/09/que-dicen-los-profesores-sobre-el.html>

- Tourón, J., & Santiago, R. (2015). El modelo Flipped Learning y el desarrollo del talento en la escuela. *Revista de Educación*, 368, 196-231. doi: <http://dx.doi.org/10.4438/1988-592X-RE-2015-368-288>
- Tourón, J., Santiago, R., & Díez, A. (2014). *The Flipped Classroom: Cómo convertir la escuela en un espacio de aprendizaje*. Barcelona: Digital-Text.
- Tourón, J., Santiago, R., & Vázquez, H. (2015). *The Flipped Classroom: Experiencias y recursos para 'darle la vuelta' a la clase* [Página web] Recuperado de <http://www.theflippedclassroom.es/>
- Turkle, S. (1995). *Life on the screen: Identity in the age of the internet*. New York: Simon & Schuster
- Turkle, S. (2011). *Alone Together: Why We Expect More from Technology and Less from Each Other*. New York: Basic Books
- Turkle, S. (2015). *Reclaiming Conversation: The Power of Talk in a Digital Age*. New York: Penguin Press.
- UK Department of Education (2013). *National curriculum in England: computing programmes of study* [Documento en línea] Recuperado de <https://www.gov.uk/government/publications/national-curriculum-in-england-computing-programmes-of-study>
- Urrea, C., & Bender, W. (2012). Making learning visible. *Mind, Brain, and Education*, 6(4), 227-241.
- U-Tad (06 de junio de 2013). *Las Apps del futuro llegan a U-tad con las I Olimpiadas Android para jóvenes*. U-Tad.com. Recuperado de <https://www.u-tad.com/noticias/las-apps-del-futuro-llegan-a-u-tad-con-las-i-olimpiadas-android-para-jovenes/>
- Valleespín, I. (19 de febrero de 2014). Las 'apps' del insti. *ElPaís.com*. Recuperado de http://tecnologia.elpais.com/tecnologia/2014/02/14/actualidad/1392406482_481618.html
- Valverde, J., Fernández, M. R., & Garrido, M. C. (2015). El pensamiento computacional y las nuevas ecologías de aprendizaje. *RED, Revista de Educación a Distancia*, 46. Recuperado de http://www.um.es/ead/red/46/valverde_et_al.pdf
- VanTassel-Baska, J. (1989). Appropriate curriculum for gifted learners. *Educational Leadership*, 46(6), 13-15.
- VanTassel-Baska, J. (2015). La diferenciación en acción: el Modelo de Currículo Integrado. *Revista de Educación*, 368, 232-254. doi: <http://dx.doi.org/10.4438/1988-592X-RE-2015-368-296>
- Von Stumm, S., Hell, B., & Chamorro-Premuzic, T. (2011). The hungry mind: intellectual curiosity is the third pillar of academic performance. *Perspectives on Psychological Science*, 6(6), 574-588. doi: <http://dx.doi.org/10.1177/1745691611421204>

- Voyer, D., Voyer, S., & Bryden, M. P. (1995). Magnitude of sex differences in spatial abilities: A meta-analysis and consideration of critical variables. *Psychological Bulletin*, 117(2), 250.
- Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes*. Cambridge, MA: Harvard University Press.
- Vygotsky, L. (1986). *Thought and language*. Cambridge, MA: MIT Press.
- Walden, J., Doyle, M., Garns, R., & Hart, Z. (2013). An informatics perspective on computational thinking. *Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education*, 4-9.
- Wang, L., & Chen, M. (2010). The effects of game strategy and preference-matching on flow experience and programming performance in game-based learning. *Innovations in Education and Teaching International*, 47(1), 39-52.
- Wang, M. C., & Lindvall, C. M. (1984). Individual differences and school learning environments. *Review of Research in Education*, 11, 161-225.
- Ward, M. (03 de marzo de 2014). An hour to catch the coding bug. *BBC.com*. Recuperado de <http://www.bbc.com/news/technology-26415021>
- Watt, M. (1982). *What is Logo? Creative Computing*, 8(10), 112-129.
- Webb, J. T. (1993). Nurturing social emotional development of gifted children. En K. A. Heller, F. J. Mönks, & A. H. Passow (eds.), *International Handbook of Research and Development of Giftedness and Talent* (pp. 525-538). Oxford: Pergamon Press.
- Webb, J. (2011). *The iPad as a Tool For Education - a case study* [Informe técnico] Recuperado de <http://www.naace.co.uk/publications/longfieldipadresearch>
- Weber, C. L., & Cavanaugh, T. W. (2006). Promoting reading: Using eBooks with gifted and advanced readers. *Gifted Child Today*, 29(4), 56-63.
- Weintrop, D., & Wilensky, U. (2015a). Using commutative assessments to compare conceptual understanding in blocks-based and text-based programs. *Proceedings of the Eleventh Annual International Conference on International Computing Education Research, ICER15*, 101-110. doi: <http://dx.doi.org/10.1145/2787622.2787721>
- Weintrop, D., & Wilensky, U. (2015b). To block or not to block, that is the question: Students' perceptions of blocks-based programming. *Proceedings of the 14th International Conference on Interaction Design and Children*, 199-208. doi: <http://dx.doi.org/10.1145/2771839.2771860>
- Werner, L., Denner, J., Campe, S., & Kawamoto, D. C. (2012). The fairy performance assessment: Measuring computational thinking in middle school. *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, 215-220.

- Whitmore, J. (1988). Nuevos retos a los métodos de investigación habituales. En J. Freeman (Dir.), *Los niños superdotados. Aspectos psicológicos y pedagógicos* (pp. 115-138). Madrid: Santillana.
- Wilensky, U., & Reisman, K. (2006). Thinking like a wolf, a sheep, or a firefly: Learning biology through constructing and testing computational theories—an embodied modeling approach. *Cognition and instruction*, 24(2), 171-209.
- Williamson, B. (2015). Political computational thinking: Policy networks, digital governance and 'learning to code'. *Critical Policy Studies*, 1-20. doi: <http://dx.doi.org/10.1080/19460171.2015.1052003>
- Wilson, A., Hainey, T., & Connolly, T. (2012). Evaluation of computer games developed by primary school children to gauge understanding of programming concepts. *6th European Conference on Games-Based Learning (ECGBL)*, 4-5.
- Wilson, C. (2015a). Hour of code: Bringing research to scale. *ACM Inroads*, 6(2), 18-18. doi: <http://dx.doi.org/10.1145/2746406>
- Wilson, C. (2015b). Hour of code---a record year for computer science. *ACM Inroads*, 6(1), 22-22. doi: <http://dx.doi.org/10.1145/2723168>
- Wilson, Z. S., Iyengar, S. S., Pang, S., Warner, I. M., & Luces, C. A. (2012). Increasing access for economically disadvantaged students: The NSF/CSEM & S-STEM programs at Louisiana State University. *Journal of Science Education and Technology*, 21(5), 581-587.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.
- Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions. Series A, Mathematical, Physical, and Engineering Sciences*, 366(1881), 3717-3725. doi: <http://dx.doi.org/10.1098/rsta.2008.0118>
- Wolfe, J. M. (1968). *Aptitude Assessment Battery Programming*. Montreal: Walden Personnel Testing & Consulting Inc.
- Wolz, U., Hallberg, C., & Taylor, B. (2011). Scrape: A tool for visualizing the code of scratch programs. *Poster Presented at the 42nd ACM Technical Symposium on Computer Science Education*, Dallas, TX.
- Wong, J. (22 de agosto de 2012a). *What I did at Khan Academy*. [Mensaje en un blog]. Recuperado de <http://jamie-wong.com/2012/08/22/what-i-did-at-khan-academy/>
- Wong, J. (14 de agosto de 2012b). *Khan Academy Computer Science: Instant Gratification and Bragging Rights*. [Mensaje en un blog]. Recuperado de <http://jamie-wong.com/2012/08/14/khan-academy-computer-science/>

- Wurst, C., Smarkola, C., & Gaffney, M. A. (2008). Ubiquitous laptop usage in higher education: Effects on student achievement, student satisfaction, and constructivist measures in honors and traditional classrooms. *Computers & Education*, 51(4), 1766-1783.
- Yang, Y., Cho, Y., Mathew, S., & Worth, S. (2011). College student effort expenditure in online versus face-to-face courses: The role of gender, team learning orientation, and sense of classroom community. *Journal of Advanced Academics*, 22(4), 619-638. doi: <http://dx.doi.org/10.1177/1932202X11415003>
- Yáñez, A. (10 de diciembre de 2015). *El futuro de la programación es femenino plural* [Mensaje en un blog] Recuperado de <http://programamos.es/programacion-femenino/>
- Yarbro, J., Arfstrom, K. M., McKnight, K., & McKnight, P. (2014). *Extension of a Review of Flipped Learning* [Informe técnico] Recuperado de <http://flippedlearning.org/cms/lib07/VA01923112/Centricity/Domain/41/Extension%20of%20Flipped%20Learning%20Lit%20Review%20June%202014.pdf>
- Ysseldyke, J., Tardrew, S. P., Betts, J., Thill, T. L., & Hannigan, E. (2004). Use of an instructional management system to enhance math instruction of gifted and talented students. *Journal for the Education of the Gifted*, 27(4), 293-310.
- Zapata-Ros, M. (2015). Pensamiento computacional y alfabetización digital. *RED, Revista de Educación a Distancia*, 46. Recuperado de <http://www.um.es/ead/red/46/zapata.pdf>
- Zavala, L. A., Gallardo, S. C. H., & García-Ruiz, M. A. (2013). Designing interactive activities within Scratch 2.0 for improving abilities to identify numerical sequences. *Proceedings of the 12th International Conference on Interaction Design and Children* (pp. 423–426). New York, NY, USA: ACM. doi: <http://dx.doi.org/10.1145/2485760.2485831>
- Zeilinger, J. (22 de abril de 2015). Meet The Group of Young Female Coders Who Developed A Yelp-Like Safety App. *Mic.com*. Recuperado de <http://mic.com/articles/116204/how-a-group-of-young-female-coders-developed-a-yelp-like-safety-app>
- Zhuang, H., & Morgera, S. D. (2007). Development of an undergraduate course. Internet-based instrumentation and control. *Computers & Education*, 49(2), 330-344.
- Zucker, A. A. (2009). Assessment made easy: Students flourish in a one-to-one laptop program. *Learning & Leading with Technology*, 36(8), 18-21.

ÍNDICE DE FIGURAS Y TABLAS

Figuras ordenadas por capítulos y anexos	Pág.
Figura 0.1. Códigos QR que enlazan a los tableros sobre codigoalfabetización curados por el doctorando en Pinterest y Scoop.it.....	16
Figura 1.1. Delimitación inicial del problema de investigación.....	19
Figura 1.2. Consola de programación en lenguaje JavaScript & ProcessingJS e interfaz de la aplicación resultante.....	20
Figura 1.3. Comandos y parámetros para la programación en lenguaje JavaScript & ProcessingJS.....	20
Figura 1.4. Sucesivos derivados o versiones de un objeto original de programación.....	22
Figura 1.5. Entorno de trabajo de la plataforma ‘MIT – App Inventor’.....	26
Figura 1.6. Mapa de Conocimiento (‘Knowledge Map’) del área de Matemáticas en KA.....	29
Figura 1.7. Relación detallada del progreso en las habilidades de un estudiante en el área de Matemáticas en KA.....	30
Figura 1.8. Interfaz con el conjunto de recursos para desarrollar la habilidad ‘Basic Division’ en KA.....	30
Figura 1.9. Videotutorial de KA en YouTube, de apoyo al aprendizaje de la habilidad ‘Basic Division’.....	31
Figura 1.10. ‘Badges’ (insignias digitales) obtenidas en KA por un estudiante tras el logro de sucesivas habilidades.....	31
Figura 1.11. Ejemplo de la actividad del estudiante en la plataforma KA a lo largo de ‘los pasados 30 días’.....	32
Figura 1.12. Conjunto de habilidades y videotutoriales abordados por el estudiante en ‘los pasados 30 días’, con los respectivos niveles de dominio alcanzados en KA.....	32
Figura 1.13. Itinerario formativo del módulo ‘Intro to JS: Drawing & Animation’ en KA.....	35
Figura 1.14. Tutorial de ‘consola de programación interactiva’ denominado ‘Intro to coloring’ en KA.....	35
Figura 1.15. Ilustración del modelo ‘flipped classroom’ (1).....	39
Figura 1.16. Ilustración del modelo ‘flipped classroom’ (2).....	40
Figura 1.17. Tiempo en la ‘clase invertida’ según la taxonomía de Bloom.....	41
Figura 1.18. Noticia en prensa de implantación pionera de ‘clase invertida’ en España.....	43
Figura 1.19. Oferta de ‘apps’ en ‘App Store’.....	49
Figura 1.20. Santiago González, programador prodigio de ‘apps’ con 14 años.....	51
Figura 1.21. Web de descarga de ‘apps’ de Santiago González.....	51

Figura 1.22. Página Web de las ‘Olimpiadas Android’	53
Figura 1.23. Países de la UE que ya integran, en alguna medida, la programación informática (‘coding’) en sus currículos.....	55
Figura 1.24. Mapa conceptual sobre la adecuación, viabilidad y relevancia de la programación de ‘apps’ como enriquecimiento curricular en alumnado de alta capacidad.....	58
Figura 1.25. Proceso de fijación de los parámetros de búsqueda.....	61
Figura 1.26. Códigos QR de enlace a la hoja de tabulación de las referencias de la revisión sistemática; y al documento con la recopilación de <i>abstracts</i>	63
Figura 1.27. Revistas de publicación de los artículos revisados.....	65
Figura 1.28. Año de publicación de los artículos revisados.....	65
Figura 1.29. Nivel educativo sobre el que se centran los artículos.....	67
Figura 1.30. Porcentaje de artículos revisados en cuyo <i>abstract</i> se encuentran indicios para el contraste de las respectivas hipótesis directivas enunciadas.....	68

Figura 2.1. El programa ‘Hello World!’ escrito en código HTML, JavaScript, Python (versión 3) y Ruby.....	77
Figura 2.2. Estándares de ‘alfabetización web’ de la Fundación Mozilla.....	87
Figura 2.3. Alfabetización Digital, Alfabetización Web, y Codigoalfabetización.....	88
Figura 2.4. Entorno de programación Scratch.....	92
Figura 2.5. Algunas cifras del evento ‘La Hora del Código’	93
Figura 2.6. La ‘brecha laboral’ en el campo de la programación informática.....	94
Figura 2.7. Cita de Steve Jobs utilizada como <i>leitmotiv</i> en ‘La Hora del Código’	95
Figura 2.8. Figuras relevantes de Silicon Valley promoviendo el evento ‘La Hora del Código’.	95
Figura 2.9. El presidente Barack Obama anima a los estudiantes a programar sus teléfonos móviles.....	96
Figura 2.10. Captura de pantalla de 1 de los 20 retos de ‘coding’ que se proponen en el tutorial interactivo de ‘La Hora del Código’ diseñado por Code.org.....	97
Figura 2.11. Plantilla personalizable para la certificación de participación el ‘La Hora del Código’	97
Figura 2.12. Delimitación final del problema de investigación.....	98
Figura 2.13. El lenguaje-entorno de programación informática Logo.....	100
Figura 2.14. ‘Modelo Computacional Multiagente’ de depredación lobos-ovejas en NetLogo...	105
Figura 2.15. Conceptos de programación implícitos en un <i>script</i> de Scratch.....	106
Figura 2.16. Ejemplo de lenguaje de programación visual por flechas.....	108
Figura 2.17. Ejemplo de lenguaje de programación visual por bloques.....	108
Figura 2.18. Ejemplo de lenguaje de programación textual.....	109

Figura 2.19. Ejemplo de una de las muchas interfaces tipo ‘laberinto’ existentes para el aprendizaje del ‘coding’.....	109
Figura 2.20. Ejemplo de una de las muchas interfaces tipo ‘lienzo o pizarra’ existentes para el aprendizaje del ‘coding’.....	110
Figura 2.21. Ejemplo de ‘coding’ en un proyecto abierto de narración digital, a través de Alice	111
Figura 2.22. Ejemplo de ‘coding’ a través de la programación directa de robots Bee-Bots.....	111
Figura 2.23. Ejemplo de ‘coding’ desde dispositivos móviles para controlar robots y drones, a través de Tickle.....	112
Figura 2.24. Ejemplo de ‘coding’ desde un ordenador para controlar una placa Arduino, a través de S4A.....	112
Figura 2.25. Cejilla para guitarra impresa en 3D, cuyos códigos se comparten y remezclan en Thingiverse.....	112
Figura 2.26. Proyecto Scratch “ <i>Flotar o no flotar: Esa es la cuestión</i> ”.....	126
Figura 2.27. Proyecto Scratch “ <i>La aventura de Mark</i> ”.....	127
Figura 2.28. Proyecto Scratch “ <i>Concurs</i> ”.....	127
Figura 2.29. Composición interna, editable, del proyecto “ <i>Concurs</i> ”.....	128
Figura 2.30. <i>Script</i> que controla uno de los objetos del proyecto “ <i>Concurs</i> ”.....	128
Figura 2.31. Ranking de sueldos para un programador informático en función del lenguaje que domine.....	130
Figura 2.32. Captura de pantalla de la comunidad ‘on-line’ DIY.org.....	132
Figura 2.33. Proyectos y Mentoras de la plataforma ‘ <i>Made with Code</i> ’ de Google.....	138
Figura 2.34. Ejemplo de cartel promocional de ‘ <i>Girls who code</i> ’.....	139
Figura 2.35. Símil entre el mundo globalizado y el mundo digitalizado.....	142

Figura 3.1. Codigoalfabetización y Pensamiento Computacional: dos caras de una misma moneda.....	143
Figura 3.2. Las 4 fases-pasos cognitivos del pensamiento computacional.....	148
Figura 3.3. <i>Algoritmo</i> del ‘cambio porcentual’ escrito como <i>programa</i> informático a través de <i>código</i> Python.....	151
Figura 3.4. Marco-modelo de pensamiento computacional en el aula propuesto desde ‘CAS Barefoot’.....	154
Figura 3.5. Ejemplo de algoritmo: secuencia de instrucciones para preparar una tostada.....	155
Figura 3.6. La detección del patrón revela la regla ortográfica (ejemplo tomado de ‘CAS Barefoot’)......	156
Figura 3.7. Ejemplo de ‘ <i>pair programming</i> ’.....	158
Figura 3.8. Diagrama resumen del Modelo MIT-Harvard de pensamiento computacional.....	159
Figura 3.9. Una secuencia con instrucciones repetidas, expresada a través de un bucle.....	160
Figura 3.10 Ejemplo de 3 secuencias ejecutadas en paralelo en respuesta al mismo evento.....	160

Figura 3.11. Operadores disponibles en Scratch.....	161
Figura 3.12. Familias de estándares curriculares para las ‘Ciencias de la Computación’ de la CSTA.....	167
Figura 3.13. El mismo programa informático escrito en un lenguaje textual y en un lenguaje visual.....	183
Figura 3.14. Galería de diseños artísticos producidos por estudiantes a través del lenguaje Logo.....	187
Figura 3.15. Red de ordenación y clasificación.....	191
Figura 3.16. Código QR de enlace al curso ‘K-8 Intro to Computer Science’.....	192
Figura 3.17. Itinerario curricular-formativo del curso ‘K-8 Intro to Computer Science’.....	194
Figura 3.18. Nivel 15 de la Etapa 2 del curso ‘K-8 Intro to Computer Science’.....	195
Figura 3.19. Nivel 7 de la Etapa 7 del curso ‘K-8 Intro to Computer Science’.....	195
Figura 3.20. Panel de seguimiento (‘tracking’) del profesor, y galería de trofeos digitales (‘badges’) del estudiante en el curso ‘K-8 Intro to Computer Science’.....	197
Figura 3.21. Visualización resultante del análisis de un conjunto de proyectos Scratch a través de la herramienta ‘Scrape’.....	206
Figura 3.22. Ítem utilizado en el ‘Test for Computational Thinking’ (Gouws, Bradshaw & Wentworth, 2013b), perteneciente a la categoría de ‘Patrones y Algoritmos’.....	210
Figura 3.23. Aproximación kinestésica al aprendizaje (‘embodied learning’) del PC.....	212
Figura 3.24. ‘Computational Thinking Patterns Graph’ desarrollado por la Universidad de Colorado.....	213
Figura 3.25. Ejemplo de ítem del ‘Test for Measuring Basic Programming Abilities’.....	216
Figura 3.26. Ejemplo de ítem del ‘Commutative Assessment’.....	217
Figura 3.27. Tarea ‘Ice Cream Machine’ de la Edición 2013 del Concurso Internacional Bebras.....	220
Figura 3.28. Tarea ‘Visiting friends’ de la Edición 2013 del Concurso Internacional Bebras.....	221
Figura 3.29. Diferentes niveles de competencia en la dimensión ‘Pensamiento lógico’.....	224
Figura 3.30. Retroalimentación de Dr. Scratch en función del nivel de competencia demostrado.....	224
<hr/>	
Figura 5.1. Histograma con la distribución por edad de la muestra de estudiantes.....	245
Figura 5.2. Centros educativos de procedencia de la muestra de estudiantes.....	246
Figura 5.3. Código QR que enlaza al ‘CSEW: Cuestionario para profesores (previo)’.....	249
Figura 5.4. Código QR que enlaza al ‘CSEW: Cuestionario para profesores (post)’.....	249
Figura 5.5. Código QR que enlaza al ‘CSEW: Cuestionario para estudiantes’.....	250
Figura 5.6. Distribución de los centros invitados por área geográfica, diferenciada para ambas ediciones.....	252

Figura 5.7. Distribución de los centros invitados por Comunidad Autónoma, diferenciada para ambas ediciones.....	253
Figura 5.8. Distribución de los centros invitados por titularidad, diferenciado por cada edición HoC.....	254
Figura 5.9. Distribución de los centros invitados por tipología, diferenciado por cada edición HoC.....	254
Figura 5.10. Código QR al documento con el contenido completo de las respuestas al cuestionario <i>pre</i> para profesores.....	255
Figura 5.11. Captura de una de las pantallas del tutorial propuesto por Code.org, utilizando los personajes de ‘Angry Birds’.....	260
Figura 5.12. Captura de una de las pantallas del tutorial ‘Lightbot’ para la HoC.....	261
Figura 5.13. Ejemplo del ‘tracking’ que se puede realizar a los estudiantes, sobre los niveles que van intentando y completando a lo largo de ‘La Hora del Código’.....	266
Figura 5.14. Código QR al documento con el contenido completo de las respuestas al cuestionario <i>post</i> para profesores.....	267
Figura 5.15. ⁴²⁵ Evolución en la Pregunta 1 por sexo a lo largo de las distintas etapas educativas.....	273
Figura 5.16. Evolución en la Pregunta 2 por sexo a lo largo de las distintas etapas educativas...	276
Figura 5.17. Penetración de los lenguajes de programación entre los estudiantes.....	278
Figura 5.18. Índices de penetración de cada lenguaje sobre el total de la muestra, diferenciado por sexos.....	279
Figura 5.19. Factor de género sobre el índice de penetración $[(\%chicos) \div (\%chicas)]$, según lenguajes.....	280
Figura 5.20. Comparativa de medias en las distintas aptitudes-habilidades.....	281
Figura 5.21. Diferencias por etapa educativa para la Pregunta 4.....	282
Figura 5.22. Diferencias según experiencia previa en ‘coding’ para la Pregunta 4.....	283
Figura 5.23. Solución de 3 factores para la Pregunta 4.....	286
Figura 5.24. Solución de 4 factores para la Pregunta 4.....	288
Figura 5.25. Distribución de frecuencias para la Pregunta 5, en función de la respuesta a la Pregunta 6.....	290
Figura 5.26. Media en la Pregunta 5 según etapa educativa.....	292
Figura 5.27. Media en la Pregunta 6 según etapa educativa.....	292
Figura 5.28. Medias de las respuestas a la Pregunta 4, en función de la Pregunta 5.....	293
Figura 5.29. Medias de las respuestas a la Pregunta 4, en función de la Pregunta 6.....	294
Figura 5.30. Medias en la Pregunta 7, en función de sexo y etapa educativa.....	296
Figura 5.31. Medias en la Pregunta 7, en función de la experiencia previa en ‘coding’ (Pregunta 2) y sexo.....	298

⁴²⁵ Para orientar al lector, aquí comienzan las figuras relativas a los resultados del ‘CSEW: Cuestionario para estudiantes’ (Anexo C)

Figura 5.32. Medias en la Pregunta 7, en función de la autoeficacia en ‘coding’ (Pregunta 5)...	299
Figura 5.33. Medias en la Pregunta 7, en función de la autoeficacia en ‘coding’ (Pregunta 5) y sexo.....	300
Figura 5.34. Medias en la Pregunta 7, en función de haber conseguido completar la actividad HoC (Pregunta 6) y sexo.....	301
Figura 5.35. Distribución de frecuencias de las respuestas a la Pregunta 8 del cuestionario para estudiantes.....	302
Figura 5.36. Medias en la Pregunta 8 según la etapa educativa.....	304
Figura 5.37. Medias en la Pregunta 9 según la etapa educativa.....	305
Figura 5.38. Media en la Pregunta 8, en función de la percepción de autoeficacia (Pregunta 5).	306
Figura 5.39. Media en la Pregunta 9, en función de la percepción de autoeficacia (Pregunta 5).	306
Figura 5.40. Distribución de frecuencias según sexo en los clústeres.....	310
Figura 5.41. Distribución de frecuencias según sexo y etapa educativa en los clústeres.....	311
Figura 5.42. Distribución de frecuencias según titularidad de centro en los clústeres.....	311
Figura 5.43. Diagrama de dispersión entre puntuaciones observadas-predichas por el modelo para la variable dependiente.....	315
Figura 5.44. Puntuaciones en la función discriminante para los sujetos que contestaron originalmente ‘sí’ a la P9.....	319
Figura 5.45. Puntuaciones en la función discriminante para los sujetos que contestaron originalmente ‘no’ a la P9.....	319
Figura 5.46. Distribución de casos anómalos del análisis discriminante por sexo.....	320
Figura 5.47. Distribución de casos anómalos del análisis discriminante por etapa educativa.....	320
Figura 5.48. Distribución de casos anómalos del análisis discriminante por clúster de pertenencia.....	321
Figura 5.49. Porcentajes de penetración de las distintas plataformas-tutoriales ‘on-line’ sobre el total de la muestra.....	323
Figura 5.50. Tutorial HoC de Code.org.....	324
Figura 5.51. Tutorial HoC de Scratch.....	324
Figura 5.52. Distribución de ‘coders’ y ‘scratchers’ según sexo.....	325
Figura 5.53. Distribución de ‘coders’ y ‘scratchers’ según etapa educativa.....	325
Figura 5.54. Medias en la Pregunta 4 para ‘coders’ y ‘scratchers’.....	326
Figura 5.55. Medias en la Pregunta 4 para ‘coders’ y ‘scratchers’, diferenciado para Primaria y Secundaria.....	327
Figura 5.56. Medias en las preguntas 5 y 8 para ‘coders’ y ‘scratchers’.....	328
Figura 5.57. Medias en las preguntas 5 y 8 para ‘coders’ y ‘scratchers’, diferenciado para Primaria y Secundaria.....	328
Figura 5.58. Códigos QR a sendos PDF con el conjunto de respuestas emitidas en las preguntas 11 y 12.....	329

Figura 5.59. Frecuencias en la Pregunta 11 en función de la Pregunta 12.....	330
Figura 5.60. Gráfico radial con la frecuencia de aparición relativa a cada tipo de razones para aprender ‘coding’.....	333
Figura 5.61. Evolución de ‘La Hora del Código’ a lo largo de las ediciones 2013 y 2014.....	335
Figura 5.62. Aspectos más positivos y más negativos reportados por los profesores en el cuestionario posterior a ‘La Hora del Código’.....	337
Figura 5.63. Análisis combinado de la brecha de género.....	339
Figura 5.64. Diagrama de dispersión entre el rendimiento informado en Tecnología / Informática y la percepción de autoeficacia en ‘coding’, diferenciado por sexo.....	343
Figura 5.65. Diagrama de dispersión tridimensional: “alta autoeficacia * alta motivación * sí a proseguir aprendizaje del ‘coding’ on-line”.....	344
Figura 5.66. Evolución de la ‘cascada motivacional virtuosa’, en función del sexo.....	345
Figura 5.67. Evolución de la ‘cascada motivacional virtuosa’ por sexo y etapa educativa (I)....	346
Figura 5.68. Evolución de la ‘cascada motivacional virtuosa’ por sexo y etapa educativa (II)...	346

Figura 6.1. Un problema computacional, y su solución expresada en lenguaje visual ‘por flechas’, visual ‘por bloques’, y ‘textual’.....	354
Figura 6.2. Cuadro resumen de especificaciones de los 40 ítems del TPC (<i>versión 1.0</i>) en los 5 ejes de diseño.....	356
Figura 6.3. Se corresponde con el ítem 8 del TPC (<i>versión 1.0</i>).....	357
Figura 6.4. Se corresponde con el ítem 9 del TPC (<i>versión 1.0</i>).....	357
Figura 6.5. Se corresponde con el ítem 19 del TPC (<i>versión 1.0</i>).....	358
Figura 6.6. Se corresponde con el ítem 31 del TPC (<i>versión 1.0</i>).....	358
Figura 6.7. Códigos QR de enlace a los 3 cuestionarios en línea utilizados para el procedimiento de validación de contenido.....	360
Figura 6.8. Nivel de dificultad percibido por los expertos para cada ítem.....	368
Figura 6.9. Pertinencia para medir el PC percibido por los expertos para cada ítem.....	369
Figura 6.10. ‘Índice de aceptación’ por ítem.....	369
Figura 6.11. ‘Índice de aceptación’ por ítem (+ ítems que son eliminados tras el proceso de validación de contenido).....	370
Figura 6.12. Revisión del ítem 9 de la versión original del TPC (ver anterior Figura 6.4), para la versión final del TPC.....	371
Figura 6.13. Cuadro resumen de especificaciones de los 28 ítems del TPC (<i>versión 2.0</i>) en los 5 ejes de diseño.....	373
Figura 6.14. Código QR de enlace al formulario de prueba de virtualización del TPC.....	374
Figura 6.15 Código QR de enlace a la versión final del TPC.....	380
Figura 6.16. Ejemplos de ítems de la Batería PMA.....	381
Figura 6.17. Ítem del Test de Resolución de Problemas RP30.....	382

Figura 6.18. Ítem del Test de Formas Idénticas-Revisadas (FI-R).....	383
Figura 6.19. Ítem del Cuestionario “ <i>Big Five</i> ” de Personalidad para Niños y Adolescentes (BFQ-NA).....	384
Figura 6.20. Códigos QR de enlace a las versiones TPC-B1, TPC-B2, y TPC-B3.....	386
Figura 6.21. Histograma de las puntuaciones totales en el TPC para toda la muestra.....	388
Figura 6.22. Comparativa de las distribuciones de la puntuación total en el TPC por ciclo educativo.....	390
Figura 6.23. Diagrama de cajas de la puntuación total en el TPC por ciclo educativo.....	391
Figura 6.24. Media en la puntuación total del TPC por ciclo educativo.....	391
Figura 6.25. Diagrama de cajas de la puntuación total en el TPC por curso académico.....	392
Figura 6.26. Media en la puntuación total del TPC por curso académico.....	393
Figura 6.27. Diferencias por sexo en la puntuación total en el TPC, a lo largo de los ciclos educativos.....	394
Figura 6.28. Evolución de la ‘autoeficacia específica’ por ciclo educativo.....	396
Figura 6.29. Evolución de la ‘autoeficacia general’ por ciclo educativo.....	396
Figura 6.30. Índice de dificultad (sin corregir y corregido) para cada uno de los ítems del TPC	399
Figura 6.31. Correlación ítem-total para cada uno de los ítems del TPC.....	400
Figura 6.32. Índice de Discriminación (<i>D</i>) para cada uno de los ítems del TPC.....	401
Figura 6.33. Fiabilidad del TPC como estabilidad temporal.....	403
Figura 6.34. Diagrama de dispersión TPC * PMA-R.....	405
Figura 6.35. Diagrama de dispersión TPC * PMA-E.....	405
Figura 6.36. Diagrama de dispersión TPC * PMA-N.....	406
Figura 6.37. Diagrama de dispersión TPC * PMA-V.....	406
Figura 6.38. Gráficos de regresión parcial de PMA-R sobre TPC, y PMA-E sobre TPC.....	408
Figura 6.39. Diagrama de dispersión entre las puntuaciones totales observadas en el TPC y las predichas por el modelo PMA.....	408
Figura 6.40. Diagrama de dispersión TPC * RP30.....	409
Figura 6.41. Diagrama de dispersión TPC*Inestabilidad Emocional, y TPC*Amabilidad.....	410
Figura 6.42. Diagrama de dispersión TPC*Conciencia, y TPC*Extraversión.....	411
Figura 6.43. Diagrama de dispersión TPC*Apertura.....	411
Figura 6.44. Gráficos de regresión parcial de BFQ-NA (Ap) sobre TPC, y BFQ-NA (Ex) sobre TPC.....	413
Figura 6.45. Diagrama de dispersión entre las puntuaciones observadas TPC y las predichas por el modelo BFQ-NA.....	413
Figura 6.46. Diagrama de dispersión TPC*Nota en Matemáticas.....	414
Figura 6.47. Diagrama de dispersión TPC*Nota en Lengua.....	415
Figura 6.48. Diagrama de dispersión TPC*Nota en Informática.....	415

Figura 6.49. Diagrama de dispersión entre las puntuaciones observadas TPC y las predichas por el modelo RA.....	416
Figura 6.50. Diagrama de dispersión TPC* Niveles-Pantallas completados en Code.org.....	418
Figura 6.51. Diagrama de dispersión TPC* Líneas de código escritas en Code.org.....	418
Figura 6.52. Diagrama de dispersión TPC*Niveles completados en Code.org, según sistema metodológico.....	419
Figura 6.53. Curva COR de rendimiento diagnóstico del TPC sobre la ‘alta capacidad computacional’	421
Figura 6.54. Diagramas de dispersión TPC * Tareas Bebras.....	424
Figura 6.55. Diagrama de dispersión TPC _{pre} * Dr. Scratch.....	426
Figura 6.56. Diagrama de dispersión TPC _{post} * Dr. Scratch.....	426
Figura 6.57. Gráfico de sedimentación del análisis factorial exploratorio inicial.....	429
Figura 6.58. Diagrama de dispersión TPC * TPC-R.....	434
Figura 6.59. Propuesta de revisión futura para el ítem 23.....	440
Figura 6.60. Modelo de validez factorial del TPC: una estructura factorial mixta.....	454

Figura 7.1. Código QR de enlace al “Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org”	461
Figura 7.2. Código QR de enlace al “Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org”	461
Figura 7.3. Diagrama del diseño <i>cuasi-experimental</i> seguido para la evaluación del curso K-8.	462
Figura 7.4. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘razonamiento lógico’ (PMA-R).....	467
Figura 7.5. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘espacial’ (PMA-E).....	468
Figura 7.6. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘numérica’ (PMA-N).....	468
Figura 7.7. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘verbal’ (PMA-V).....	468
Figura 7.8. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘resolución de problemas’ (RP30).....	470
Figura 7.9. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘perceptivo-atencional’ (A-E).....	472
Figura 7.10. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘perceptivo-atencional’ (ICI).....	472
Figura 7.11. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre la aptitud ‘pensamiento computacional’ (TPC).....	473
Figura 7.12. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre el rendimiento académico en Informática.....	476

Figura 7.13. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre el rendimiento académico en Matemáticas.....	476
Figura 7.14. Resumen de los resultados del diseño <i>cuasi-experimental</i> sobre el rendimiento académico en Lengua.....	477
Figura 7.15. Distribución de frecuencias del número de ‘niveles-pantallas’ completados por los estudiantes del curso K-8.....	483
Figura 7.16. Código QR de enlace a las respuestas abiertas del “ <i>Cuestionario de satisfacción para estudiantes sobre el Curso K-8 de Code.org</i> ”.....	487
Figura 7.17. Diagrama radial de las razones esgrimidas por los estudiantes K-8 que defienden la importancia del ‘coding’.....	488
Figura 7.18. Curva COR de potencia diagnóstica del TPC (<i>postest</i>) para discriminar entre sujetos que hicieron el K-8 frente al TIC Tradicional.....	491
Figura 7.19. Diagrama de dispersión ‘Niveles completados’ * ‘Líneas de código escritas’ en Code.org.....	494
Figura 7.20. Diagrama del diseño pre-experimental seguido para la evaluación del K-5 (‘ <i>Course 2</i> ’).....	499
Figura 7.21. Comparación de las medias TPC <i>pretest-postest</i> , en función del curso-programa seguido.....	502
Figura 7.22. Gráfica detallada del progreso del Caso 1 en el módulo ‘ <i>Intro to JS: Drawing & Animation</i> ’ del curso ‘ <i>Computer Programming</i> ’.....	506
Figura 7.23. Gráfica detallada del progreso del Caso 2 en el módulo ‘ <i>Intro to JS: Drawing & Animation</i> ’ del curso ‘ <i>Computer Programming</i> ’.....	507
Figura 7.24. Programa escrito por el Caso 1 en el módulo ‘ <i>Intro to JS: Drawing & Animation</i> ’ del curso ‘ <i>Computer Programming</i> ’.....	507
Figura 7.25. Programa escrito por el Caso 2 en el módulo ‘ <i>Intro to JS: Drawing & Animation</i> ’ del curso ‘ <i>Computer Programming</i> ’.....	508
<hr/>	
Figura 8.1. Modelo integral de codigoalfabetización y desarrollo del pensamiento computacional en Educación Primaria y Secundaria.....	523
<hr/>	
Figura A.1. Captura de pantalla del ‘CSEW: Cuestionario para profesores (previo)’.....	591
<hr/>	
Figura B.1. Captura de pantalla del ‘CSEW: Cuestionario para profesores (post)’.....	593
<hr/>	
Figura C.1. Captura de pantalla del ‘CSEW: Cuestionario para estudiantes’.....	598
<hr/>	
Figura D.1. Área geográfica de procedencia de los centros invitados a la investigación.....	599
Figura D.2. Comunidad Autónoma de procedencia de los centros invitados a la investigación..	599
Figura D.3. Titularidad de los centros educativos invitados a la investigación.....	601

Figura D.4. Tipología de los centros educativos invitados a la investigación.....	601
Figura D.5. Porcentajes de respuesta a la Pregunta 1 del cuestionario para estudiantes.....	602
Figura D.6. Frecuencias de respuesta a la Pregunta 1 según sexo.....	602
Figura D.7. Frecuencias de respuesta a la Pregunta 1 según etapa educativa.....	603
Figura D.8. Frecuencias de respuesta a la Pregunta 1 según la tipología de centro.....	603
Figura D.9. Porcentajes de respuesta a la Pregunta 2 del cuestionario para estudiantes.....	604
Figura D.10. Frecuencias de respuesta a la Pregunta 2 por sexo.....	604
Figura D.11. Frecuencias de respuesta a la Pregunta 2 por etapa educativa.....	604
Figura D.12. Frecuencias en la Pregunta 2 según la tipología de centro.....	605
Figura D.13. Índices de penetración según lenguajes sobre el total de la muestra, y diferenciado por etapa educativa.....	605
Figura D.14. Distribución de frecuencias para la Pregunta 4 (‘pensamiento lógico’).....	606
Figura D.15. Distribución de frecuencias para la Pregunta 4 (‘creatividad’).....	607
Figura D.16. Distribución de frecuencias para la Pregunta 4 (‘resolución de problemas’).....	607
Figura D.17. Distribución de frecuencias para la Pregunta 4 (‘trabajo en equipo’).....	608
Figura D.18. Distribución de frecuencias para la Pregunta 4 (‘persistencia-perseverancia’).....	609
Figura D.19. Distribución de frecuencias para la Pregunta 4 (‘capacidad de atención’).....	609
Figura D.20. Distribución de frecuencias en la Pregunta 5 del cuestionario para estudiantes.....	611
Figura D.21. Distribución de frecuencias en la Pregunta 6 del cuestionario para estudiantes.....	611
Figura D.22. Distribución de frecuencias en la Pregunta 5, según el sexo.....	612
Figura D.23. Distribución de frecuencias en la Pregunta 6, según el sexo.....	612
Figura D.24. Distribución de frecuencias en la Pregunta 6 en función de la respuesta a la Pregunta 2.....	613
Figura D.25. Medias relativas a la Pregunta 7 del cuestionario para estudiantes.....	614
Figura D.26. Medias en la Pregunta 7, diferenciadas por sexo.....	615
Figura D.27. Medias en la Pregunta 7, en función de la experiencia previa en ‘coding’ (Pregunta 2).....	615
Figura D.28. Medias en la Pregunta 7, en función de haber conseguido completar la actividad HoC (Pregunta 6).....	616
Figura D.29. Distribución de frecuencias de las respuestas a la Pregunta 9 del cuestionario para estudiantes.....	617
Figura D.30. Diagrama de dispersión de las respuestas a las preguntas 8 y 9.....	617
Figura D.31. Histograma de residuos tipificados.....	621
Figura D.32. Gráfico P-P de los residuos tipificados.....	621
Figura D.33. Gráfica de regresión parcial P5 sobre P8.....	622
Figura D.34. Gráfica de regresión parcial P4.2 sobre P8.....	622

Figura D.35. Gráfica de regresión parcial P4.3 sobre P8.....	623
Figura D.36. Gráfica de regresión parcial ‘Etapa Educativa’ sobre P8.....	623
Figura D.37. Distribución de frecuencias de respuesta a la Pregunta 11.....	627
Figura D.38. Distribución de frecuencias de respuesta a la Pregunta 12.....	628

Figura L.1. Foto de la Etapa 1 (‘Intro a las Ciencias de la Computación’) del Curso K-8.....	719
Figura L.2. Foto de la Etapa 3 (‘Pensamiento Computacional’) del Curso K-8.....	719
Figura L.3. Foto de la Etapa 4 (‘Programación sobre hoja cuadriculada’) del Curso K-8.....	720
Figura L.4. Foto de la Etapa 6 (‘Algoritmos’) del Curso K-8.....	720
Figura L.5. Foto de la Etapa 8 (‘Funciones’) del Curso K-8.....	720

Tablas ordenadas por capítulos y anexos	Pág.
--	-------------

Tabla 1.1. Convergencia entre principios curriculares para alumnos de AACC y el aprendizaje de programación de ‘apps’.....	23
Tabla 1.2. Convergencia entre estilo de aprendizaje de alumnos de alta capacidad y las características de Khan Academy.....	34
Tabla 1.3. Convergencia entre estilo de aprendizaje de alumnos de alta capacidad y el modelo ‘flipped classroom’.....	47
Tabla 1.4. Convergencia entre los parámetros de valoración de los productos de aprendizaje de alumnos de alta capacidad, y las características propias de las ‘apps’.....	54
Tabla 1.5. Comparativa de resultados de búsqueda por bases de datos.....	62
Tabla 1.6. Naturaleza u orientación de los artículos revisados.....	64
Tabla 1.7. Revistas de publicación de los artículos revisados.....	66
Tabla 1.8. Contingencia de aparición de los artículos en las distintas bases de datos.....	67
Tabla 1.9. Nivel educativo sobre el que se centran los artículos.....	67
Tabla 1.10. Indicios de contraste, encontrados en los <i>abstract</i> de las 67 referencias revisadas...	69

Tabla 2.1. Listado de recursos de enseñanza-aprendizaje del ‘coding’ clasificados según nuestra propuesta taxonómica.....	114
Tabla 2.2. Relación de países europeos (más Israel) que incluyen el ‘coding’ en su currículum actual o inmediato.....	117
Tabla 2.3. Razones por las cuales incorporar el ‘coding’ en el currículum, por países.....	119
Tabla 2.4. Objetivos específicos de la asignatura “Computing” del Reino Unido en función del ciclo educativo.....	121
Tabla 2.5. Contenidos de la asignatura “Tecnología, Programación y Robótica” de la Comunidad de Madrid.....	123

Tabla 3.1 Modelo transversal de desarrollo del PC de Barr & Stephenson.....	165
Tabla 3.2. Estándares curriculares propuestos por la CSTA para el ‘Pensamiento Computacional’ y las ‘Prácticas de Computación y Programación’.....	169
Tabla 3.3. ‘CAS Computing Progressión Pathways’.....	174
Tabla 3.4. Relación de artículos que estudian explícitamente el desarrollo del pensamiento computacional a través de la programación en contextos escolares K-12.....	184
Tabla 3.5. Comparación de artículos según diseño de investigación y dimensiones del pensamiento computacional.....	188
Tabla 3.6. Resumen de las 9 etapas de ‘coding on-line’ del curso ‘K-8 Intro to Computer Science’.....	198
Tabla 3.7. Resumen de las 11 etapas ‘unplugged’ del curso ‘K-8 Intro to Computer Science’...	201
Tabla 3.8. Conexión entre las 3 aproximaciones de evaluación del PC y el ‘Computational Thinking Framework’.....	208
Tabla 3.9. Comparativa de resultados de búsqueda entre el input “computational thinking test” y otros afines.....	210
Tabla 3.10. Criterios para diseñar una adecuada tarea Bebras.....	219
Tabla 3.11. Criterios que aplica Dr. Scratch para medir el nivel de competencia en las distintas dimensiones del PC.....	223
<hr/>	
Tabla 5.1. Centros registrados, invitados y aceptantes por edición de ‘La Hora del Código’.....	235
Tabla 5.2. Muestra de centros invitados a la edición HoC-2013.....	236
Tabla 5.3. Muestra de centros invitados a la edición HoC-2014.....	237
Tabla 5.4. Muestra de centros educativos aceptantes en ambas ediciones HoC.....	241
Tabla 5.5. Relación de sujetos que contestaron al cuestionario <i>pre</i> para profesores.....	242
Tabla 5.6. Número de estudiantes registrados y finalmente productores de datos por edición...	243
Tabla 5.7. Distribución por sexo de la muestra de estudiantes.....	243
Tabla 5.8. Estadísticos descriptivos relativos a la edad de la muestra de estudiantes.....	244
Tabla 5.9. Distribución por edad de la muestra de estudiantes.....	244
Tabla 5.10. Distribución de la muestra de estudiantes según etapa educativa.....	245
Tabla 5.11. Centros educativos de procedencia de la muestra de estudiantes.....	246
Tabla 5.12. Distribución de la muestra de estudiantes por área geográfica.....	247
Tabla 5.13. Distribución de la muestra de estudiantes por Comunidad Autónoma.....	247
Tabla 5.14. Distribución de la muestra de estudiantes por titularidad de su centro.....	247
Tabla 5.15. Distribución de la muestra de estudiantes por tipología de su centro.....	248
Tabla 5.16. Área geográfica de procedencia de los centros invitados a la investigación.....	251
Tabla 5.17. Comunidad Autónoma de procedencia de los centros invitados a la investigación..	251
Tabla 5.18. Titularidad de los centros educativos invitados a la investigación.....	253

Tabla 5.19. Tipología de los centros educativos invitados a la investigación.....	254
Tabla 5.20. Resumen de significatividad de las variaciones en la distribución de centros invitados a la investigación, a lo largo de las ediciones HoC.....	255
Tabla 5.21. ⁴²⁶ Frecuencias y porcentajes de respuesta a la pregunta 1 del cuestionario para estudiantes.....	272
Tabla 5.22. Tabla de contingencia Pregunta 1 * Sexo.....	272
Tabla 5.23. Tabla de contingencia Pregunta 1 * Etapa educativa.....	273
Tabla 5.24. Evolución en la Pregunta 1 por sexo a lo largo de las distintas etapas educativas...	273
Tabla 5.25. Resumen de significatividad de las diferencias por sexo en la Pregunta 1, diferenciado por etapas educativas.....	274
Tabla 5.26. Tabla de contingencia Pregunta 1 * Tipología de Centro.....	274
Tabla 5.27. Frecuencias y porcentajes de respuesta a la pregunta 2 del cuestionario para estudiantes.....	274
Tabla 5.28. Tabla de contingencia Pregunta 2 * Sexo.....	275
Tabla 5.29. Tabla de contingencia Pregunta 2 * Etapa educativa.....	275
Tabla 5.30. Evolución en la Pregunta 2 por sexo a lo largo de las distintas etapas educativas...	276
Tabla 5.31. Resumen de significatividad de las diferencias por sexo en la Pregunta 2, diferenciado por etapas educativas.....	276
Tabla 5.32. Tabla de contingencia Pregunta 2 * Tipología de centro.....	277
Tabla 5.33. Penetración de los lenguajes de programación entre los estudiantes.....	277
Tabla 5.34. Índices de penetración según lenguajes sobre el total de la muestra, y diferenciado por sexos.....	279
Tabla 5.35. Índices de penetración según lenguajes sobre el total de la muestra, y diferenciado por etapa educativa.....	280
Tabla 5.36. Estadísticos descriptivos relativos a la Pregunta 4 del cuestionario para estudiantes	281
Tabla 5.37. Diferencias según experiencia previa en ‘coding’ para la Pregunta 4.....	283
Tabla 5.38. Matriz de correlaciones entre las 6 aptitudes-habilidades interrogadas en la Pregunta 4.....	284
Tabla 5.39. Comunalidades de las variables en la solución de 3 factores.....	284
Tabla 5.40. Varianza total explicada por el modelo de 3 factores.....	285
Tabla 5.41. Matriz factorial rotada para 3 factores.....	285
Tabla 5.42. Solución de 3 factores para la Pregunta 4.....	286
Tabla 5.43. Comunalidades de las variables en la solución de 4 factores.....	286
Tabla 5.44. Varianza total explicada por el modelo de 4 factores.....	287
Tabla 5.45. Matriz factorial rotada para 4 factores.....	287

⁴²⁶ Para orientar al lector, aquí comienzan las tablas relativas a los resultados del ‘CSEW: Cuestionario para estudiantes’ (Anexo C)

Tabla 5.46. Solución de 4 factores para la Pregunta 4.....	287
Tabla 5.47. Estadísticos descriptivos relativos a la Pregunta 5 del cuestionario para estudiantes	288
Tabla 5.48. Distribución de frecuencias en la Pregunta 5 del cuestionario para estudiantes.....	289
Tabla 5.49. Distribución de frecuencias en la Pregunta 6 del cuestionario para estudiantes.....	289
Tabla 5.50. Prueba de diferencia de medias en la Pregunta 5, en función de la respuesta a la Pregunta 6.....	290
Tabla 5.51. Diferencias por sexo en las preguntas 5 y 6.....	290
Tabla 5.52. Correlaciones de la Pregunta 2 con las preguntas 5 y 6.....	291
Tabla 5.53. Diferencias en las preguntas 5 y 6, en función de la experiencia previa en ‘coding’ (Pregunta 2).....	291
Tabla 5.54. MANOVA sobre la Pregunta 6, en función del sexo y la experiencia previa en ‘coding’.....	291
Tabla 5.55. Correlaciones de las preguntas 5 y 6, con la pregunta 4.....	293
Tabla 5.56. Diferencias en la Pregunta 4, en función de la Pregunta 6.....	294
Tabla 5.57. Estadísticos descriptivos de la Pregunta 7 del cuestionario para estudiantes.....	295
Tabla 5.58. Diferencias en la Pregunta 7, en función del sexo.....	295
Tabla 5.59. Diferencias en la Pregunta 7, en función de la etapa educativa.....	296
Tabla 5.60. Correlaciones entre los elementos de la Pregunta 7.....	297
Tabla 5.61. Correlaciones entre los elementos de la Pregunta 7, con las preguntas 2, 5 y 6.....	297
Tabla 5.62. Diferencias en la Pregunta 7, en función de la experiencia previa en ‘coding’ (Pregunta 2).....	297
Tabla 5.63. Correlaciones entre preguntas 2 y 7, diferenciado por sexos.....	298
Tabla 5.64. Correlación entre las preguntas 5 y 7, diferenciado por sexo.....	299
Tabla 5.65. Diferencias en la Pregunta 7, en función de haber completado la actividad HoC (Pregunta 6).....	300
Tabla 5.66. Correlación entre las preguntas 6 y 7, diferenciado por sexo.....	301
Tabla 5.67. Descriptivos relativos a la Pregunta 8 del cuestionario para estudiantes.....	302
Tabla 5.68. Distribución de frecuencias de las respuestas a la Pregunta 8 del cuestionario para estudiantes.....	302
Tabla 5.69. Distribución de frecuencias de las respuestas a la Pregunta 9 del cuestionario para estudiantes.....	303
Tabla 5.70. Diferencias para las preguntas 8 y 9, según el sexo.....	303
Tabla 5.71. Diferencias para las preguntas 8 y 9, según el sexo; diferenciado por etapas educativas.....	303
Tabla 5.72. Correlaciones de la Pregunta 5 con las preguntas 8 y 9.....	305
Tabla 5.73. Diferencias en las preguntas 8 y 9, en función de la Pregunta 6.....	307

Tabla 5.74. Matriz de correlaciones de las preguntas 1, 2, 5, 6, 8 y 9 del cuestionario para estudiantes.....	307
Tabla 5.75. Centros de clústeres finales (expresados como puntuaciones típicas o <i>Zscore</i>).....	308
Tabla 5.76. Número de sujetos asignados a cada clúster.....	309
Tabla 5.77. Tabla de contingencia sexo * clúster * etapa educativa.....	310
Tabla 5.78. Variables introducidas y eliminadas en los sucesivos modelos de regresión según el método ‘por pasos’.....	312
Tabla 5.79. Resumen del modelo de regresión múltiple.....	313
Tabla 5.80. ANOVA del Modelo 4 de regresión múltiple.....	313
Tabla 5.81. Coeficientes para el Modelo 4 de regresión múltiple.....	314
Tabla 5.82. Correlación entre las puntuaciones observadas y predichas por el modelo en la variable dependiente.....	314
Tabla 5.83. Variables que entran en la función discriminante en cada escalón.....	316
Tabla 5.84. Variables que componen la función discriminante en cada escalón.....	316
Tabla 5.85. Autovalores de la función discriminante.....	317
Tabla 5.86. Discriminación residual.....	317
Tabla 5.87. Coeficientes estandarizados de la función discriminante.....	317
Tabla 5.88. Coeficientes no estandarizados de la función discriminante.....	317
Tabla 5.89. Valores de la función discriminante en los centroides de grupo.....	317
Tabla 5.90. Estadísticas de clasificación de la función discriminante.....	318
Tabla 5.91. Porcentajes de penetración de las distintas plataformas-tutoriales ‘ <i>on-line</i> ’ de aprendizaje del ‘ <i>coding</i> ’.....	321
Tabla 5.92. Diferencias en la Pregunta 4, en función de ser ‘ <i>coder</i> ’ o ‘ <i>scratcher</i> ’.....	326
Tabla 5.93. Diferencias en las preguntas 5, 8 y 9; en función de ser ‘ <i>coder</i> ’ o ‘ <i>scratcher</i> ’.....	327
Tabla 5.94. Distribución de frecuencias para la Pregunta 11.....	329
Tabla 5.95. Distribución de frecuencias para la Pregunta 12.....	329
Tabla 5.96. Tabla de contingencia Pregunta 11 * Pregunta 12.....	330
Tabla 5.97. Evolución de la ‘cascada motivacional virtuosa’, en función del sexo.....	345

Tabla 6.1. Panel de expertos aceptantes para participar en el proceso de validación de contenido.....	359
Tabla 6.2. Perfil profesional del panel de expertos.....	359
Tabla 6.3. Resumen de la valoración de los expertos sobre el instrumento en su conjunto.....	362
Tabla 6.4. Resumen de la valoración de los expertos sobre los ejes de diseño del TPC.....	363
Tabla 6.5. Resumen de resultados y decisiones para los ítems 1-5.....	364
Tabla 6.6. Resumen de resultados y decisiones para los ítems 6-10.....	365

Tabla 6.7. Resumen de resultados y decisiones para los ítems 11-15.....	365
Tabla 6.8. Resumen de resultados y decisiones para los ítems 16-20.....	366
Tabla 6.9. Resumen de resultados y decisiones para los ítems 21-25.....	366
Tabla 6.10. Resumen de resultados y decisiones para los ítems 26-30.....	367
Tabla 6.11. Resumen de resultados y decisiones para los ítems 31-35.....	367
Tabla 6.12. Resumen de resultados y decisiones para los ítems 36-40.....	367
Tabla 6.13. Distribución por sexo de la muestra total de aplicación del TPC.....	376
Tabla 6.14. Distribución por ciclo educativo de la muestra total de aplicación del TPC.....	376
Tabla 6.15. Distribución por curso académico de la muestra total de aplicación del TPC.....	376
Tabla 6.16. Distribución del total de la muestra según el dispositivo de aplicación.....	377
Tabla 6.17. Centro Educativo de procedencia de los sujetos de la muestra.....	377
Tabla 6.18. Centros Educativos participantes en la validación del TPC, agrupados según el procedimiento de muestreo.....	379
Tabla 6.19. Especificaciones de las 9 Tareas Bebras seleccionadas para la validación convergente del TPC.....	385
Tabla 6.20. Descriptivos de la puntuación total en el TPC para toda la muestra.....	388
Tabla 6.21. Descriptivos de la puntuación total en el TPC por ciclo educativo.....	389
Tabla 6.22. Descriptivos de la puntuación total en el TPC por curso académico.....	392
Tabla 6.23. Significatividad de la prueba <i>post-hoc</i> de Tukey para cada diferencia de medias....	393
Tabla 6.24. Diferencias de rendimiento en el TPC por sexo.....	394
Tabla 6.25. Diferencias de rendimiento en el TPC por sexo, a lo largo de los ciclos educativos.	394
Tabla 6.26. Descriptivos relativos a las preguntas de autoevaluación.....	395
Tabla 6.27. Correlaciones entre autoeficacia específica, autoeficacia general y puntuación total TPC.....	395
Tabla 6.28. Correlaciones entre autoeficacia específica, autoeficacia general y puntuación total TPC, por ciclo educativo.....	397
Tabla 6.29. Diferencias en ‘autoeficacia específica’ y ‘autoeficacia general’ por sexo.....	397
Tabla 6.30. Índice de dificultad (sin corregir y corregido) para cada uno de los ítems del TPC..	398
Tabla 6.31. Correlación ítem-total para cada uno de los ítems del TPC.....	399
Tabla 6.32. Índice de Discriminación (<i>D</i>) para cada uno de los ítems del TPC.....	400
Tabla 6.33. Fiabilidad del TPC como consistencia interna.....	401
Tabla 6.34. Fiabilidad del TPC en caso de eliminar cada uno de los ítems.....	402
Tabla 6.35. Fiabilidad del TPC como consistencia interna, en función de ciclo educativo y curso académico.....	402
Tabla 6.36. Fiabilidad del TPC como consistencia interna, en función del dispositivo de aplicación.....	403
Tabla 6.37. Fiabilidad del TPC como estabilidad temporal.....	403

Tabla 6.38. Matriz de correlaciones del TPC con el PMA-R, PMA-E, PMA-N, y PMA-V.....	404
Tabla 6.39. Resumen del modelo de regresión sobre el TPC a partir de las puntuaciones en la batería PMA.....	407
Tabla 6.40. ANOVA del TPC según el modelo de regresión construido a partir de la batería PMA.....	407
Tabla 6.41. Coeficientes del modelo de regresión sobre el TPC a partir de la batería PMA.....	407
Tabla 6.42. Correlación del TPC con el RP30.....	409
Tabla 6.43. Correlación del TPC con el FI-R (índices A-E e ICI).....	409
Tabla 6.44. Matriz de correlaciones entre el TPC y las dimensiones de personalidad del BFQ-NA.....	410
Tabla 6.45. Resumen del modelo de regresión sobre el TPC a partir de las puntuaciones en el cuestionario BFQ-NA.....	412
Tabla 6.46. ANOVA del TPC según el modelo de regresión construido a partir del cuestionario BFQ-NA.....	412
Tabla 6.47. Coeficientes del modelo de regresión sobre el TPC a partir del cuestionario BFQ-NA.....	412
Tabla 6.48. Matriz de correlaciones entre el TPC (al comienzo del trimestre) y las calificaciones en Informática, Matemáticas y Lengua (al final del trimestre).....	414
Tabla 6.49. Resumen del modelo de regresión sobre el TPC (al comienzo del trimestre) a partir de las calificaciones en Informática, Matemáticas y Lengua (al final del trimestre).....	416
Tabla 6.50. ANOVA del TPC según el modelo de regresión construido a partir de las calificaciones académicas.....	416
Tabla 6.51. Coeficientes del modelo de regresión sobre el TPC a partir de las calificaciones académicas.....	416
Tabla 6.52. Correlación TPC*Nota en Informática, en función del currículum seguido.....	417
Tabla 6.53. Correlaciones entre las puntuaciones totales en el TPC y las estadísticas de desempeño en Code.org.....	417
Tabla 6.54. Correlación TPC * Niveles completados en Code.org, en función de la metodología de aula.....	419
Tabla 6.55. Prueba <i>t</i> de diferencia de medias en el TPC entre el grupo ‘normal’ y el de ‘alta capacidad computacional’.....	420
Tabla 6.56. Área bajo la Curva COR de rendimiento diagnóstico del TPC sobre la ‘alta capacidad computacional’.....	421
Tabla 6.57. Coordenadas de la curva COR y puntos de corte de mayor potencia discriminante.	422
Tabla 6.58. Correlaciones del TPC con las Tareas Bebras.....	423
Tabla 6.59. Estadísticas de muestras emparejadas TPC <i>pre</i> y <i>post</i>	425
Tabla 6.60. Matriz de correlaciones TPC <i>pre</i> , TPC <i>post</i> , y Dr. Scratch.....	425
Tabla 6.61. Correlaciones TPC * Dr. Scratch, en función de la metodología de programación..	427

Tabla 6.62. Correlaciones TPC * Dimensiones del Pensamiento Computacional medidas por Dr. Scratch.....	427
Tabla 6.63. Varianza explicada por la solución inicial de 7 factores.....	428
Tabla 6.64. Matriz de estructura resultante del análisis factorial exploratorio inicial.....	429
Tabla 6.65. Especificaciones de los ítems que componen el Factor 1.....	430
Tabla 6.66. Estadísticos y fiabilidad de la subescala ‘Secuencias y Bucles’.....	430
Tabla 6.67. Especificaciones de los ítems que componen el Factor 2.....	430
Tabla 6.68. Estadísticos y fiabilidad de la subescala ‘Funciones’.....	431
Tabla 6.69. Especificaciones de los ítems que componen el Factor 3.....	431
Tabla 6.70. Estadísticos y fiabilidad de la subescala ‘Condicionales’.....	431
Tabla 6.71. Matriz factorial resultante del análisis unifactorial.....	431
Tabla 6.72. Estadísticos y fiabilidad de la subescala ‘Común-Principal’.....	432
Tabla 6.73. Estadísticas y fiabilidad de la versión reducida TPC-R.....	432
Tabla 6.74. Resumen de escalas y subescalas definidas.....	433
Tabla 6.75. Matriz de correlaciones entre las escalas y subescalas definidas.....	433
Tabla 6.76. Matriz de correlaciones de las escalas y subescalas, con las medidas de validez criterial concurrente.....	434
Tabla 6.77. Correlaciones TPC*PMA y TPC*RP30, y contingencia con los factores <i>Gf</i> , <i>Gv</i> , y <i>Gsm</i> del modelo CHC.....	443
Tabla 6.78. Alineamiento de los conceptos computacionales medidos por el TPC, con las dimensiones del pensamiento computacional medidas por Dr. Scratch.....	447

Tabla 7.1. Distribución de la muestra de estudiantes K-8 por centro educativo.....	458
Tabla 7.2. Distribución de la muestra de estudiantes K-8 por sexo.....	459
Tabla 7.3. Distribución de la muestra de estudiantes K-8 por curso académico.....	459
Tabla 7.4. Distribución de la muestra K-8 en función de la condición ‘experimental’ o ‘control’ de los sujetos.....	459
Tabla 7.5. Profesores participantes en la evaluación K-8.....	459
Tabla 7.6. Composición y medidas de los grupos participantes en la investigación.....	463
Tabla 7.7. Prueba <i>t</i> para muestras independientes en el PMA-R.....	464
Tabla 7.8. Prueba <i>t</i> para muestras relacionadas en el PMA-R.....	464
Tabla 7.9. ANCOVA sobre el PMA-R.....	464
Tabla 7.10. Prueba <i>t</i> para muestras independientes en el PMA-E.....	465
Tabla 7.11. Prueba <i>t</i> para muestras relacionadas en el PMA-E.....	465
Tabla 7.12. ANCOVA sobre el PMA-E.....	465
Tabla 7.13. Prueba <i>t</i> para muestras independientes en el PMA-N.....	465

Tabla 7.14. Prueba <i>t</i> para muestras relacionadas en el PMA-N.....	466
Tabla 7.15. ANCOVA sobre el PMA-N.....	466
Tabla 7.16. Prueba <i>t</i> para muestras independientes en el PMA-V.....	466
Tabla 7.17. Prueba <i>t</i> para muestras relacionadas en el PMA-V.....	466
Tabla 7.18. ANCOVA sobre el PMA-V.....	467
Tabla 7.19. Prueba <i>t</i> para muestras independientes en el RP30.....	469
Tabla 7.20. Prueba <i>t</i> para muestras relacionadas en el RP30.....	469
Tabla 7.21. ANCOVA sobre el RP30.....	470
Tabla 7.22. Prueba <i>t</i> para muestras independientes en el Índice A-E del FI-R.....	470
Tabla 7.23. Prueba <i>t</i> para muestras relacionadas en el Índice A-E del FI-R.....	470
Tabla 7.24. ANCOVA sobre el Índice A-E del FI-R.....	471
Tabla 7.25. Prueba <i>t</i> para muestras independientes en el Índice ICI del FI-R.....	471
Tabla 7.26. Prueba <i>t</i> para muestras relacionadas en el Índice ICI del FI-R.....	471
Tabla 7.27. ANCOVA sobre el Índice ICI del FI-R.....	471
Tabla 7.28. Prueba <i>t</i> para muestras independientes en el TPC.....	472
Tabla 7.29. Prueba <i>t</i> para muestras relacionadas en el TPC.....	473
Tabla 7.30. ANCOVA sobre el TPC.....	473
Tabla 7.31. Prueba <i>t</i> para muestras independientes en RA-Inf.....	474
Tabla 7.32. Prueba <i>t</i> para muestras relacionadas en RA-Inf.....	474
Tabla 7.33. ANCOVA sobre el RA-Inf.....	474
Tabla 7.34. Prueba <i>t</i> para muestras independientes en RA-Mat.....	474
Tabla 7.35. Prueba <i>t</i> para muestras relacionadas en RA-Mat.....	475
Tabla 7.36. ANCOVA sobre el RA-Mat.....	475
Tabla 7.37. Prueba <i>t</i> para muestras independientes en RA-Len.....	475
Tabla 7.38. Prueba <i>t</i> para muestras relacionadas en RA-Len.....	475
Tabla 7.39. ANCOVA sobre el RA-Len.....	475
Tabla 7.40. Resumen de resultados del diseño <i>cuasi-experimental</i> sobre todas las variables aptitudinales y de rendimiento.....	478
Tabla 7.41. Respuestas de los profesores sobre el número de semanas lectivas dedicadas al curso K-8.....	479
Tabla 7.42. Respuestas de los profesores sobre el % de etapas del curso K-8 impartidas realmente.....	479
Tabla 7.43. Respuestas de los profesores sobre qué partes del curso K-8 (<i>'on-line'</i> , <i>'unplugged'</i> o ambas) se han impartido realmente.....	480
Tabla 7.44. Respuestas de los profesores sobre su percepción del grado de aprendizaje de sus estudiantes con el K-8.....	480

Tabla 7.45. Estadísticos relativos a la pregunta “Globalmente, ¿cuál crees que ha sido el grado de aprendizaje de tus estudiantes con el Curso K-8?”	480
Tabla 7.46. Respuestas de los profesores sobre su nivel de satisfacción global con el K-8.....	480
Tabla 7.47. Estadísticos relativos a la pregunta “Como profesor/a responsable de la implantación del Curso K-8, ¿cuál es tu nivel de satisfacción con el curso globalmente?”	481
Tabla 7.48. Respuestas de los profesores sobre su intención de volver a impartir el curso K-8..	481
Tabla 7.49. Resumen de las respuestas de los profesores a las preguntas abiertas del cuestionario sobre el curso K-8.....	481
Tabla 7.50. Estadísticos sobre los ‘niveles-pantallas’ completados por los estudiantes del curso K-8.....	483
Tabla 7.51. Respuestas de los estudiantes sobre su percepción de autoeficacia en ‘coding’ tras el curso K-8.....	484
Tabla 7.52. Estadísticos relativos a la pregunta sobre percepción de autoeficacia en ‘coding’ tras el curso K-8.....	484
Tabla 7.53. Respuestas de los estudiantes sobre su motivación por continuar aprendiendo ‘coding’ tras el K-8.....	485
Tabla 7.54. Estadísticos relativos a la pregunta sobre motivación por continuar aprendiendo ‘coding’ tras el K-8.....	485
Tabla 7.55. Respuestas de los estudiantes sobre su intención de continuar aprendiendo ‘coding online’ tras el K-8.....	485
Tabla 7.56. Respuestas de los estudiantes sobre su grado de aprendizaje con el curso K-8.....	485
Tabla 7.57. Estadísticos relativos a la pregunta sobre grado de aprendizaje percibido con el curso K-8.....	486
Tabla 7.58. Respuestas de los estudiantes sobre su valoración global del curso K-8.....	486
Tabla 7.59. Estadísticos relativos a la pregunta sobre la valoración global del curso K-8.....	486
Tabla 7.60. Pruebas <i>t</i> de diferencia de medias en las preguntas del cuestionario para estudiantes del K-8, según sexo.....	487
Tabla 7.61. Respuestas de los estudiantes acerca de la importancia del ‘coding’ para su formación como ciudadanos.....	487
Tabla 7.62. Área bajo la Curva COR de potencia diagnóstica del TPC (<i>postest</i>) para discriminar sujetos K-8 vs. TIC.....	491
Tabla 7.63. Estructura y contenido del itinerario curricular del K-5 (‘Course 2’).....	495
Tabla 7.64. Prueba <i>t</i> para muestras relacionadas en el TPC.....	499
Tabla 7.65. Prueba <i>t</i> para muestras relacionadas en el TPC, segmentada por curso académico..	499
Tabla 7.66. Estadísticas de ‘niveles-pantallas’ completados en Code.org por los sujetos K-5...	500
Tabla 7.67. Comparación de los tamaños del efecto <i>pre-post</i> del TPC, en función del curso-programa seguido.....	501
Tabla 7.68. Puntuaciones directas y percentiles de los dos sujetos-casos de alta capacidad computacional en TPC y RP30.....	505

Tabla 7.69. Estadísticas de seguimiento (<i>'tracking'</i>) de los dos sujetos-casos en el módulo <i>'Intro to JS: Drawing & Animation'</i> del curso <i>'Computer Programming'</i>	505
<hr/>	
Tabla D.1. Tabla de contingencia entre el área geográfica de procedencia de los centros invitados y la edición HoC.....	600
Tabla D.2. Tabla de contingencia entre la CCAA de procedencia de los centros invitados y la edición HoC.....	600
Tabla D.3. Tabla de contingencia entre la Titularidad de los centros invitados y la edición HoC.....	601
Tabla D.4. Tabla de contingencia entre la Tipología de los centros invitados y la edición HoC.	602
Tabla D.5. Distribución de frecuencias para la Pregunta 4 (<i>'pensamiento lógico'</i>).....	606
Tabla D.6. Distribución de frecuencias para la Pregunta 4 (<i>'creatividad'</i>).....	606
Tabla D.7. Distribución de frecuencias para la Pregunta 4 (<i>'resolución de problemas'</i>).....	607
Tabla D.8. Distribución de frecuencias para la Pregunta 4 (<i>'trabajo en equipo'</i>).....	608
Tabla D.9. Distribución de frecuencias para la Pregunta 4 (<i>'persistencia-perseverancia'</i>).....	608
Tabla D.10. Distribución de frecuencias para la Pregunta 4 (<i>'capacidad de atención'</i>).....	609
Tabla D.11. Diferencias por sexo para la Pregunta 4.....	610
Tabla D.12. Diferencias por etapa educativa para la Pregunta 4.....	610
Tabla D.13. Pruebas KMO y Bartlett para la matriz de correlaciones relativa a la Pregunta 4...	611
Tabla D.14. ANOVAs en función de la etapa educativa para las preguntas 5 y 6.....	613
Tabla D.15. Diferencias en la Pregunta 4 en función de la Pregunta 5.....	613
Tabla D.16. Diferencias en la Pregunta 7, en función de la percepción de autoeficacia (Pregunta 5).....	616
Tabla D.17. Diferencias para las preguntas 8 y 9, en función de la etapa educativa.....	618
Tabla D.18. Diferencias en las preguntas 8 y 9, en función de la percepción de autoeficacia (Pregunta 5).....	618
Tabla D.19. Estadísticos descriptivos de las variables incluidas en el análisis de clúster.....	618
Tabla D.20. Centros de clústeres iniciales (expresados como puntuaciones típicas o <i>Zscore</i>)....	618
Tabla D.21. Historial de iteraciones (convergencia conseguida en 10 iteraciones).....	619
Tabla D.22. ANOVA de diferencias entre los clústeres para todas las variables incluidas en el análisis.....	619
Tabla D.23. Tabla de contingencia sexo * clúster.....	619
Tabla D.24. Tabla de contingencia titularidad * clúster.....	619
Tabla D.25. Estadísticos descriptivos de las variables inicialmente incluidas en la regresión múltiple.....	620
Tabla D.26. Estadísticas de residuos del modelo de regresión múltiple.....	620

Tabla D.27. Estadísticas descriptivas de las variables independientes inicialmente incluidas en el análisis discriminante.....	624
Tabla D.28. Prueba de igualdad de medias entre los grupos de la variable dependiente.....	625
Tabla D.29. Probabilidades previas para los grupos.....	626
Tabla D.30. Coeficientes de función de clasificación para los grupos.....	626
Tabla D.31. Tabla de contingencia ¿Caso anómalo? * Sexo.....	626
Tabla D.32. Tabla de contingencia ¿Caso anómalo? * Etapa Educativa.....	626
Tabla D.33. Tabla de contingencia ¿Caso anómalo? * Clúster de pertenencia.....	626
Tabla D.34. Tabla de contingencia ‘ <i>Coders vs. Scratchers</i> ’ * sexo.....	627
Tabla D.35. Tabla de contingencia ‘ <i>Coders vs. Scratchers</i> ’ * etapa educativa.....	627

LISTA DE SÍMBOLOS, ABREVIATURAS Y SIGLAS

α	Alfa de Cronbach
A-E	Índice de Percepción y Atención, del Test Formas Idénticas-Revisadas (FI-R)
AACC	Alumnos de Altas Capacidades
AB	Abstract
A_{COR}	Área bajo la Curva COR
AERA	American Educational Research Association
Am	Dimensión de personalidad ‘Amabilidad’, del Cuestionario BFQ-NA
ANCOVA	Análisis de Covarianza
ANOVA	Análisis de Varianza
Ap	Dimensión de personalidad ‘Apertura’, del Cuestionario BFQ-NA
APA	American Psychological Association
‘app’	Aplicación para dispositivo móvil
As	Asimetría
β	Coefficiente Beta del modelo de regresión múltiple
β_t	Coefficiente Beta tipificado del modelo de regresión múltiple
BFQ-NA	Cuestionario “ <i>Big Five</i> ” de Personalidad para Niños y Adolescentes
C	Coefficiente de Contingencia
CAS	Computing at School
CCAA	Comunidades Autónomas
CCE	Computational Creativity Exercises
CEIP	Colegio de Educación Infantil y Primaria
CHC	Modelo de Inteligencia Cattell-Horn-Carroll
CI	Coefficiente Intelectual
Co	Dimensión de personalidad ‘Conciencia’, del Cuestionario BFQ-NA
COR	Característica Operativa del Receptor
CPP	Computing Practices and Programming
CS	Computer Science
CS1	Computer Science-1: denominación anglosajona del 1º Curso del Grado de Informática
CSE	Computer Science Education
CSEW	Computer Science Education Week
CSTA	Computer Science Teachers Association
CT	Computational Thinking

CTD	Center for Talent Development
CTP	Computational Thinking Patterns
<i>d</i>	<i>d</i> de Cohen
<i>D</i>	Índice de Discriminación
DIY	Do It Yourself
e.g.	<i>exempli gratia</i> (por ejemplo)
EEES	Espacio Europeo de Educación Superior
ESO	Educación Secundaria Obligatoria
<i>et al.</i>	<i>et alii</i> (y otros)
Ex	Dimensión de personalidad ‘Extraversión’, del Cuestionario BFQ-NA
<i>F</i>	<i>F</i> de Fisher-Snedecor
FC	Flipped Classroom
FCS1	Foundational CS1 Assessment Instrument
FECYT	Fundación Española para la Ciencia y la Tecnología
FI-R	Test de Formas Idénticas-Revisadas
G _c	Grupo de Control
G _e	Grupo Experimental
<i>Gf</i>	‘ <i>Fluid reasoning</i> ’ en el Modelo de Inteligencia Cattell-Horn-Carroll
GPS	Global Positioning System
<i>Gsm</i>	‘ <i>Short-term memory</i> ’ en el Modelo de Inteligencia Cattell-Horn-Carroll
<i>Gv</i>	‘ <i>Visual processing</i> ’ en el Modelo de Inteligencia Cattell-Horn-Carroll
HoC	Hour of Code (Hora del Código)
I+D+i	Investigación, Desarrollo e Innovación
ICI	Índice de Control de la Impulsividad, del Test de Formas Idénticas-Revisadas (FI-R)
ICT	Information and Communication Technology
Ie	Dimensión de personalidad ‘Inestabilidad Emocional’, del Cuestionario BFQ-NA
IES	Instituto de Educación Secundaria
IM	Inteligencias Múltiples
INTEF	Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado
IRT	Item-Response Theory
ISTE	International Society for Technology in Education
ITC	Item-Total Correlation
JS	JavaScript
KA	Khan Academy
KMO	Medida Kaiser-Meyer-Olkin

KS	Key Stage
MABCM	Multi-Agent-Based Computational Model
MANOVA	Análisis Multivariante de la Varianza
MIDI	Musical Instrument Digital Interface
MIT	Massachusetts Institute of Technology
MOOC	Massive Open Online Course
N	Tamaño de la muestra
N/A	Not Available (No disponible)
NCME	National Council on Measurement in Education
NEE	Necesidades Educativas Especiales
NFER	National Foundation for Educational Research
NSF	National Science Foundation
OCDE	Organización para la Cooperación y el Desarrollo Económicos
$p_{(x)}$	Probabilidad asociada al estadístico x
p.e.	Por ejemplo
P2P	Aprendizaje entre pares o ' <i>peer to peer</i> '
P_c	Percentil
PC	Pensamiento Computacional
PC's	Personal Computers
P_d	Puntuación directa
p_i	Índice de Dificultad del Ítem i (sin corregir)
$p_{i\text{-corregido}}$	Índice de Dificultad del Ítem i (corregido)
PDF	Portable Document Format
PISA	Programme for International Student Assessment
PMA	Batería de Aptitudes Mentales Primarias (<i>Primary Mental Abilities</i>)
PMA-E	Test de Factor Espacial de la Batería PMA
PMA-N	Test de Factor Numérico de la Batería PMA
PMA-R	Test de Factor Razonamiento de la Batería PMA
PMA-V	Test de Factor Verbal de la Batería PMA
P_t	Puntuación típica
QR	Quick Response
r	Coefficiente de Correlación de Pearson
ρ_{xx}	Fiabilidad como estabilidad temporal
r_{xx}	Fiabilidad como Consistencia Interna, por el procedimiento de las dos mitades
R^2	Coefficiente de Determinación

R^2_{ajustado}	Coefficiente de Determinación Ajustado
RA	Rendimiento Académico
RA-Inf	Rendimiento Académico en la asignatura de Informática
RA-Len	Rendimiento Académico en la asignatura de Lengua
RA-Mat	Rendimiento Académico en la asignatura de Matemáticas
ROC	Receiver Operating Characteristic
RP30	Test de Resolución de Problemas RP30
<i>rtf</i>	Rich Text Format
<i>s</i>	Desviación típica
S4A	Scratch for Arduino
<i>sic</i>	<i>sic erat scriptum</i> (así escrito en el original)
STEM	Science, Technology, Engineering and Mathematics
SU	Sector of Use
<i>t</i>	<i>t</i> de Student
TCT	Teoría Clásica de los Test
TIC	Tecnologías de la Información y la Comunicación
TIMMS	Trends in International Mathematics and Science Study
TPC	Test de Pensamiento Computacional
TPC-B1	Test de Pensamiento Computacional + 3 Tareas Bebras B1
TPC-B2	Test de Pensamiento Computacional + 3 Tareas Bebras B2
TPC-B3	Test de Pensamiento Computacional + 3 Tareas Bebras B3
TPC-R	Test de Pensamiento Computacional – Reducido a 20 ítems
TRI	Teoría de Respuesta al Ítem
TX	All Text
UK	United Kingdom
URL	Uniform Resource Locator
<i>Vs.</i>	<i>Versus</i> (contra)
' <i>web app</i> '	Aplicación ejecutada a través de un navegador web
\bar{X}	Media aritmética
χ^2	Chi-Cuadrado
Z_{k-s}	Z de Kolgomorov-Smirnov
Z_{score}	Puntuación típica o estandarizada

RESUMEN / ABSTRACT

Introducción: Vivimos inmersos en una sociedad digital repleta de objetos controlados por software. En este contexto, la capacidad para leer y escribir en el lenguaje de las máquinas emerge como un requisito esencial, un nuevo alfabetismo para la formación de ciudadanos y profesionales del siglo veintiuno. Así, se ha definido a la codigofabetización como el conjunto de procesos de enseñanza-aprendizaje de la lectoescritura con los lenguajes informáticos de programación (*'coding'*); y al pensamiento computacional como el proceso cognitivo de resolución de problemas subyacente. Nos planteamos entonces si es posible (*adecuado y viable*) y deseable (*relevante*) introducir la codigofabetización en el sistema educativo español, específicamente en las etapas de Educación Primaria y Secundaria.

Método: Para abordar dicha pregunta de investigación se acometen tres estudios empíricos concatenados. **A)** En primer lugar, se realiza un estudio descriptivo mediante encuestas acerca del evento 'La Hora del Código' en España a lo largo de sus dos primeras ediciones, 2013 y 2014; sobre una muestra total de 23 profesores y 1.662 estudiantes, procedentes de 29 centros educativos ubicados por todo el territorio nacional. **B)** En segundo lugar, se lleva a cabo un estudio instrumental de diseño, construcción y validación del 'Test de Pensamiento Computacional' (TPC), destinado fundamentalmente a población escolar española de primer ciclo de la ESO. Para la validación de su contenido, nos apoyamos en el juicio de 20 expertos; y para el estudio de su psicometría, el TPC es aplicado sobre una muestra total de 1.251 estudiantes de 24 centros educativos distintos. Durante la validación del TPC, se aplican otras medidas psicológicas estandarizadas como la batería PMA, los tests RP30 y FI-R, y el cuestionario de personalidad BFQ-NA; así como emergentes medidas alternativas del pensamiento computacional: Tareas Bebras y Dr. Scratch. **C)** En tercer lugar, se lleva a cabo una evaluación de programas de codigofabetización y desarrollo del pensamiento computacional: a) evaluación del curso '*K-8 Intro to Computer Science*' de Code.org mediante un diseño *cuasi-experimental* (N=526 sujetos de la ESO); b) evaluación del curso '*K-5 Computer Science Fundamentals (Course 2)*' de Code.org mediante un *diseño pre-experimental* (N=51 sujetos de tercer ciclo de Primaria); y c) estudio de casos múltiples sobre dos sujetos de alta capacidad computacional de 1º ESO en el contexto del curso '*Computer Programming*' de Khan Academy.

Resultados: **A)** El evento 'La Hora del Código' consigue globalmente su objetivo de introducir el *'coding'* en los centros educativos, desmitificando su dificultad y promoviendo que los estudiantes se interesen por profundizar en el aprendizaje de la programación informática. **B)** Se consigue diseñar un instrumento, el TPC, que presenta propiedades psicométricas adecuadas: fiabilidad (consistencia interna y estabilidad) y validez (criterial, discriminante, convergente y factorial). **C)** El curso K-8, y de manera plausible el curso K-5, mejora el pensamiento computacional de los estudiantes, no así las otras variables psicológicas estudiadas; y ambos cursos gozan de excelente viabilidad para ser implantados, respectivamente, en primer ciclo de la ESO y tercer ciclo de Primaria.

Discusión: Globalmente, se concluye la posibilidad (*adecuación y viabilidad*) y deseabilidad (*relevancia*) de introducir la codigofabetización en nuestro sistema educativo. Si bien: se trata de una adecuación restringida al emergente constructo del pensamiento computacional; existen riesgos de que la viabilidad se vea descompensada en términos de género, a favor del colectivo masculino; y asoma la amenaza de que la relevancia quede reducida a aspectos utilitarios y sociodeterministas.

Palabras clave: codigofabetización, pensamiento computacional, educación primaria, educación secundaria, psicometría, evaluación de programas, alta capacidad computacional, inteligencia digital.

Introduction: We live immersed in a digital society full of objects driven by software. In this context, the ability to read and write in the language of machines emerges as an essential requirement, a new literacy for educating citizens and professionals from the twenty-first century. Thus, code-literacy is defined as all teaching-learning processes in order to read-write with computer programming languages (*'coding'*); and computational thinking as the underlying problem-solving cognitive skill. We consider then if it is possible (*suitable* and *viable*) and desirable (*relevant*) to incorporate code-literacy in the Spanish educational system, specifically in Primary and Secondary Education.

Method: To address such research question, three concatenated empirical studies are undertaken. **A)** First, a descriptive study through surveys is carried out about the 'Hour of Code' event in Spain over its first two editions, 2013 and 2014; on a total sample of 23 teachers and 1,662 students from 29 schools located all over the country. **B)** Secondly, an instrumental study is tackled to design, construct, and validate the 'Test de Pensamiento Computacional' (TPC), mainly aimed at Spanish school population from 7th and 8th Grade. To validate its content, we rely on the judgment of 20 experts; and, in order to study its psychometric properties, the TPC is applied on a total sample of 1,251 students from 24 different schools. During the TPC validation process, other standardized psychological measures are applied, as PMA battery, RP30 and FI-R tests, and BFQ-NA personality questionnaire; as well as emerging alternative measures of computational thinking: Bebras Tasks and Dr. Scratch. **C)** Thirdly, evaluations of code-literacy and computational thinking development programs are carried out: a) an evaluation of the '*K-8 Intro to Computer Science*' course from Code.org using a *quasi-experimental* design (N = 526 secondary students); b) an evaluation of the '*K-5 Computer Science Fundamentals (Course 2)*' from Code.org through a *pre-experimental* design (N = 51 primary students); and c) multiple case study on two secondary students with computational high abilities in the context of the '*Computer Programming*' course from Khan Academy.

Results: **A)** The 'Hour of Code' event globally achieves its goal of introducing the '*coding*' in schools, demystifying its difficulty and promoting students to deepen the learning of computer programming. **B)** It is achieved to design an instrument, the TPC, with adequate psychometric properties: reliability (internal consistency and stability) and validity (criterial, discriminant, convergent and factorial). **C)** The K-8 course, and plausibly the K-5 course, improves students' computational thinking, not the other psychological variables studied; and both courses have excellent viability of implantation, respectively, in the 7th - 8th Grade and in the 5th - 6th Grade.

Discussion: Globally, the possibility (*suitability* and *viability*) and desirability (*relevance*) to incorporate code-literacy in the Spanish educational system is concluded. Though: suitability is restricted to the emerging construct of computational thinking; there are risks about an unbalanced viability in terms of gender, in favor of the male group; and we overlook the threat that the relevance is reduced to utilitarian and social-deterministic aspects.

Keywords: code-literacy, computational thinking, primary education, secondary education, psychometrics, program evaluation, computational high abilities, digital intelligence.

ANEXO A

CSEW: Cuestionario para profesores/as (previo)



COMPUTER SCIENCE EDUCATION WEEK (CSEW)

Cuestionario para profesores/as (PREVIO)

OBJETIVO GENERAL: documentar e investigar cómo el aprendizaje de la programación con lenguajes informáticos (*'Coding'*) se está introduciendo en nuestros centros educativos, dada su posible consideración como competencia digital clave para el futuro personal y profesional de los estudiantes. En este marco general, la próxima celebración de la *'Computer Science Education Week'* (CSEW), del 8 al 14 de diciembre de 2014 (<http://csedweek.org/>), es una clara oportunidad para tratar de hacer un retrato de la situación del *'Coding'* en nuestro sistema educativo.

(EN) **GENERAL AIM:** To document and investigate how the learning of computer programming languages (*'Coding'*) is being introduced into our schools, since it may be considered as a key digital competence for personal and professional future of students. Within this general framework, the next celebration of the *'Computer Science Education Week'* (CSEW) from 8 to 14 December 2014 (<http://csedweek.org/>) is a clear opportunity to try to do a portrait of the situation of the *'Coding'* in our educational system.

Instrucciones

Este cuestionario está dirigido a profesores/as que van a coordinar la realización de las actividades relativas a la *'Computer Science Education Week'* (CSEW) en su centro educativo. Por favor, conteste a las siguientes preguntas basándose en su opinión y experiencia, tratando de ejemplificar sus afirmaciones. (Para cualquier duda o sugerencia sobre este cuestionario, puede consultar con el Prof. Marcos Román: mroman@edu.uned.es)

(EN) This survey is aimed at teachers who will coordinate the implementation of activities related to the *'Computer Science Education Week'* (CSEW) in their school. Please answer the following questions based on your opinion and experience, trying to illustrate your claims. (For any questions or suggestions regarding this questionnaire, please consult with Prof. Marcos Román: mroman@edu.uned.es)

Nombre y apellidos:

(EN) Name:

[Volver al índice](#)

Centro Educativo:

(EN) Education Center:

Provincia:

(EN) Province:

Titularidad del Centro:

(EN) Ownership of the center:

- Público
- Concertado
- Privado

Etapas educativas impartidas en el Centro:

(EN) Educational Levels:

- Infantil (Kindergarden)
- Primaria (K1 - K6)
- Secundaria (K7 - K10)
- Bachillerato (K11 - K12)
- FP - Grado Medio (Job Training - Medium)
- FP - Grado Superior (Job Training - High)

Cargo/s que ocupa usted en el Centro. Indique, además de su docencia, si tiene alguna otra responsabilidad en su centro (p.ej. Coordinador TIC, Coordinador de Etapa, Jefe de Departamento, Dirección, etc...):

(EN) Your position at the Center. Please besides your teaching, state if you have any other responsibility for your school (e.g. ICT Coordinator, Stage Coordinator, Head of Department, Direction, etc ...):

Preguntas

(EN): Questions

1. ¿Qué aptitudes y habilidades considera usted que desarrolla el aprendizaje de la programación informática (*'coding'*)? *

(EN): What skills and abilities do you consider develops learning computer programming (*'coding'*)?

2. ¿Considera usted el aprendizaje del *'coding'* especialmente adecuado para algún tipo específico de estudiante? (p.ej. para los chicos frente a las chicas, para los alumnos de alta capacidad frente a los alumnos de capacidad media, para los estudiantes de ESO frente a los estudiantes de Primaria, etc...) ¿Por qué? *

(EN): Do you consider learning the ‘coding’ particularly suitable for a specific type of student? (e.g. for boys versus girls, for students of high ability versus average ability students, for secondary students versus elementary students, etc ...) Why?

3. ¿Espera usted algún tipo de correlación entre el rendimiento habitual del estudiante en algún/as área/s curricular/es y su nivel de motivación y ejecución en la actividad de la CSEW? (p.ej. posible correlación entre el rendimiento en matemáticas y su futuro desempeño en la CSEW)

(EN): Do you expect some correlation between the usual student performance on any curricular area/s and their level of motivation and execution in the CSEW activity? (e.g. possible correlation between mathematics performance and future performance in CSEW activity)

4. Existen distintas plataformas-tutoriales ‘on-line’ (p.e. ‘Scratch’) para el aprendizaje del ‘coding’ ¿Cuáles de ellas conoce? ¿Cuáles de ellas le parecen más apropiadas para cada edad? *

(EN): There are different platforms-tutorials ‘on-line’ (e.g. ‘Scratch’) for learning the ‘coding’. Which ones do you know? Which ones seem most appropriate for which age?

5. Para planificar las actividades de la CSEW, ¿ha comparado distintas plataformas-tutoriales ‘on-line’ tal y como se sugiere en <http://csedweek.org/learn>? ¿Por cuál/es se ha decidido? ¿Por qué? *

(EN): To plan activities CSEW, have you compared different ‘on-line’ platforms-tutorials as suggested in <http://csedweek.org/learn>? By what have you been decided? Why?

6. Entendemos que al incorporar a su centro a la CSEW (‘An Hour of Code’) considera viable que con una actividad de una hora se pueda introducir a los estudiantes en el mundo del ‘coding’, ¿es así? ¿cómo es esto posible? *

(EN): We understand that by incorporating your center to the CSEW (‘An Hour of Code’), you considered feasible for a one-hour activity can introduce students to the world of ‘coding’. Is that right? How is this possible?

7. Durante la preparación de las actividades de la CSEW, ¿qué apoyos está encontrando entre la comunidad educativa de su centro? ¿qué resistencias? *

(EN): During the preparation of CSEW activities are you finding support among the educational community of your school? Are you finding resistance?

8. ¿Considera ‘aprender a programar’ una competencia relevante para el futuro personal (como ciudadanos) de sus estudiantes? ¿Por qué? *

(EN): Do you think ‘learn to program’ is a relevant skill for the personal future (as citizens) of your students? Why?

9. ¿Considera ‘aprender a programar’ una competencia relevante para el futuro profesional (como trabajadores) de sus estudiantes? ¿Por qué? *


(EN): Do you think ‘learn to program’ is a relevant skill for the professional future (as workers) of your students? Why?

10. Por favor, describa la/s actividad/es que tiene previsto realizar durante la CSEW: ¿cuántos estudiantes? ¿De qué edades? ¿Con qué recursos? ¿Con qué metodología? ¿Habrá algún tipo de evaluación? Puede incluir en su descripción igualmente cualquier otro comentario (o enlaces) que desee hacer al respecto. *

(EN): Please describe the activity/s that you’re planning for the CSEW: how many students? What ages? What resources? By what method? Will there be some kind of assessment? You can also include in your description any other comments (or links) you want to do about it.

MUCHAS GRACIAS POR SU COLABORACIÓN

THANK YOU FOR YOUR COOPERATION



**COMPUTER SCIENCE
EDUCATION WEEK
(CSEW): Cuestionario para
profesores/as (PREVIO)**

OBJETIVO GENERAL: documentar e investigar cómo el aprendizaje de la programación con lenguajes informáticos ('Computer Science' o 'Coding') se está introduciendo en nuestros centros educativos, dada su posible consideración como competencia digital clave para el futuro personal y profesional de los estudiantes. En este marco general, la próxima celebración de la 'Computer Science Education Week' (CSEW), del 8 al 14 de diciembre de 2014 (<http://csedweek.org/>), es una clara oportunidad para tratar de hacer un retrato de la situación del 'Coding' en nuestro sistema educativo.

(EN) **GENERAL AIM:** To document and investigate how the learning of computer programming languages ('Computer Science' or 'Coding') is being introduced into our schools, since it may be considered as a key digital competence for personal and professional future of students. Within this general framework, the next celebration of the 'Computer Science Education Week' (CSEW) from 8 to 14 December 2014 (<http://csedweek.org/>), is a clear opportunity to try to do a portrait of the situation of the 'Coding' in our educational system.



Figura A.1. Captura de pantalla del 'CSEW: Cuestionario para profesores (previo)' [Ubicado en <https://goo.gl/xCRO26>]

ANEXO B

CSEW: Cuestionario para profesores/as (post)



COMPUTER SCIENCE EDUCATION WEEK (CSEW)

Cuestionario para profesores/as (POST)

Instrucciones

Este cuestionario está dirigido a profesores/as que hayan coordinado la realización de las actividades relativas a la ‘*Computer Science Education Week*’ (CSEW) - LA HORA DEL CÓDIGO en su centro educativo. Por favor, conteste a las siguientes preguntas basándose en su opinión y experiencia durante la CSEW, tratando de ejemplificar sus afirmaciones. (Para cualquier duda o sugerencia sobre este cuestionario, puede consultar con el Prof. Marcos Román: mroman@edu.uned.es)

(EN) This survey is aimed at teachers who have coordinated the implementation of activities related to the ‘*Computer Science Education Week*’ (CSEW) - THE HOUR OF CODE in their school. Please answer the following questions based on your opinion and experience during the CSEW, trying to illustrate your claims. (For any questions or suggestions regarding this questionnaire, please consult with Prof. Marcos Román: mroman@edu.uned.es)

Nombre y apellidos:

(EN) Name:

Centro Educativo:

(EN) Education Center:

Preguntas

(EN): Questions

1. ¿Ha podido desarrollar las actividades de la CSEW tal y como tenía planificado? ¿Sí, No? ¿Por qué? Describa las actividades CSEW que finalmente han podido realizar: n° de estudiantes, niveles educativos, tutoriales utilizados, metodología seguida, sistema de evaluación, etc...

(EN): Have you been able to develop CSEW activities as you had planned? Yes, No? Why? Describe the CSEW activities you have been finally able to make: number of students, educational levels, tutorials used, methodology used, the evaluation system, etc.

2. ¿Cuáles son los aspectos MÁS POSITIVOS que desea resaltar del desarrollo de la CSEW en su centro?

(EN): What are the MOST POSITIVE aspects you want to highlight of the CSEW development in your school?

3. ¿Cuáles son los aspectos MÁS NEGATIVOS que desea resaltar del desarrollo de la CSEW en su centro? ¿Qué propuestas de mejora se plantea para los mismos?

(EN): What are the MOST NEGATIVE aspects you want to highlight of the CSEW development in your school? What suggestions to improve them?

4. Relate alguna/s anécdota relevante o caso excepcional ocurrido a lo largo de la CSEW (p.e. algún estudiante que haya demostrado especial talento, algún estudiante que haya mostrado una conducta inusual, etc...)

(EN): Relate a relevant anecdote/s or exceptional case/s occurred along the CSEW (e.g. a student who has demonstrated special talent, a student who has shown unusual behavior, etc ...)

5. Desde la fundación Code.org (promotora de la CSEW) se propone continuar y profundizar con la introducción del ‘coding’ en las escuelas: <http://code.org/learn/beyond> ¿Tienen intención de continuar en los próximos meses con nuevas actividades curriculares relacionadas con el ‘coding’? ¿Cuáles?

(EN): From the foundation Code.org (promoter of CSEW) intends to continue and deepen with the introduction of ‘coding’ in schools: <http://code.org/learn/beyond> Do you intend to continue in the coming months with new curricular activities related to the ‘coding’? Which?

MUCHAS GRACIAS POR SU COLABORACIÓN
THANK YOU FOR YOUR COOPERATION

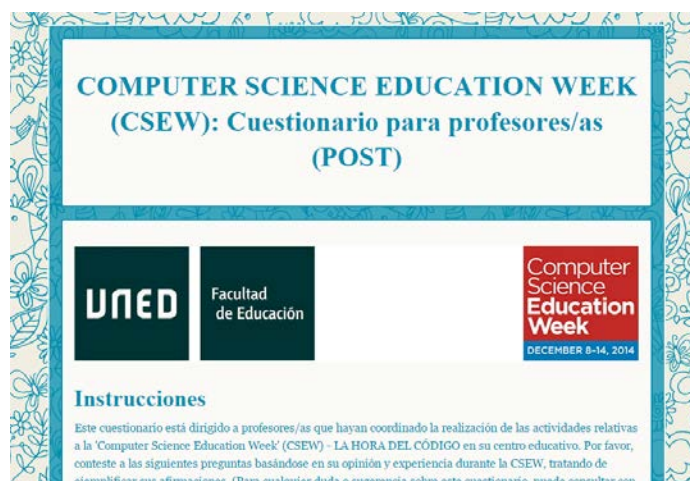


Figura B.1. Captura de pantalla del ‘CSEW: Cuestionario para profesores (post)’ [Ubicado en <https://goo.gl/RRV9E5>]

ANEXO C

CSEW: Cuestionario para estudiantes



COMPUTER SCIENCE EDUCATION WEEK (CSEW)

Cuestionario para Estudiantes

Instrucciones

Este cuestionario está dirigido a estudiantes que hayan participado en alguna de las actividades relativas a la ‘*Computer Science Education Week (CSEW): The Hour of Code*’ organizadas en su centro educativo. Por favor, contesta a las siguientes preguntas:

(EN): This survey is aimed at students who have participated in any activity relating to the ‘*Computer Science Education Week (CSEW): The Hour of Code*’ organized in their school. Please answer the following questions:

Datos Genéricos

(EN): Generic Data

¿Chica o Chico?

(EN): ¿Girl or Boy?

- Chica (Girl)
- Chico (Boy)

Edad

(EN): Age

- 8 ó menos
- 9
- 10
- (...)
- 19
- 20 ó más

Nivel Educativo

(EN): Educational Level

- Primaria
- Secundaria
- Bachillerato
- Formación Profesional

Centro Educativo

(EN): School

¿Es bilingüe tu Centro Educativo?

(EN): Is your school bilingual?

- Sí (Yes)
- No (No)

Preguntas

(EN): Questions

1. ¿Habías oído hablar de la ‘programación informática’ (*‘coding’*) antes de esta actividad de la ‘Computer Science Education Week (CSEW)’?

(EN): Have you ever heard of the ‘computer programming’ (*‘coding’*) before this activity of the ‘Computer Science Education Week (CSEW)’?

- Sí (Yes)
- No (No)

2. ¿Habías ‘programado’ alguna vez antes de esta actividad?

(EN): Had you ‘programmed’ (*‘coded’*) sometime before this activity?

- Sí (Yes)
- No (No)

3. En caso afirmativo a la pregunta anterior, ¿con qué lenguaje/s informático/s de programación tienes experiencia?

(EN): If yes to the previous question, what computer language/s you have programming experience?

- HTML
- CSS
- JavaScript
- jQuery
- PHP

- Python
- Ruby
- Otro:

4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?

(EN): How far do you think the activity has helped you to develop each of the following skills and abilities?

	1 = Nada (Nothing)	2 = Poco (Few)	3 = Algo (Something)	4 = Bastante (Quite)	5 = Mucho (A lot)
Pensamiento Lógico (Logical Thinking)					
Creatividad (Creativity)					
Resolución de Problemas (Problem Solving)					
Trabajo en Equipo (Teamwork)					
Persistencia-Perseverancia (Persistence-Perseverance)					
Capacidad de Atención (Attention Capacity)					

5. Tras tu experiencia con la actividad de la CSEW, ¿cómo dirías que se te da la ‘programación informática’ (‘coding’)?

(EN): After your experience with CSEW activity, how would you say you are gifted in ‘computer programming’ (‘coding’)?

- Se me da muy mal
- Se me da mal
- Se me da regular
- Se me da bien
- Se me da muy bien

6. ¿Has conseguido completar el 100% de la actividad propuesta?

(EN): Did you get 100% completion of the proposed activity?

- Sí (Yes)
- No (No)

7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?

(EN): What is your usual mark in the following subjects (or similars)?

	Suspensó (E)	Aprobado (D)	Notable Bajo (C)	Notable Alto (B)	Sobresaliente (A)
Matemáticas (Mathematics)					
Lengua (Language)					
Música (Music)					
Tecnología / Informática (Technology /Computing)					
Plástica (Plastics)					

8. Tras la actividad, ¿cómo de motivado te sientes para continuar aprendiendo ‘programación informática’ (‘coding’)?

(EN): After the activity, how do you feel motivated to continue learning ‘computer programming’ (‘coding’)?

- Nada motivado
- Poco motivado
- Algo motivado
- Bastante motivado
- Muy motivado

9. ¿Te gustaría continuar aprendiendo ‘programación informática’ (‘coding’) a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)?

(EN): Would you like to continue learning ‘computer programming’ (‘coding’) across platforms-tutorials on the Internet (at your own pace, interactively...)?

- Sí (Yes)
- No (No)

10. En caso afirmativo a la respuesta anterior, existen en Internet diversas plataformas-tutoriales para el aprendizaje de la ‘programación informática’ (‘coding’), ¿Cuál/es de ellas conoces?

(EN): If yes to the previous question, there are various platforms online tutorials for learning the ‘computer programming’ (‘coding’), which do you know?

- Code.org (<http://code.org/>)
- Scratch (<http://scratch.mit.edu/>)
- Blockly (<http://code.google.com/p/blockly/>)
- Tynker (<http://www.tynker.com/>)
- Light-Bot (<http://light-bot.com/>)
- App Inventor (<http://appinventor.mit.edu/explore/>)

- Khan Academy (<https://www.khanacademy.org/cs>)
- Codecademy (<http://www.codecademy.com>)
- CodeHS (<http://codehs.com/>)
- Otro:

11. ¿Consideras importante la ‘programación informática’ (‘coding’) para tu formación personal, como ciudadano del siglo XXI? ¿Sí, No? ¿Por qué?

(EN): Is it important to ‘computer programming’ (‘coding’) for your personal training, as a citizen of the XXI century? Yes, No? Why?

12. ¿Consideras importante la ‘programación informática’ (‘coding’) para tu formación profesional, como trabajador del siglo XXI? ¿Sí, No? ¿Por qué?

(EN): Is it important to ‘computer programming’ (‘coding’) for your professional training, as a worker of the XXI century? Yes, No? Why?

MUCHAS GRACIAS POR TU PARTICIPACIÓN

THANK YOU FOR YOUR COOPERATION



Figura C.1. Captura de pantalla del 'CSEW: Cuestionario para estudiantes' [Ubicado en <https://goo.gl/c3j5xq>]

ANEXO D

Figuras y Tablas complementarias al Capítulo 5

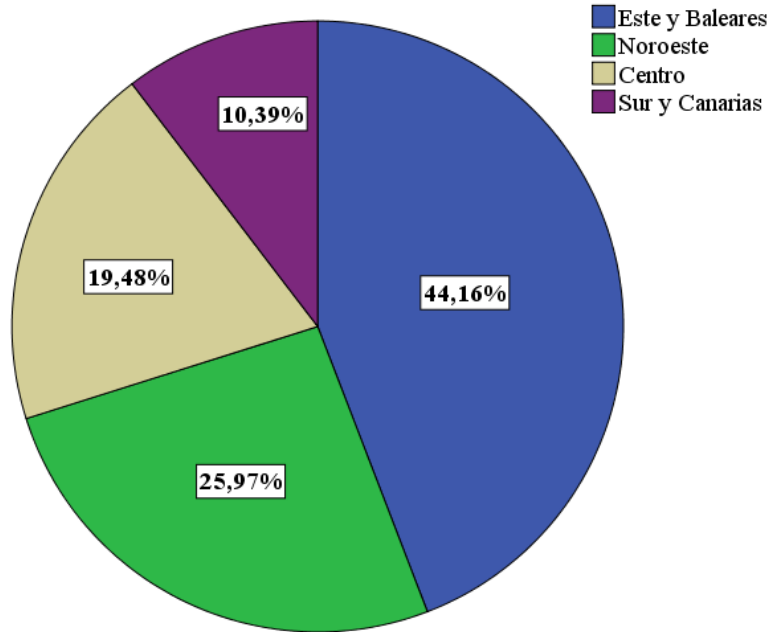


Figura D.1. Área geográfica de procedencia de los centros invitados a la investigación

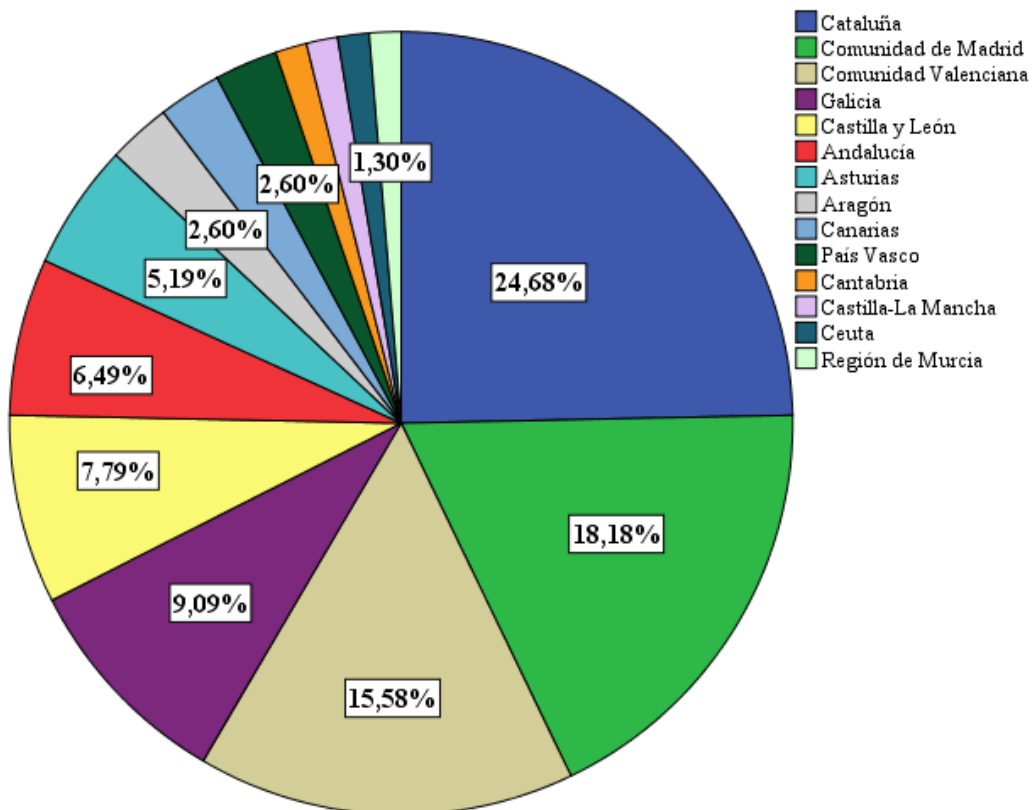


Figura D.2. Comunidad Autónoma de procedencia de los centros educativos invitados a la investigación

Tabla D.1. Tabla de contingencia entre el área geográfica de procedencia de los centros invitados y la edición HoC

		Edición		Total	
		HoC-2013	HoC-2014		
Área Geográfica	Centro	Recuento	4	11	15
		% dentro de Edición	17,4%	20,4%	19,5%
	Noroeste	Recuento	3	17	20
		% dentro de Edición	13,0%	31,5%	26,0%
	Este y Baleares	Recuento	13	21	34
		% dentro de Edición	56,5%	38,9%	44,2%
	Sur y Canarias	Recuento	3	5	8
		% dentro de Edición	13,0%	9,3%	10,4%
	Total	Recuento	23	54	77
		% dentro de Edición	100,0%	100,0%	100,0%

Tabla D.2. Tabla de contingencia entre la CCAA de procedencia de los centros invitados y la edición HoC

		Edición		Total	
		HoC-2013	HoC-2014		
Comunidad Autónoma	Andalucía	Recuento	2	3	5
		% dentro de Edición	8,7%	5,6%	6,5%
	Aragón	Recuento	0	2	2
		% dentro de Edición	0,0%	3,7%	2,6%
	Asturias	Recuento	1	3	4
		% dentro de Edición	4,3%	5,6%	5,2%
	Canarias	Recuento	1	1	2
		% dentro de Edición	4,3%	1,9%	2,6%
	Cantabria	Recuento	0	1	1
		% dentro de Edición	0,0%	1,9%	1,3%
	Castilla y León	Recuento	0	6	6
		% dentro de Edición	0,0%	11,1%	7,8%
	Castilla-La Mancha	Recuento	1	0	1
		% dentro de Edición	4,3%	0,0%	1,3%
	Cataluña	Recuento	7	12	19
		% dentro de Edición	30,4%	22,2%	24,7%
	Ceuta	Recuento	0	1	1
		% dentro de Edición	0,0%	1,9%	1,3%
	Comunidad de Madrid	Recuento	3	11	14
		% dentro de Edición	13,0%	20,4%	18,2%
	Comunidad Valenciana	Recuento	6	6	12
		% dentro de Edición	26,1%	11,1%	15,6%
	Galicia	Recuento	0	7	7
		% dentro de Edición	0,0%	13,0%	9,1%
	País Vasco	Recuento	2	0	2
		% dentro de Edición	8,7%	0,0%	2,6%
	Región de Murcia	Recuento	0	1	1
		% dentro de Edición	0,0%	1,9%	1,3%
Total	Recuento	23	54	77	
	% dentro de Edición	100,0%	100,0%	100,0%	

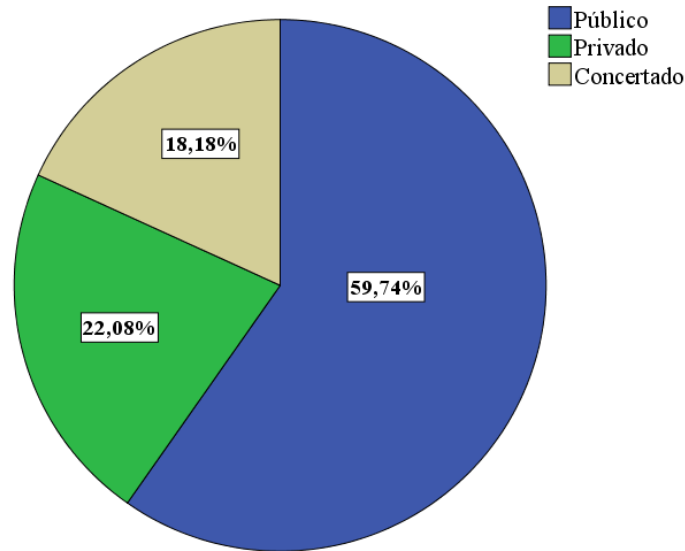


Figura D.3. Titularidad de los centros educativos invitados a la investigación

Tabla D.3. Tabla de contingencia entre la Titularidad de los centros invitados y la edición HoC

		Edición		Total	
		HoC-2013	HoC-2014		
Titularidad	Público	Recuento	9	37	46
		% dentro de Edición	39,1%	68,5%	59,7%
	Concertado	Recuento	6	8	14
		% dentro de Edición	26,1%	14,8%	18,2%
	Privado	Recuento	8	9	17
		% dentro de Edición	34,8%	16,7%	22,1%
Total	Recuento	23	54	77	
	% dentro de Edición	100,0%	100,0%	100,0%	

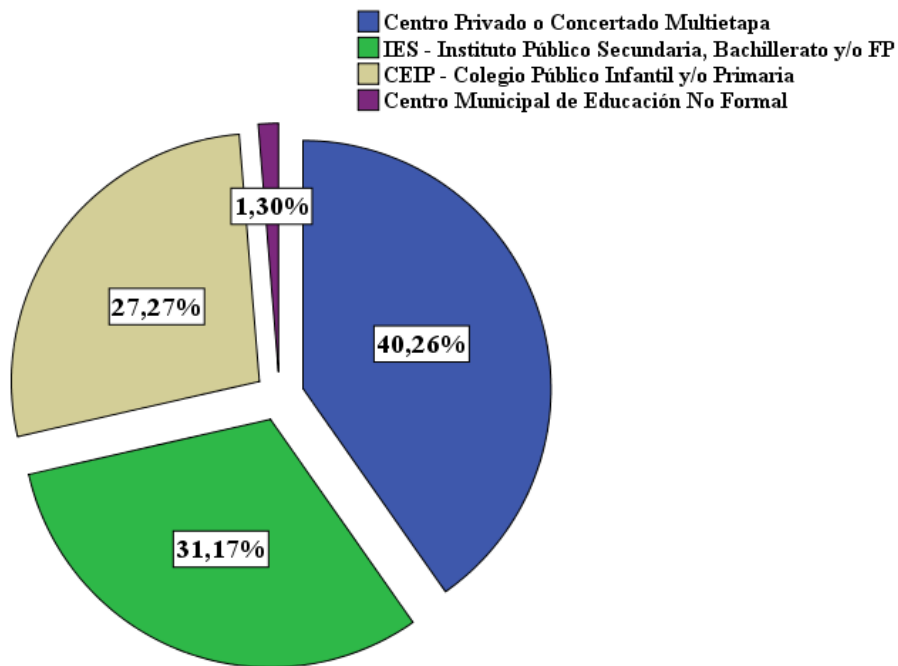


Figura D.4. Tipología de los centros educativos invitados a la investigación

Tabla D.4. Tabla de contingencia entre la Tipología de los centros invitados y la edición HoC

		Edición		Total	
		HoC-2013	HoC-2014		
Tipología	CEIP - Colegio Público Infantil y/o Primaria	Recuento	1	20	21
		% dentro de Edición	4,3%	37,0%	27,3%
	IES - Instituto Público Secundaria, Bachillerato y/o FP	Recuento	8	16	24
		% dentro de Edición	34,8%	29,6%	31,2%
	Centro Privado o Concertado Multietapa	Recuento	14	17	31
		% dentro de Edición	60,9%	31,5%	40,3%
Centro Municipal de Educación No Formal	Recuento	0	1	1	
	% dentro de Edición	0,0%	1,9%	1,3%	
Total	Recuento	23	54	77	
	% dentro de Edición	100,0%	100,0%	100,0%	

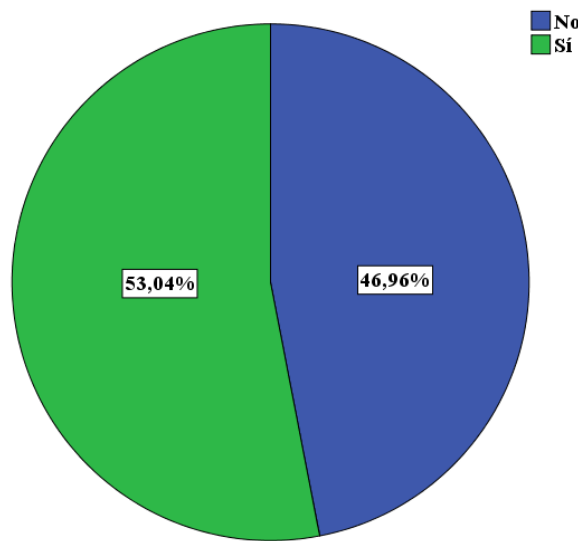


Figura D.5. Porcentajes de respuesta a la Pregunta 1 del cuestionario para estudiantes

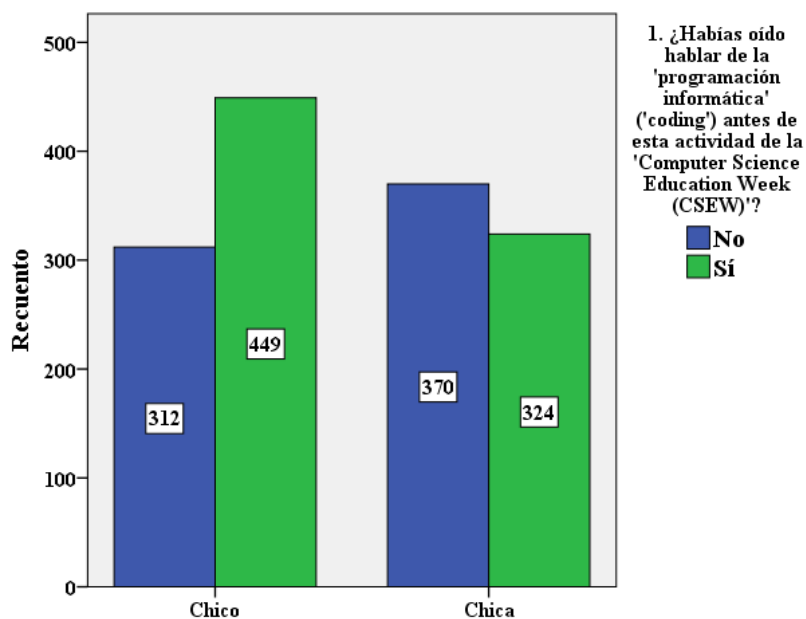


Figura D.6. Frecuencias de respuesta a la Pregunta 1 según sexo

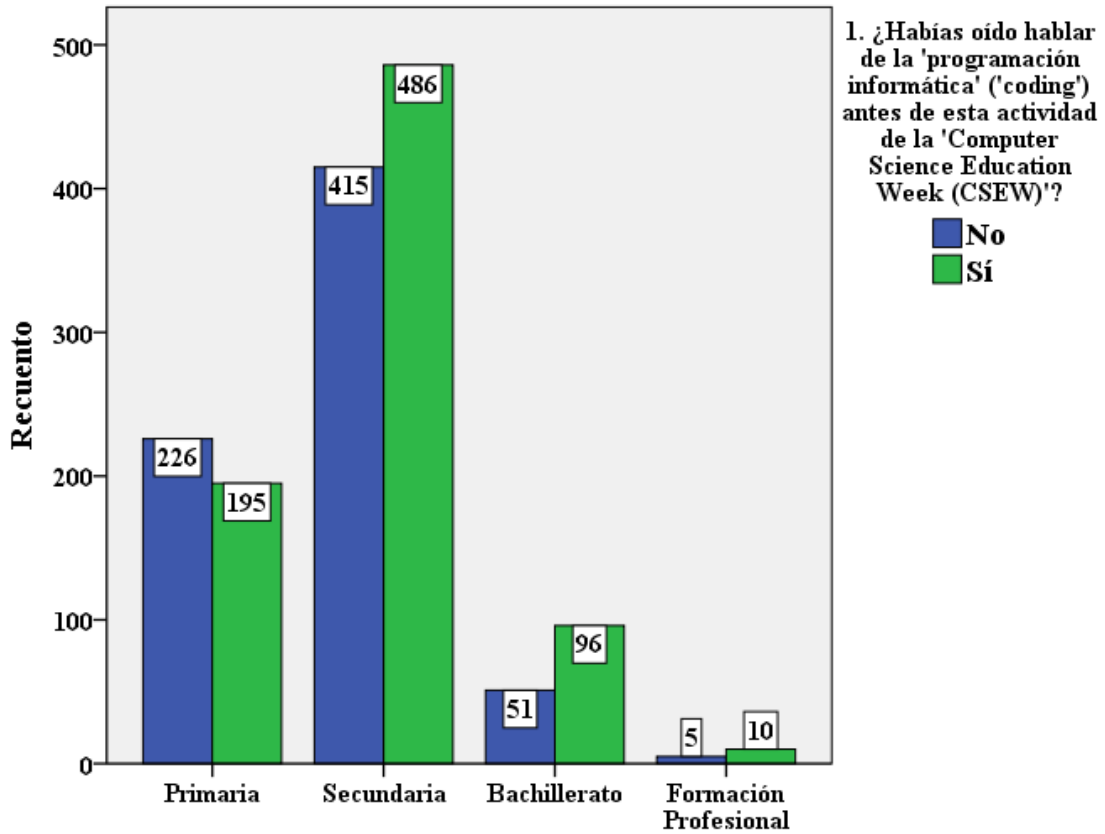


Figura D.7. Frecuencias de respuesta a la Pregunta 1 según etapa educativa

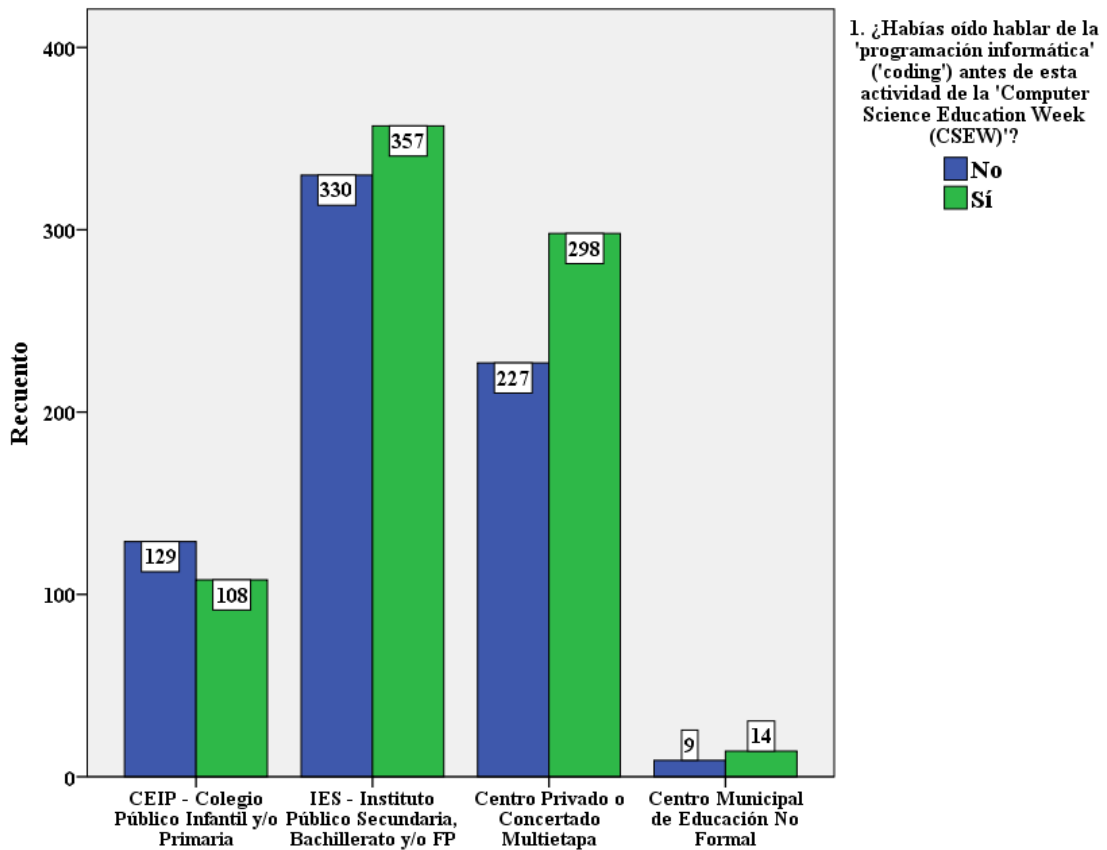


Figura D.8. Frecuencias de respuesta a la Pregunta 1 según la tipología de centro

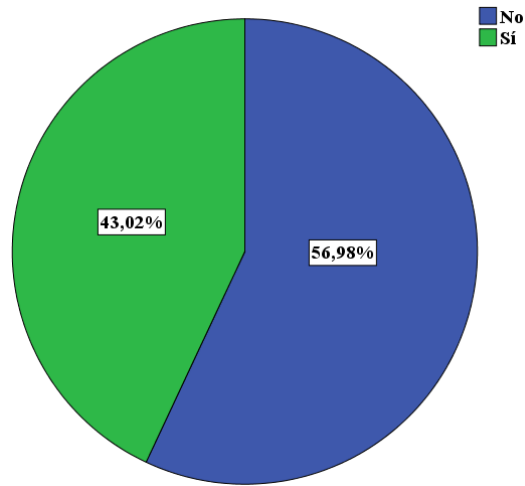


Figura D.9. Porcentajes de respuesta a la Pregunta 2 del cuestionario para estudiantes

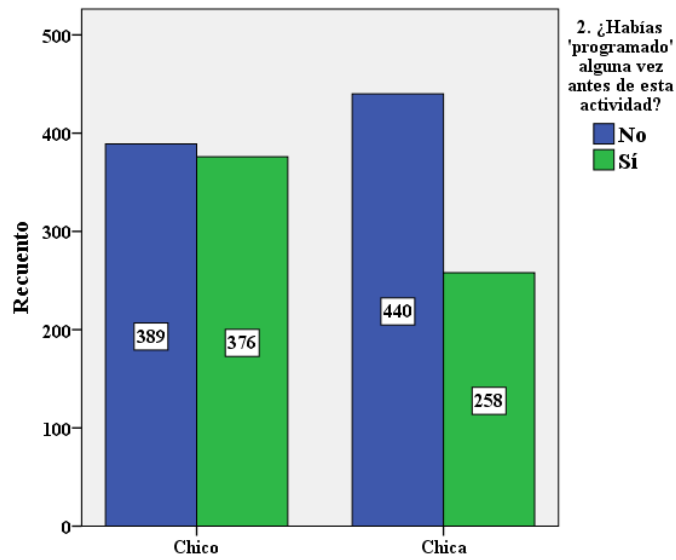


Figura D.10. Frecuencias de respuesta a la Pregunta 2 por sexo

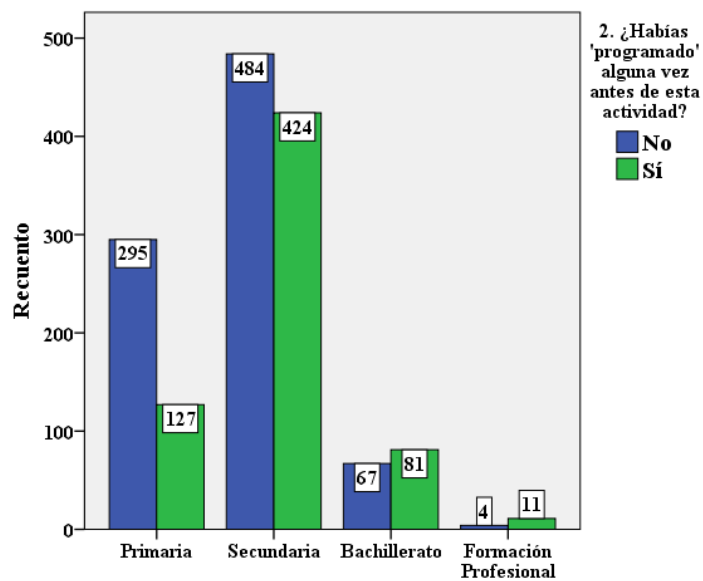


Figura D.11. Frecuencias de respuesta a la Pregunta 2 por etapa educativa

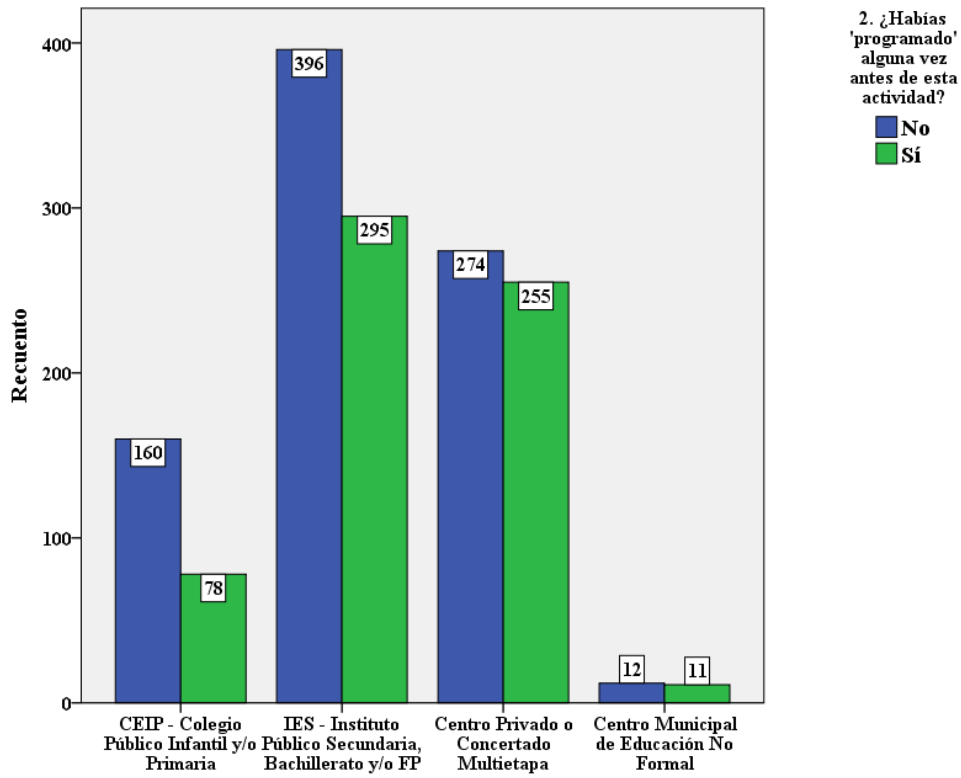


Figura D.12. Frecuencias en la Pregunta 2 según la tipología de centro

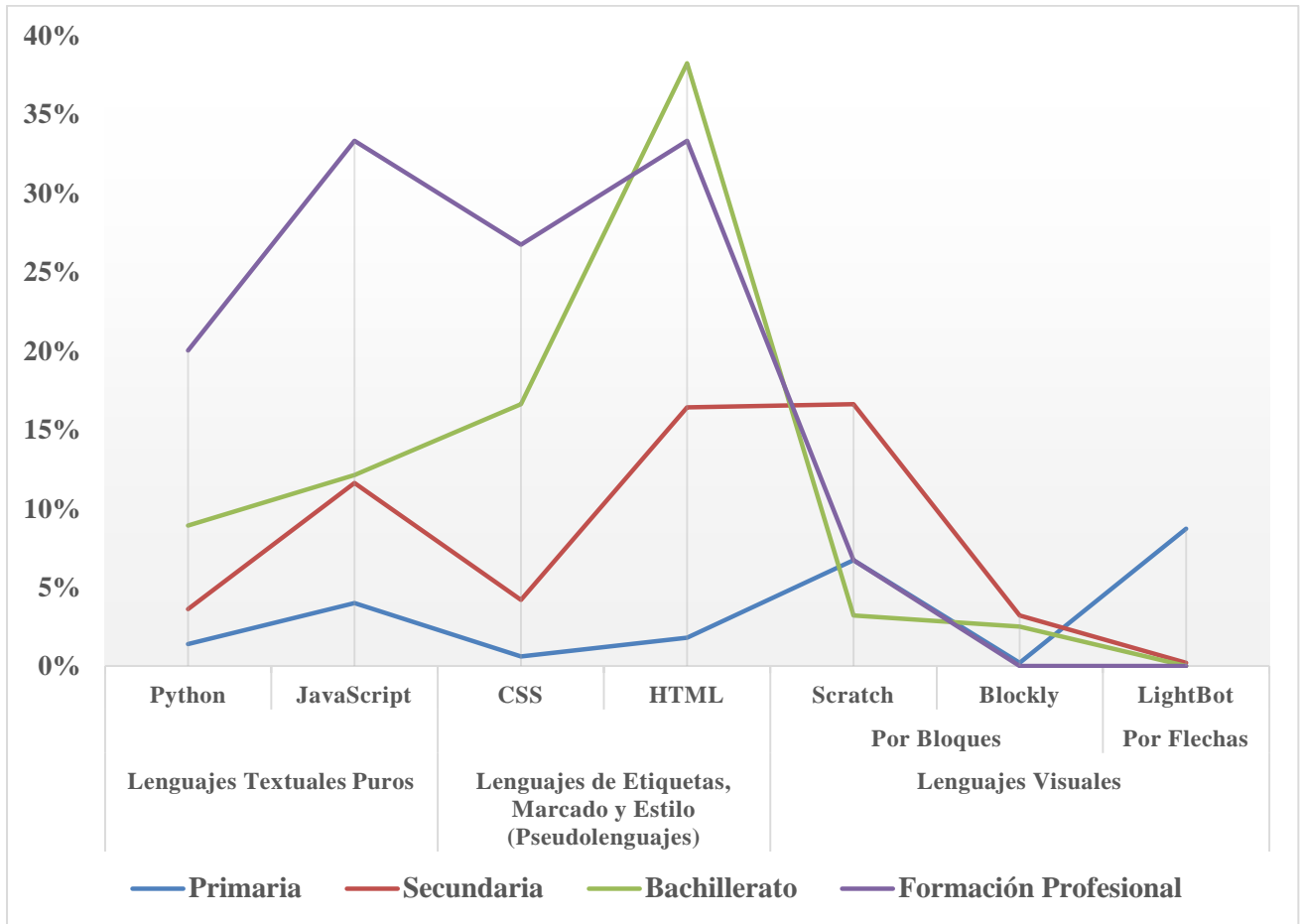


Figura D.13. Índices de penetración según lenguajes sobre el total de la muestra, y diferenciado por etapa educativa

Tabla D.5. Distribución de frecuencias para la Pregunta 4 ('pensamiento lógico')

<i>P4.1 ¿En qué medida crees que la actividad te ha ayudado a desarrollar... Pensamiento Lógico?</i>		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada	100	6,0	6,8	6,8
	Poco	106	6,4	7,2	14,1
	Algo	353	21,2	24,1	38,2
	Bastante	532	32,0	36,3	74,5
	Mucho	373	22,4	25,5	100,0
	Total	1464	88,1	100,0	
Perdidos	Sistema	198	11,9		
Total		1662	100,0		

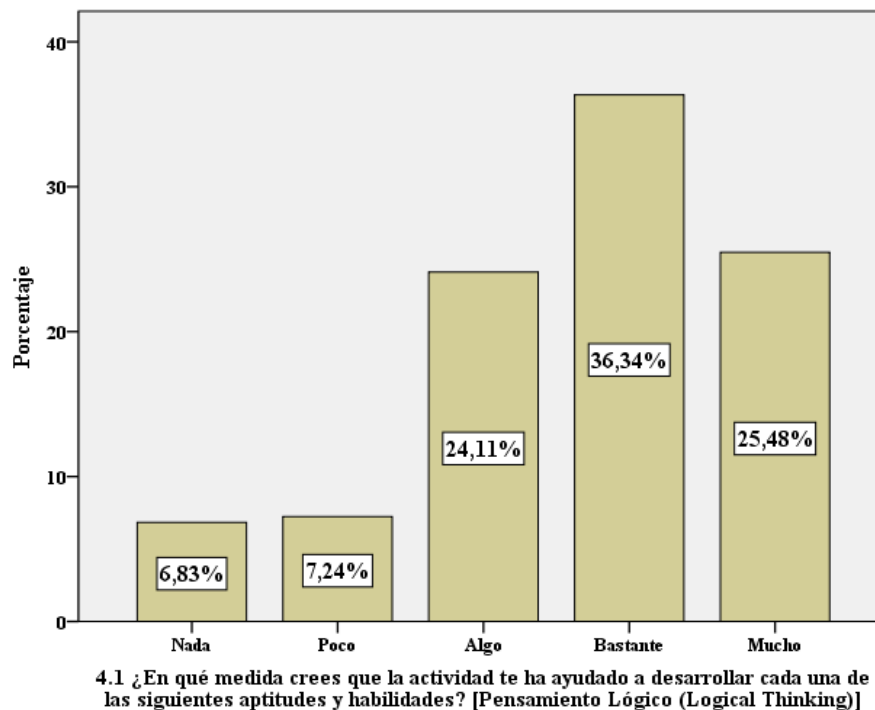
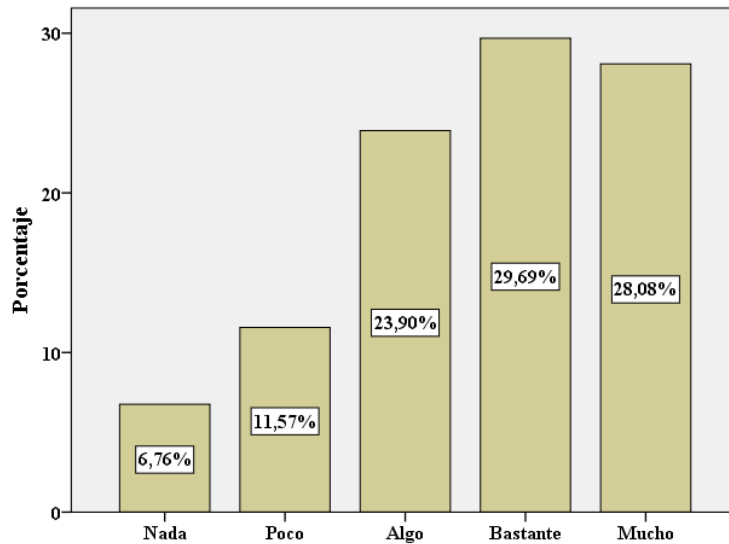


Figura D.14. Distribución de frecuencias para la Pregunta 4 ('pensamiento lógico')

Tabla D.6. Distribución de frecuencias para la Pregunta 4 ('creatividad')

<i>P4.2 ¿En qué medida crees que la actividad te ha ayudado a desarrollar... Creatividad?</i>		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada	97	5,8	6,8	6,8
	Poco	166	10,0	11,6	18,3
	Algo	343	20,6	23,9	42,2
	Bastante	426	25,6	29,7	71,9
	Mucho	403	24,2	28,1	100,0
	Total	1435	86,3	100,0	
Perdidos	Sistema	227	13,7		
Total		1662	100,0		

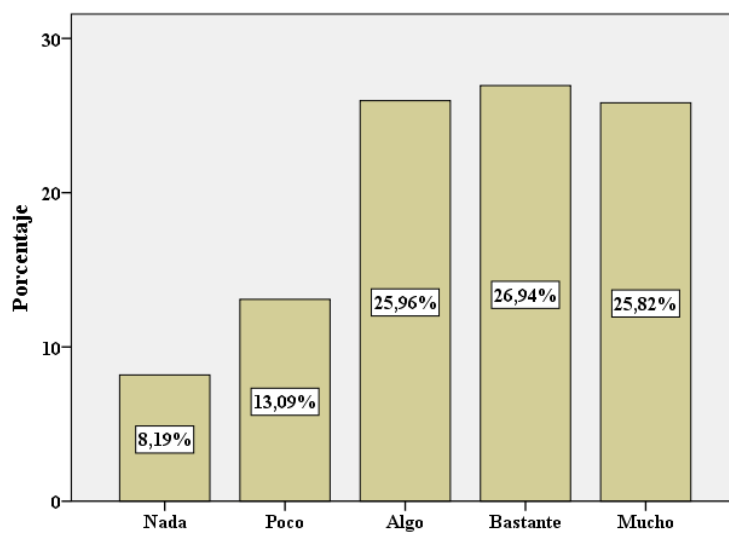


4.2 ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades? [Creatividad (Creativity)]

Figura D.15. Distribución de frecuencias para la Pregunta 4 ('creatividad')

Tabla D.7. Distribución de frecuencias para la Pregunta 4 ('resolución de problemas')

<i>P4.3 ¿En qué medida crees que la actividad te ha ayudado a desarrollar... Resolución de Problemas?</i>					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada	117	7,0	8,2	8,2
	Poco	187	11,3	13,1	21,3
	Algo	371	22,3	26,0	47,2
	Bastante	385	23,2	26,9	74,2
	Mucho	369	22,2	25,8	100,0
	Total	1429	86,0	100,0	
Perdidos	Sistema	233	14,0		
Total		1662	100,0		



4.3 ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades? [Resolución de Problemas (Problem Solving)]

Figura D.16. Distribución de frecuencias para la Pregunta 4 ('resolución de problemas')

Tabla D.8. Distribución de frecuencias para la Pregunta 4 ('trabajo en equipo')

<i>P4.4 ¿En qué medida crees que la actividad te ha ayudado a desarrollar... Trabajo en Equipo?</i>				
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada	354	21,3	24,6
	Poco	222	13,4	15,4
	Algo	249	15,0	17,3
	Bastante	278	16,7	19,3
	Mucho	336	20,2	23,3
	Total	1439	86,6	100,0
Perdidos	Sistema	223	13,4	
Total		1662	100,0	

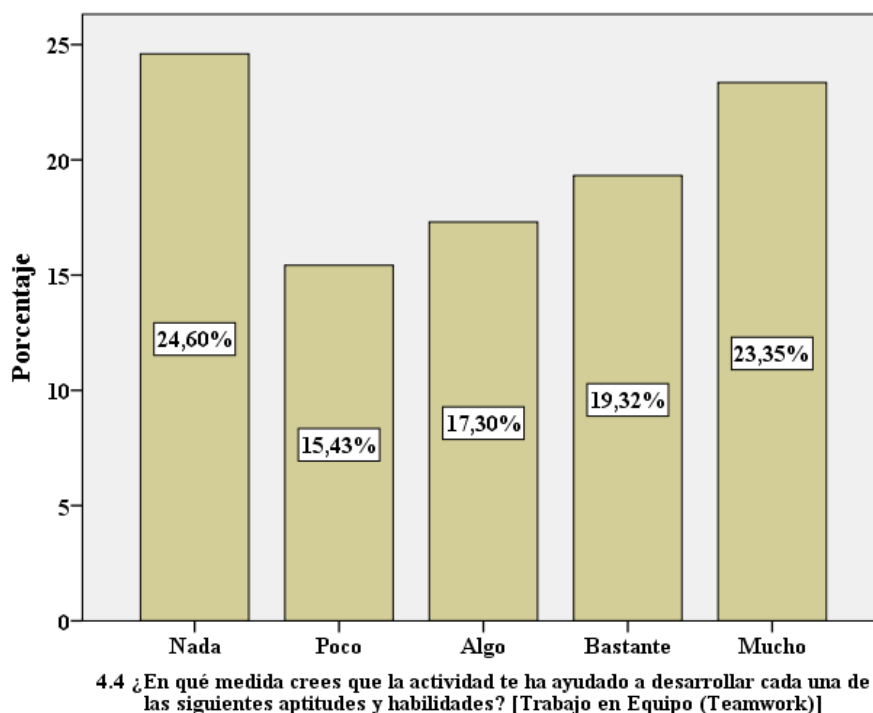


Figura D.17. Distribución de frecuencias para la Pregunta 4 ('trabajo en equipo')

Tabla D.9. Distribución de frecuencias para la Pregunta 4 ('persistencia-perseverancia')

<i>P4.5 ¿En qué medida crees que la actividad te ha ayudado a desarrollar... Persistencia-Perseverancia?</i>				
	Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada	127	7,6	8,9
	Poco	195	11,7	13,7
	Algo	380	22,9	26,7
	Bastante	365	22,0	25,7
	Mucho	355	21,4	25,0
	Total	1422	85,6	100,0
Perdidos	Sistema	240	14,4	
Total		1662	100,0	

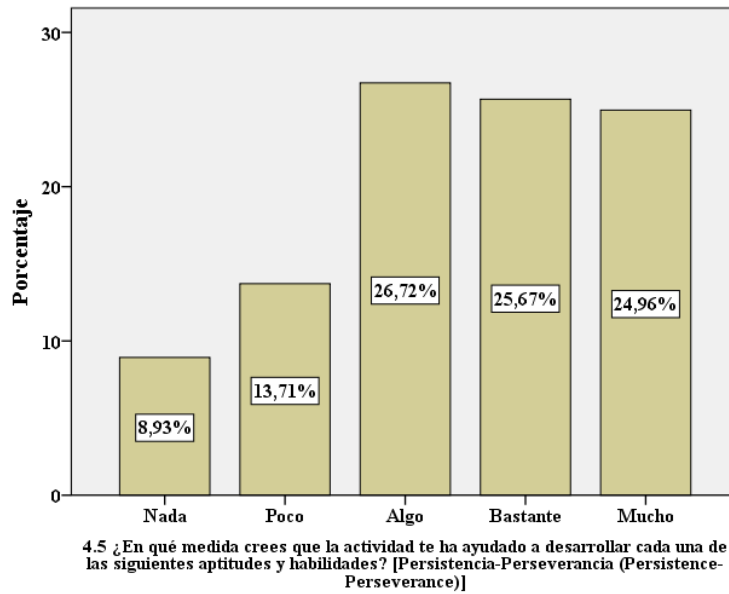


Figura D.18. Distribución de frecuencias para la Pregunta 4 ('persistencia-perseverancia')

Tabla D.10. Distribución de frecuencias para la Pregunta 4 ('capacidad de atención')

<i>P4.6 ¿En qué medida crees que la actividad te ha ayudado a desarrollar... Capacidad de Atención?</i>					
		Frecuencia	Porcentaje	Porcentaje válido	Porcentaje acumulado
Válido	Nada	101	6,1	7,0	7,0
	Poco	132	7,9	9,2	16,2
	Algo	266	16,0	18,5	34,6
	Bastante	427	25,7	29,6	64,3
	Mucho	515	31,0	35,7	100,0
	Total	1441	86,7	100,0	
Perdidos	Sistema	221	13,3		
Total		1662	100,0		

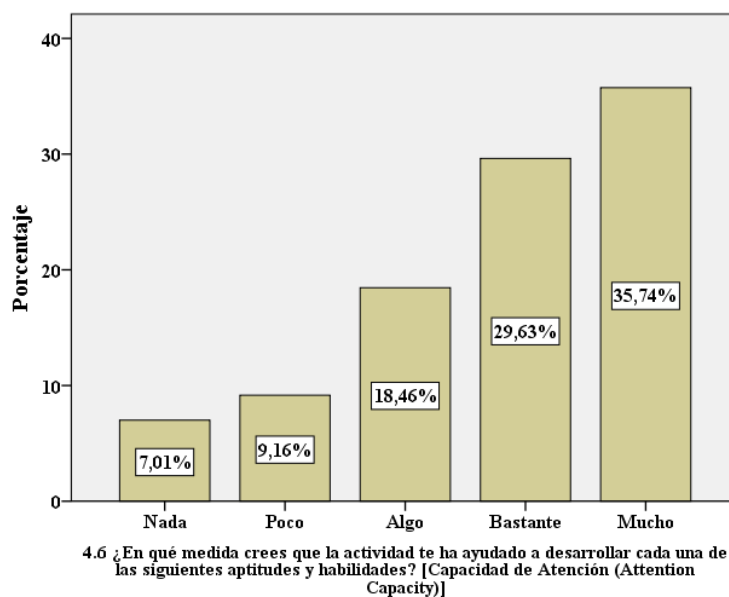


Figura D.19. Distribución de frecuencias para la Pregunta 4 ('capacidad de atención')

Tabla D.11. Diferencias por sexo para la Pregunta 4

<i>P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?</i>		N	Media	Desviación estándar	t de Student	Significación de la diferencia P(t)
Pensamiento Lógico	Chicos	741	3,66	1,193	-0,335	,738
	Chicas	682	3,68	1,071		
Creatividad	Chicos	723	3,64	1,245	1,137	,256
	Chicas	673	3,57	1,146		
Resolución de Problemas	Chicos	728	3,54	1,268	1,758	,079
	Chicas	663	3,43	1,184		
Trabajo en Equipo	Chicos	731	3,00	1,537	-0,422	,673
	Chicas	669	3,03	1,468		
Persistencia-Perseverancia	Chicos	718	3,49	1,293	1,604	,109
	Chicas	666	3,38	1,196		
Capacidad de Atención	Chicos	733	3,74	1,266	-1,133	,257
	Chicas	669	3,81	1,171		

Tabla D.12. Diferencias por etapa educativa para la Pregunta 4

<i>P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?</i>		N	Media	Desviación estándar	F	Significación de la diferencia P(F)
Pensamiento Lógico	Primaria	404	4,00	1,063	18,431**	,000
	Secundaria	889	3,55	1,104		
	Bachillerato	145	3,53	1,225		
	FP	14	2,86	1,406		
Creatividad	Primaria	394	3,94	1,117	13,959**	,000
	Secundaria	870	3,49	1,203		
	Bachillerato	145	3,52	1,155		
	FP	15	3,40	1,502		
Resolución de Problemas	Primaria	395	3,83	1,198	14,446**	,000
	Secundaria	864	3,36	1,207		
	Bachillerato	144	3,38	1,257		
	FP	15	3,07	1,438		
Trabajo en Equipo	Primaria	397	3,70	1,392	41,060**	,000
	Secundaria	871	2,75	1,450		
	Bachillerato	145	2,74	1,540		
	FP	15	3,20	1,568		
Persistencia-Perseverancia	Primaria	395	3,69	1,254	8,313**	,000
	Secundaria	856	3,32	1,216		
	Bachillerato	145	3,47	1,275		
	FP	15	3,20	1,373		
Capacidad de Atención	Primaria	396	4,28	1,060	33,058**	,000
	Secundaria	876	3,60	1,207		
	Bachillerato	144	3,61	1,263		
	FP	15	3,07	1,438		

** Significativo al nivel $p_{(F)} < 0,01$

Tabla D.13. Pruebas KMO y Bartlett para la matriz de correlaciones relativa a la Pregunta 4

Prueba de KMO y Bartlett	
Medida Kaiser-Meyer-Olkin de adecuación de muestreo	,869
Aprox. Chi-cuadrado	2797,634**
Prueba de esfericidad de Bartlett	gl
	15
	Sig.
	,000

** Significativo al nivel $p_{(2)} < 0,01$

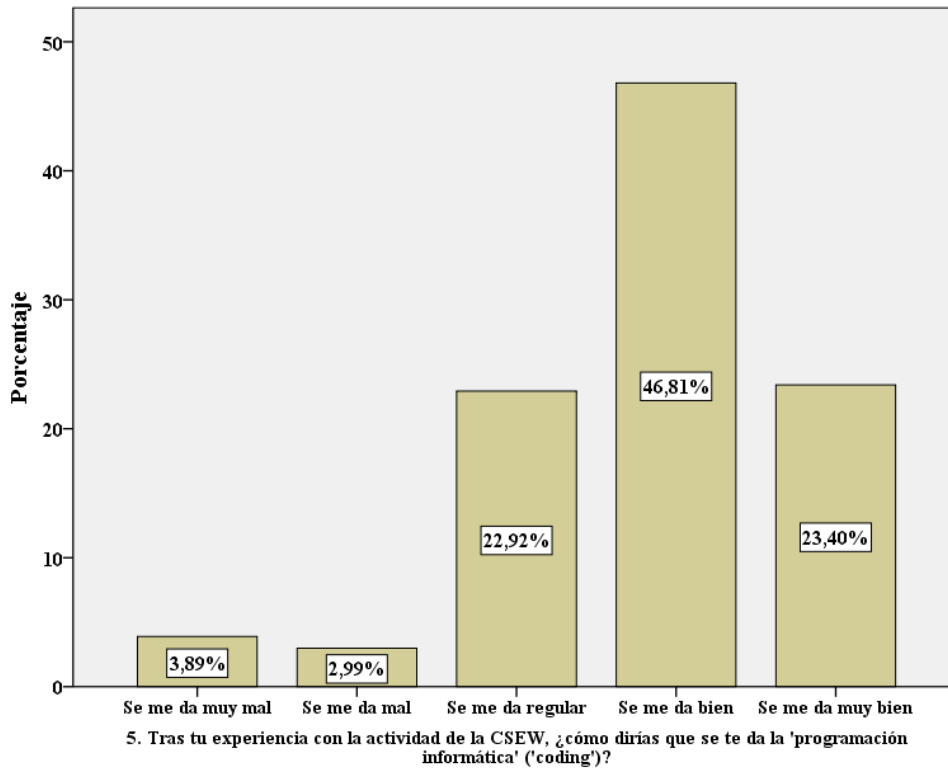


Figura D.20. Distribución de frecuencias en la Pregunta 5 del cuestionario para estudiantes

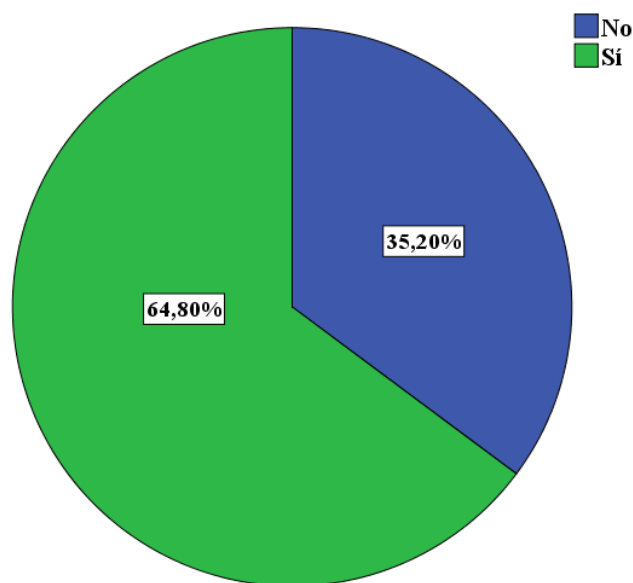


Figura D.21. Distribución de frecuencias en la Pregunta 6 del cuestionario para estudiantes

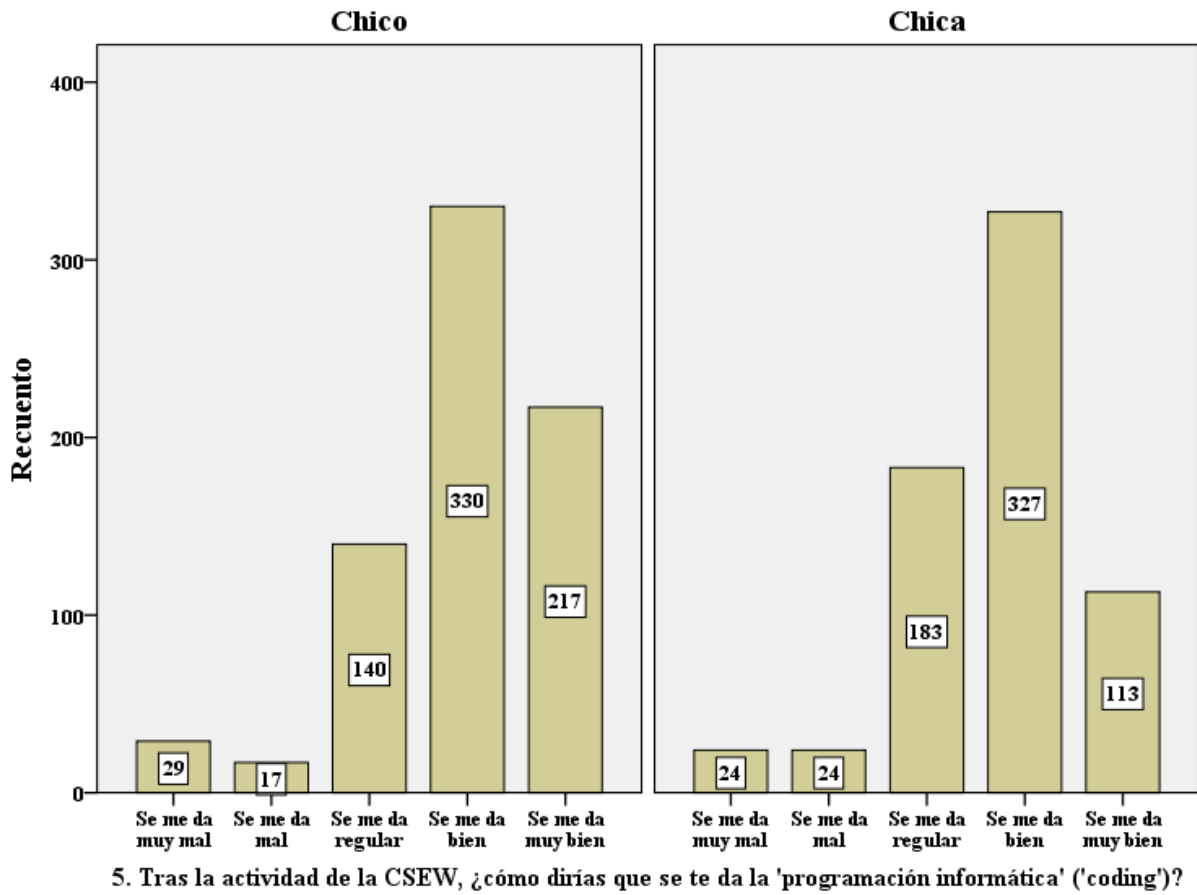


Figura D.22. Distribución de frecuencias en la Pregunta 5, según el sexo

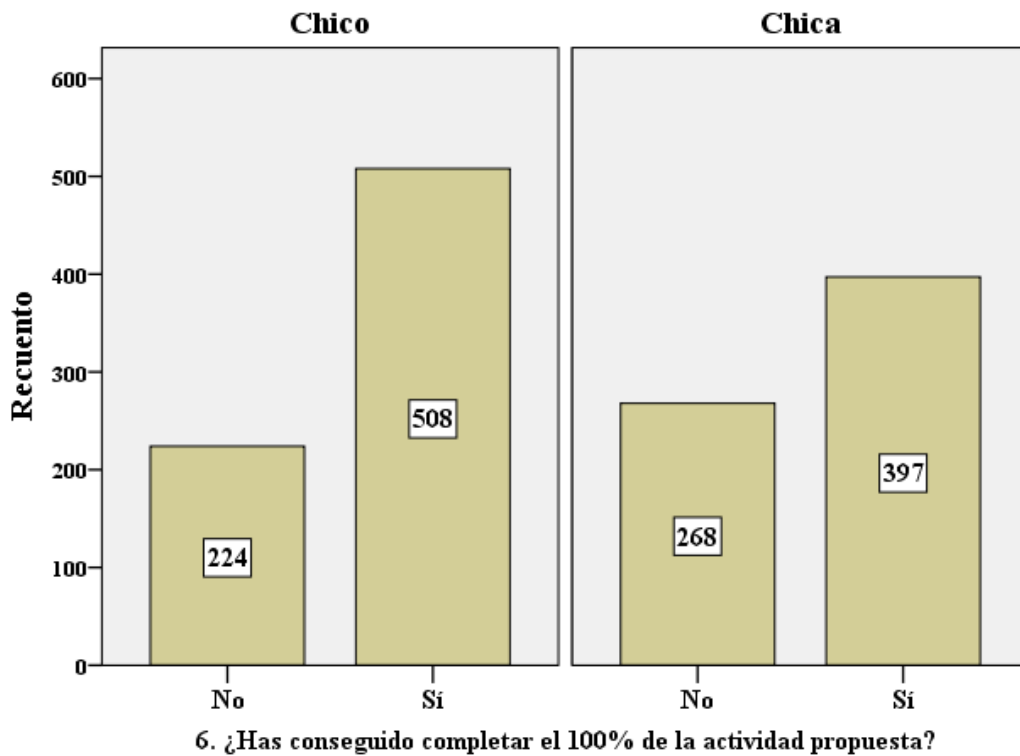


Figura D.23. Distribución de frecuencias en la Pregunta 6, según el sexo

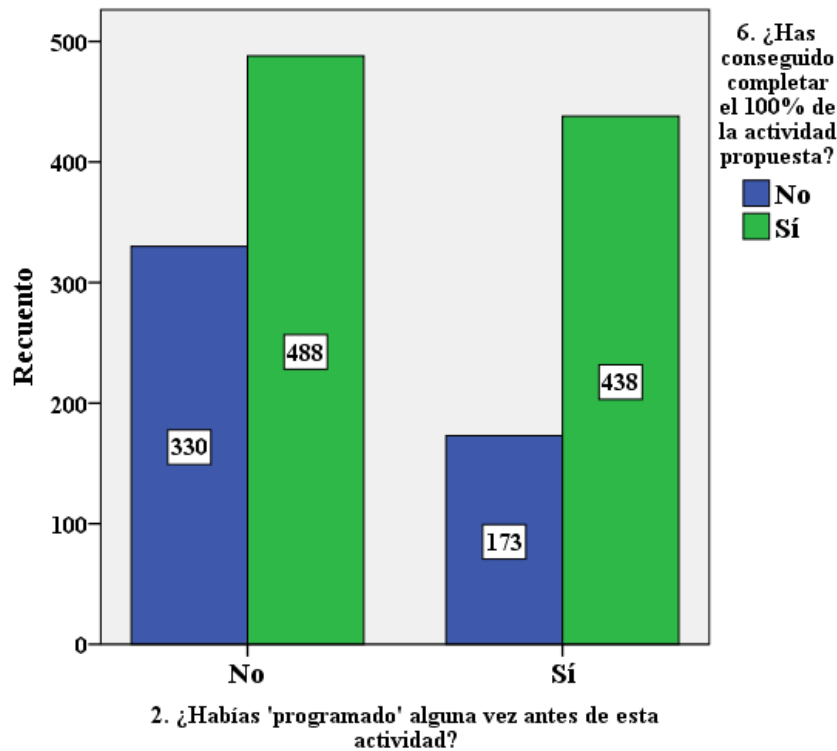


Figura D.24. Distribución de frecuencias en la Pregunta 6 en función de la respuesta a la Pregunta 2

Tabla D.14. ANOVAs en función de la etapa educativa para las preguntas 5 y 6

	Etapa Educativa	N	Media	Desviación estándar	F	p(F)
<i>P5. Autoeficacia 'coding'</i>	Primaria	396	4,07	,883	14,838**	,000
	Secundaria	877	3,71	,941		
	Bachillerato	143	3,96	,934		
	Formación Profesional	15	3,80	1,612		
<i>P6. Completar actividad</i>	Primaria	388	,66	,474	12,656**	,000
	Secundaria	876	,61	,489		
	Bachillerato	142	,87	,342		
	Formación Profesional	15	,73	,458		

** Significativo al nivel $p(F) < 0,01$

Tabla D.15. Diferencias en la Pregunta 4 en función de la Pregunta 5

<i>P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar...?</i>	<i>P5. Autoeficacia 'coding'</i>	N	Media	Desviación estándar	F	p(F)
Pensamiento Lógico	Se me da muy mal	55	2,33	1,428	33,678**	,000
	Se me da mal	42	3,19	1,018		
	Se me da regular	322	3,45	,995		
	Se me da bien	673	3,78	1,017		
	Se me da muy bien	330	3,95	1,214		
Creatividad	Se me da muy mal	53	2,30	1,339	22,591**	,000
	Se me da mal	41	3,20	1,145		
	Se me da regular	315	3,49	1,104		
	Se me da bien	662	3,67	1,109		
	Se me da muy bien	325	3,84	1,272		

<i>P4. ¿En qué medida crees que la actividad te ha ayudado a desarrollar...?</i>	<i>P5. Autoeficacia 'coding'</i>	N	Media	Desviación estándar	F	p(F)
Resolución de Problemas	Se me da muy mal	55	2,35	1,404	22,375**	,000
	Se me da mal	40	2,85	1,189		
	Se me da regular	314	3,30	1,122		
	Se me da bien	657	3,60	1,115		
	Se me da muy bien	326	3,73	1,352		
Trabajo en Equipo	Se me da muy mal	55	2,40	1,662	10,735**	,000
	Se me da mal	41	2,49	1,502		
	Se me da regular	316	2,89	1,393		
	Se me da bien	663	2,95	1,470		
	Se me da muy bien	326	3,42	1,567		
Persistencia-Perseverancia	Se me da muy mal	55	2,45	1,549	21,775**	,000
	Se me da mal	42	2,81	1,153		
	Se me da regular	312	3,21	1,145		
	Se me da bien	650	3,50	1,153		
	Se me da muy bien	326	3,79	1,325		
Capacidad de Atención	Se me da muy mal	53	2,47	1,539	25,040**	,000
	Se me da mal	42	3,29	1,293		
	Se me da regular	319	3,59	1,100		
	Se me da bien	659	3,89	1,094		
	Se me da muy bien	330	4,01	1,312		

** Significativo al nivel $p(F) < 0,01$

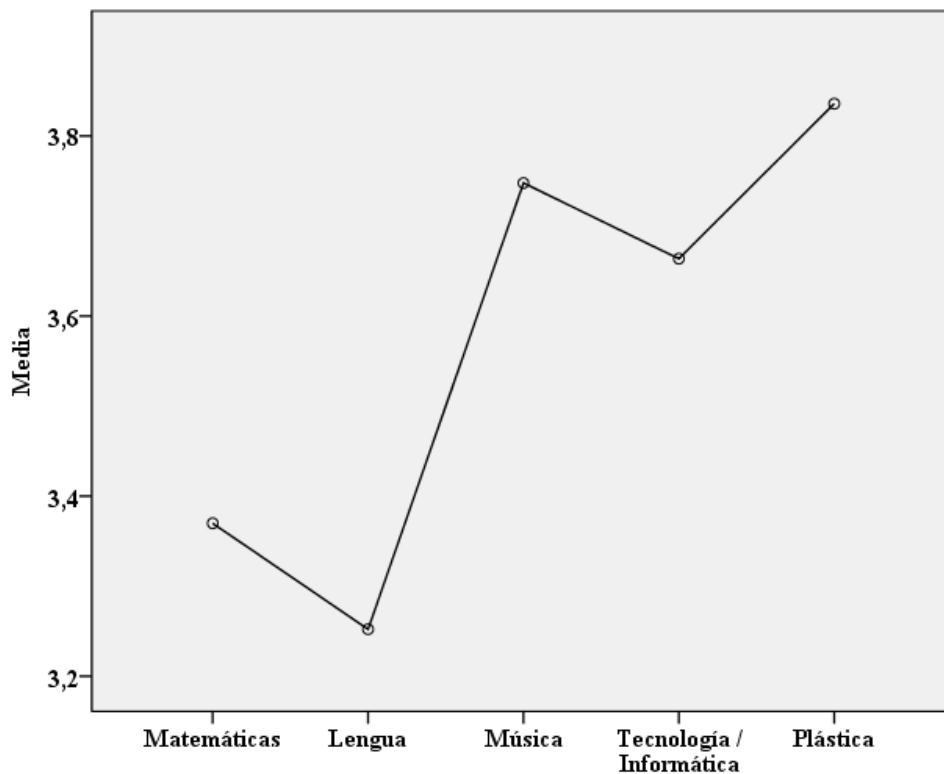


Figura D.25. Medias relativas a la Pregunta 7 del cuestionario para estudiantes

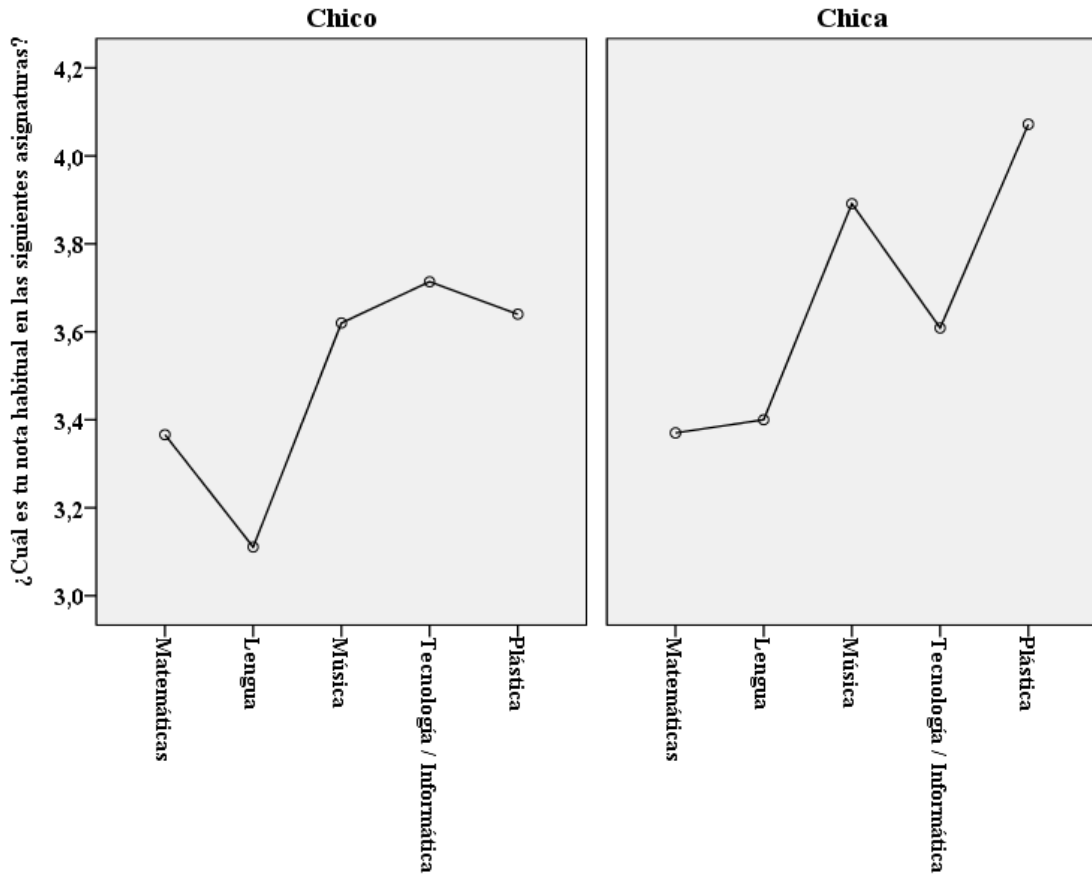


Figura D.26. Medias en la Pregunta 7, diferenciadas por sexo

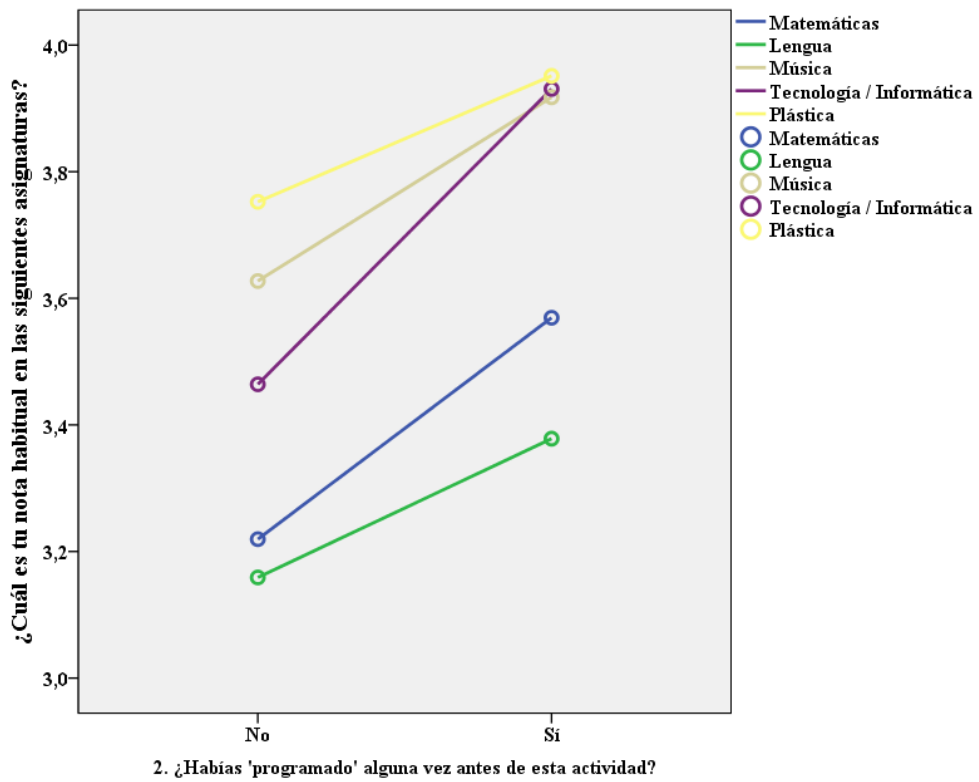


Figura D.27. Medias en la Pregunta 7, en función de la experiencia previa en 'coding' (Pregunta 2)

Tabla D.16. Diferencias en la Pregunta 7, en función de la percepción de autoeficacia (Pregunta 5)

<i>P7. ¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?</i>	<i>P5. Autoeficacia 'coding'</i>	N	Media	Desviación estándar	F	p(F)
Matemáticas	Se me da muy mal	56	2,71	1,510	12,933**	,000
	Se me da mal	43	3,09	1,411		
	Se me da regular	329	3,05	1,340		
	Se me da bien	669	3,38	1,341		
	Se me da muy bien	330	3,69	1,342		
Lengua	Se me da muy mal	56	2,82	1,608	5,778**	,000
	Se me da mal	43	3,23	1,250		
	Se me da regular	324	3,06	1,275		
	Se me da bien	664	3,22	1,242		
	Se me da muy bien	332	3,46	1,235		
Música	Se me da muy mal	54	3,43	1,561	4,987**	,001
	Se me da mal	41	3,68	1,350		
	Se me da regular	306	3,56	1,274		
	Se me da bien	618	3,73	1,215		
	Se me da muy bien	320	3,96	1,205		
Tecnología / Informática	Se me da muy mal	53	3,00	1,506	21,916**	,000
	Se me da mal	43	3,19	1,314		
	Se me da regular	316	3,28	1,283		
	Se me da bien	617	3,70	1,237		
	Se me da muy bien	313	4,08	1,124		
Plástica	Se me da muy mal	53	3,28	1,634	4,986**	,001
	Se me da mal	40	3,80	1,265		
	Se me da regular	303	3,68	1,252		
	Se me da bien	646	3,87	1,158		
	Se me da muy bien	323	3,97	1,188		

** Significativo al nivel $p(F) < 0,01$

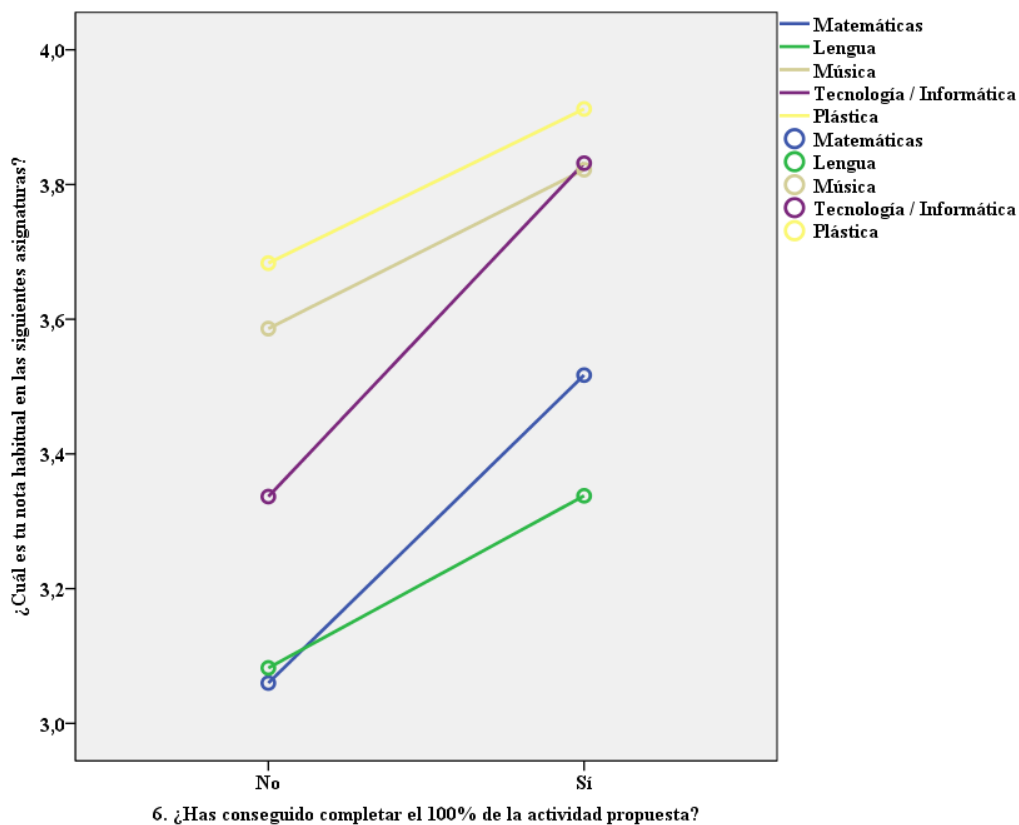


Figura D.28. Medias en la Pregunta 7, en función de haber conseguido completar la actividad HoC (Pregunta 6)

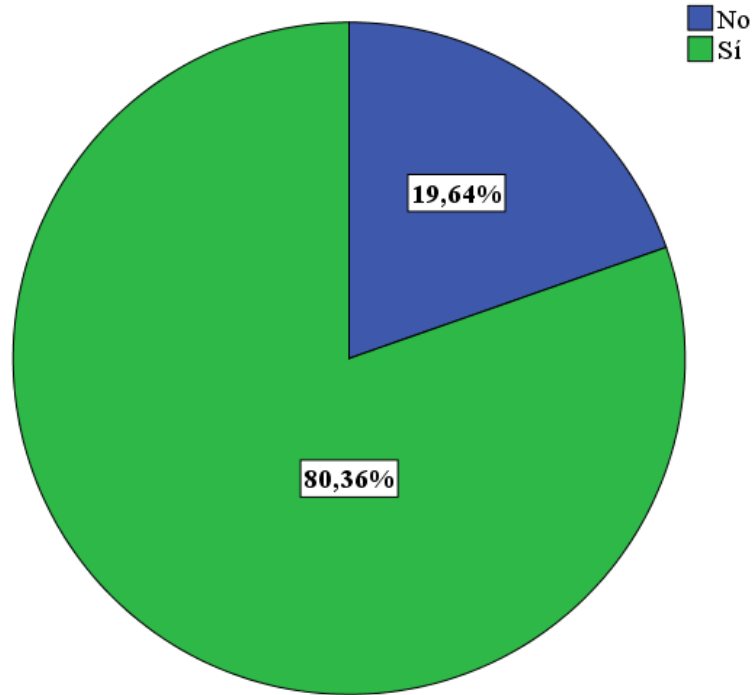


Figura D.29. Distribución de frecuencias de las respuestas a la Pregunta 9 del cuestionario para estudiantes

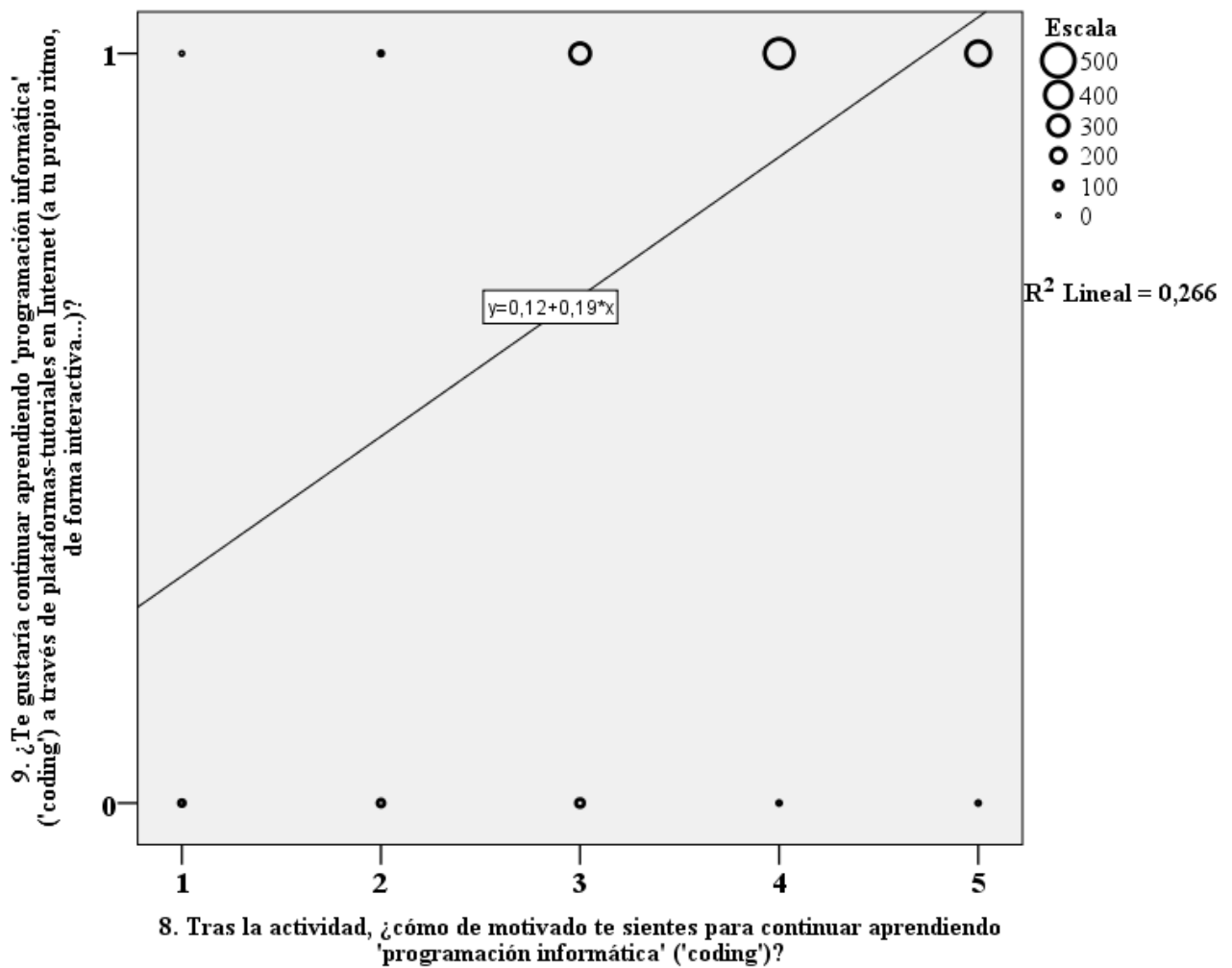


Figura D.30. Diagrama de dispersión de las respuestas a las preguntas 8 y 9

Tabla D.17. Diferencias para las preguntas 8 y 9, en función de la etapa educativa

		N	Media	Desviación estándar	F	P(F)
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	Primaria	388	4,26	,903	51,096**	,000
	Secundaria	873	3,48	1,071		
	Bachillerato	144	3,54	1,121		
	Formación Profesional	15	3,67	1,633		
<i>P9. Determinación aprender 'coding' on-line</i>	Primaria	386	,90	,295	11,240**	,000
	Secundaria	878	,77	,423		
	Bachillerato	145	,77	,421		
	Formación Profesional	14	,79	,426		

** Significativo al nivel $p_{(F)} < 0,01$

Tabla D.18. Diferencias en las preguntas 8 y 9, en función de la percepción de autoeficacia (Pregunta 5)

	<i>P5. Autoeficacia 'coding'</i>	N	Media	Desviación estándar	F	P(F)
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	Se me da muy mal	54	2,02	1,296	115,223**	,000
	Se me da mal	42	2,55	,968		
	Se me da regular	316	3,26	1,003		
	Se me da bien	662	3,79	,912		
	Se me da muy bien	330	4,35	,908		
<i>P9. Determinación aprender 'coding' on-line</i>	Se me da muy mal	55	,27	,449	47,681**	,000
	Se me da mal	42	,52	,505		
	Se me da regular	320	,73	,446		
	Se me da bien	657	,85	,361		
	Se me da muy bien	325	,92	,267		

** Significativo al nivel $p_{(F)} < 0,01$

Tabla D.19. Estadísticos descriptivos de las variables incluidas en el análisis de clúster

	N	Mínimo	Máximo	Media	Desviación estándar
<i>P1. Haber oído 'coding'</i>	1495	0	1	,53	,499
<i>P2. Experiencia previa 'coding'</i>	1504	0	1	,43	,495
<i>P5. Autoeficacia 'coding'</i>	1440	1	5	3,83	,951
<i>P6. Completar actividad HoC</i>	1432	0	1	,65	,478
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	1430	1	5	3,70	1,097
<i>P9. Determinación aprender 'coding' on-line</i>	1436	0	1	,80	,397
N válido (por lista)	1352				

Tabla D.20. Centros de clústeres iniciales (expresados como puntuaciones típicas o Zscore)

	Clúster			
	1	2	3	4
<i>Zscore: P1. Haber oído 'coding'</i>	-1,06248	,94056	-1,06248	,94056
<i>Zscore: P2. Experiencia previa 'coding'</i>	-,86859	1,15052	1,15052	-,86859
<i>Zscore: P5. Autoeficacia 'coding'</i>	-2,97512	1,23227	1,23227	-2,97512
<i>Zscore: P6. Completar actividad HoC</i>	-1,35646	,73670	-1,35646	,73670
<i>Zscore: P8. Motivación proseguir aprendizaje 'coding'</i>	-2,45855	-2,45855	1,18752	1,18752
<i>Zscore: P9. Determinación aprender 'coding' on-line</i>	-2,02221	-2,02221	,49416	,49416

Tabla D.21. Historial de iteraciones (convergencia conseguida en 10 iteraciones)

Iteración	Cambiar en centros de clústeres			
	1	2	3	4
1	1,902	2,148	2,195	2,621
2	,144	,194	,083	,242
3	,052	,450	,244	,404
4	,138	,250	,359	,445
5	,042	,085	,400	,357
6	,076	,060	,147	,119
7	,066	,048	,129	,114
8	,037	,000	,173	,144
9	,085	,000	,000	,020
10	,000	,000	,000	,000

Convergencia conseguida en 10 iteraciones, debido a que no hay ningún cambio en los centros de clústeres.

Tabla D.22. ANOVA de diferencias entre los clústeres para todas las variables incluidas en el análisis

	ANOVA				F	Sig.
	Clúster		Error			
	Media cuadrática	gl	Media cuadrática	gl		
<i>Zscore: P1. Haber oído 'coding'</i>	71,423	3	,844	1348	84,670**	,000
<i>Zscore: P2. Experiencia previa 'coding'</i>	380,311	3	,157	1348	2423,786**	,000
<i>Zscore: P5. Autoeficacia 'coding'</i>	137,602	3	,688	1348	200,125**	,000
<i>Zscore: P6. Completar actividad HoC</i>	28,410	3	,940	1348	30,219**	,000
<i>Zscore: P8. Motivación proseguir aprendizaje 'coding'</i>	145,411	3	,669	1348	217,344**	,000
<i>Zscore: P9. Determinación aprender 'coding' on-line</i>	408,791	3	,077	1348	5330,431**	,000

** Significativo al nivel $p_{(F)} < 0,01$

Tabla D.23. Tabla de contingencia sexo * clúster

		Número de caso de clúster				Total	
		1	2	3	4		
Sexo	Chico	Recuento	55	64	295	272	686
		% dentro de Sexo	8,0%	9,3%	43,0%	39,7%	100,0%
	Chica	Recuento	78	76	185	300	639
		% dentro de Sexo	12,2%	11,9%	29,0%	46,9%	100,0%
Total	Recuento	133	140	480	572	1325	
	% dentro de Sexo	10,0%	10,6%	36,2%	43,2%	100,0%	

Tabla D.24. Tabla de contingencia titularidad * clúster

		Número de caso de clúster				Total	
		1	2	3	4		
Titularidad	Público	Recuento	77	93	288	409	867
		% dentro de Titularidad	8,9%	10,7%	33,2%	47,2%	100,0%
	Concertado	Recuento	3	2	4	11	20
		% dentro de Titularidad	15,0%	10,0%	20,0%	55,0%	100,0%
	Privado	Recuento	53	44	188	158	443
		% dentro de Titularidad	12,0%	9,9%	42,4%	35,7%	100,0%

Tabla D.25. Estadísticos descriptivos de las variables inicialmente incluidas en la regresión múltiple

	Media	Desviación estándar	N
<i>Variable dependiente: P8. Motivación proseguir aprendizaje 'coding'</i>	3,70	1,098	1062
<i>Sexo Dummy</i>	,47	,499	1062
<i>Etapa Educativa Dummy</i>	,75	,433	1062
<i>Titularidad Dummy</i>	,35	,478	1062
<i>P1. Haber oído 'coding'</i>	,53	,499	1062
<i>P2. Experiencia previa 'coding'</i>	,43	,496	1062
<i>P3. HTML</i>	,17	,378	1062
<i>P3. CSS</i>	,06	,240	1062
<i>P3. JavaScript</i>	,10	,306	1062
<i>P3. jQuery</i>	,01	,101	1062
<i>P3. PHP</i>	,02	,146	1062
<i>P3. Python</i>	,04	,197	1062
<i>P3. Ruby</i>	,02	,129	1062
<i>P4.1 Desarrollar Pensamiento Lógico</i>	3,68	1,127	1062
<i>P4.2 Desarrollar Creatividad</i>	3,61	1,194	1062
<i>P4.3 Desarrollar Resolución de Problemas</i>	3,49	1,249	1062
<i>P4.4 Desarrollar Trabajo en Equipo</i>	3,00	1,522	1062
<i>P4.5 Desarrollar Persistencia-Perseverancia</i>	3,46	1,260	1062
<i>P4.6 Desarrollar Capacidad de Atención</i>	3,79	1,219	1062
<i>P5. Autoeficacia 'coding'</i>	3,86	,960	1062
<i>P6. Completar actividad HoC</i>	,67	,469	1062
<i>P7.1 Rendimiento Matemáticas</i>	3,40	1,348	1062
<i>P7.2 Rendimiento Lengua</i>	3,27	1,274	1062
<i>P7.3 Rendimiento Música</i>	3,78	1,232	1062
<i>P7.4 Rendimiento Tecnología / Informática</i>	3,71	1,240	1062
<i>P7.5 Rendimiento Plástica</i>	3,87	1,180	1062

Tabla D.26. Estadísticas de residuos del modelo de regresión múltiple

	Estadísticas de residuos ^a				N
	Mínimo	Máximo	Media	Desviación estándar	
Valor pronosticado	1,13	5,21	3,69	,722	1221
Residuo	-3,764	2,515	,018	,827	1221
Valor pronosticado estándar	-3,497	2,046	-,019	,982	1221
Residuo estándar	-4,597	3,071	,022	1,010	1221

a. Variable dependiente: P8. Motivación proseguir aprendizaje 'coding'

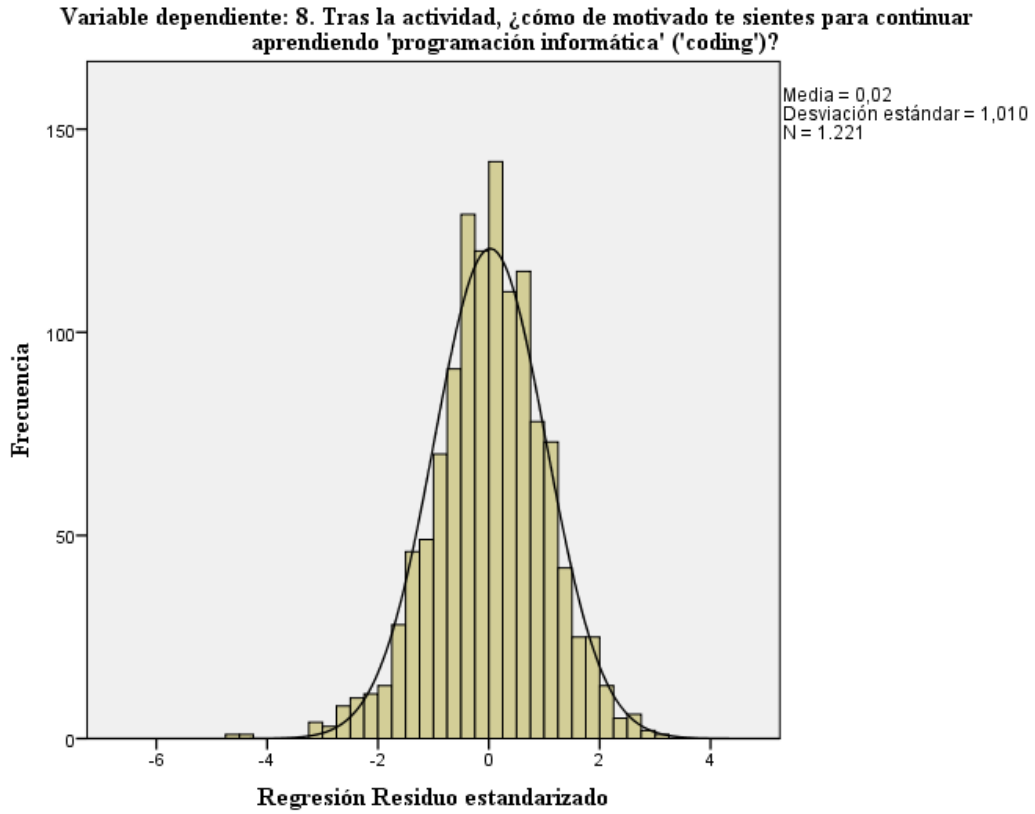


Figura D.31. Histograma de residuos tipificados; se comprueba normalidad

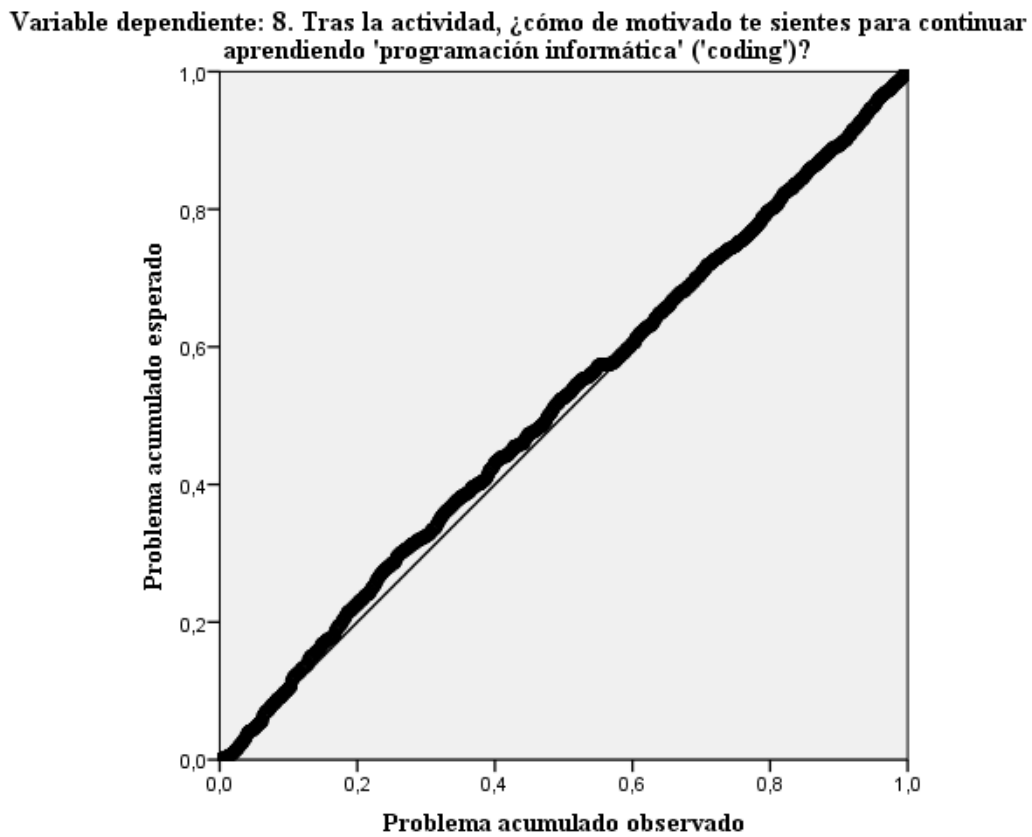


Figura D.32. Gráfico P-P de los residuos tipificados; se comprueba normalidad

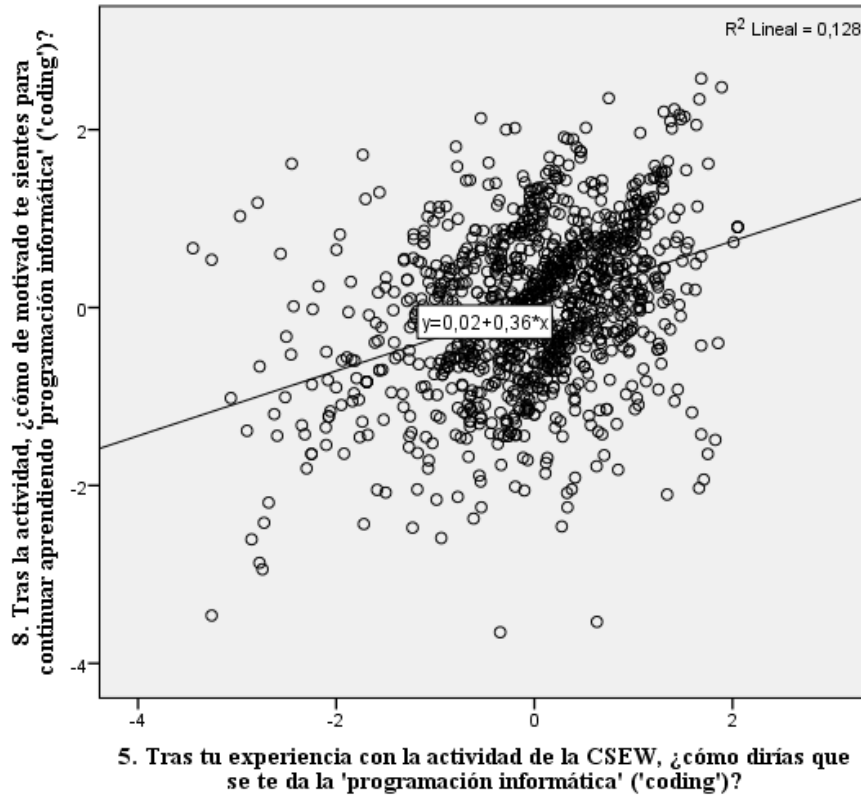


Figura D.33. Gráfica de regresión parcial P5 sobre P8

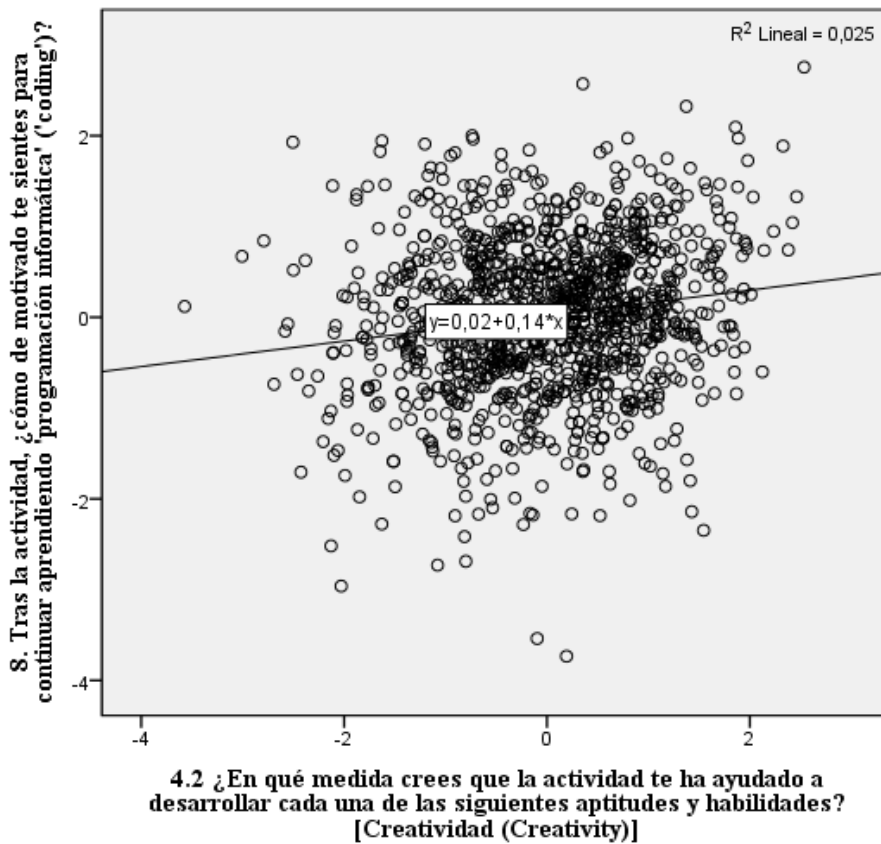


Figura D.34. Gráfica de regresión parcial P4.2 sobre P8

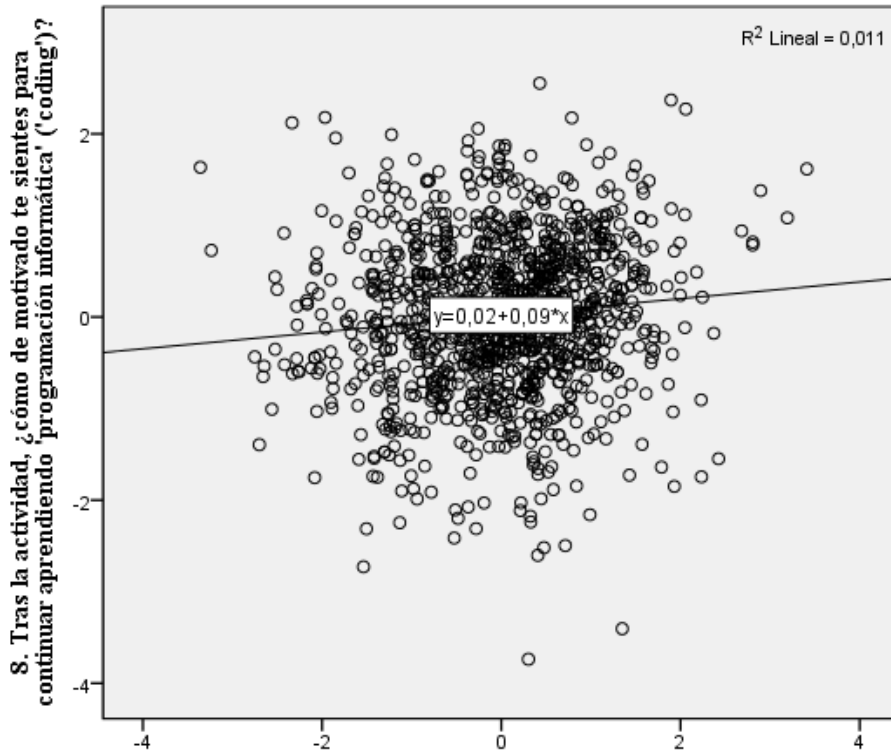


Figura D.35. Gráfica de regresión parcial P4.3 sobre P8

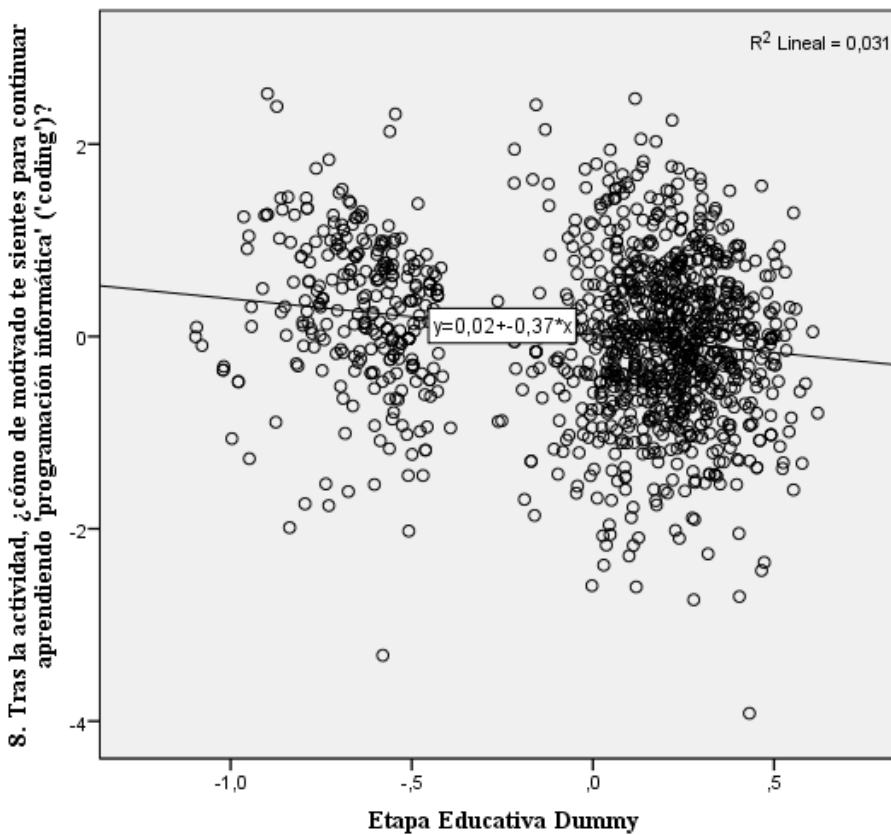


Figura D.36. Gráfica de regresión parcial 'Etapa Educativa' sobre P8

Tabla D.27. Estadísticas descriptivas de las variables independientes inicialmente incluidas en el análisis discriminante

	<i>P9. Determinación aprender 'coding' on-line</i>	Media	Desviación estándar	N válido (por lista)	
				No ponderados	Ponderados
No	<i>Sexo</i>	1,55	,499	196	196,000
	<i>Etapas Educativa</i>	2,02	,559	196	196,000
	<i>Titularidad</i>	1,73	,957	196	196,000
	<i>Tipología</i>	2,33	,579	196	196,000
	<i>P1. Haber oído 'coding'</i>	,43	,497	196	196,000
	<i>P2. Experiencia previa 'coding'</i>	,35	,479	196	196,000
	<i>P4.1 Desarrollar Pensamiento Lógico</i>	3,01	1,324	196	196,000
	<i>P4.2 Desarrollar Creatividad</i>	2,94	1,329	196	196,000
	<i>P4.3 Desarrollar Resolución de Problemas</i>	2,82	1,364	196	196,000
	<i>P4.4 Desarrollar Trabajo en Equipo</i>	2,44	1,503	196	196,000
	<i>P4.5 Desarrollar Persistencia-Perseverancia</i>	2,96	1,428	196	196,000
	<i>P4.6 Desarrollar Capacidad de Atención</i>	3,23	1,405	196	196,000
	<i>P5. Autoeficacia 'coding'</i>	3,18	1,170	196	196,000
	<i>P6. Completar actividad HoC</i>	,55	,499	196	196,000
	<i>P7.1 Rendimiento Matemáticas</i>	2,91	1,417	196	196,000
	<i>P7.2 Rendimiento Lengua</i>	3,05	1,401	196	196,000
	<i>P7.3 Rendimiento Música</i>	3,64	1,287	196	196,000
	<i>P7.4 Rendimiento Tecnología / Informática</i>	3,27	1,344	196	196,000
	<i>P7.5 Rendimiento Plástica</i>	3,73	1,274	196	196,000
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	2,52	1,139	196	196,000	
Sí	<i>Sexo</i>	1,46	,498	850	850,000
	<i>Etapas Educativa</i>	1,84	,630	850	850,000
	<i>Titularidad</i>	1,67	,937	850	850,000
	<i>Tipología</i>	2,20	,728	850	850,000
	<i>P1. Haber oído 'coding'</i>	,56	,497	850	850,000
	<i>P2. Experiencia previa 'coding'</i>	,46	,499	850	850,000
	<i>P4.1 Desarrollar Pensamiento Lógico</i>	3,83	1,015	850	850,000
	<i>P4.2 Desarrollar Creatividad</i>	3,77	1,098	850	850,000
	<i>P4.3 Desarrollar Resolución de Problemas</i>	3,64	1,165	850	850,000
	<i>P4.4 Desarrollar Trabajo en Equipo</i>	3,12	1,498	850	850,000
	<i>P4.5 Desarrollar Persistencia-Perseverancia</i>	3,57	1,191	850	850,000
	<i>P4.6 Desarrollar Capacidad de Atención</i>	3,92	1,134	850	850,000
	<i>P5. Autoeficacia 'coding'</i>	4,01	,829	850	850,000
	<i>P6. Completar actividad HoC</i>	,71	,455	850	850,000
	<i>P7.1 Rendimiento Matemáticas</i>	3,51	1,308	850	850,000
	<i>P7.2 Rendimiento Lengua</i>	3,31	1,235	850	850,000
	<i>P7.3 Rendimiento Música</i>	3,82	1,214	850	850,000
	<i>P7.4 Rendimiento Tecnología / Informática</i>	3,82	1,184	850	850,000
	<i>P7.5 Rendimiento Plástica</i>	3,90	1,157	850	850,000
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	3,98	,881	850	850,000	
Total	<i>Sexo</i>	1,47	,500	1046	1046,000
	<i>Etapas Educativa</i>	1,87	,621	1046	1046,000
	<i>Titularidad</i>	1,68	,940	1046	1046,000
	<i>Tipología</i>	2,23	,704	1046	1046,000

<i>P9. Determinación aprender 'coding' on-line</i>	Media	Desviación estándar	N válido (por lista)	
			No ponderados	Ponderados
<i>P1. Haber oído 'coding'</i>	,53	,499	1046	1046,000
<i>P2. Experiencia previa 'coding'</i>	,44	,496	1046	1046,000
<i>P4.1 Desarrollar Pensamiento Lógico</i>	3,68	1,126	1046	1046,000
<i>P4.2 Desarrollar Creatividad</i>	3,62	1,189	1046	1046,000
<i>P4.3 Desarrollar Resolución de Problemas</i>	3,49	1,246	1046	1046,000
<i>P4.4 Desarrollar Trabajo en Equipo</i>	3,00	1,522	1046	1046,000
<i>P4.5 Desarrollar Persistencia-Perseverancia</i>	3,45	1,261	1046	1046,000
<i>P4.6 Desarrollar Capacidad de Atención</i>	3,79	1,219	1046	1046,000
<i>P5. Autoeficacia 'coding'</i>	3,86	,959	1046	1046,000
<i>P6. Completar actividad HoC</i>	,68	,468	1046	1046,000
<i>P7.1 Rendimiento Matemáticas</i>	3,40	1,349	1046	1046,000
<i>P7.2 Rendimiento Lengua</i>	3,26	1,271	1046	1046,000
<i>P7.3 Rendimiento Música</i>	3,78	1,229	1046	1046,000
<i>P7.4 Rendimiento Tecnología / Informática</i>	3,71	1,233	1046	1046,000
<i>P7.5 Rendimiento Plástica</i>	3,87	1,181	1046	1046,000
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	3,70	1,095	1046	1046,000

Tabla D.28. Prueba de igualdad de medias entre los grupos de la variable dependiente

	Lambda de Wilks	F	df1	df2	Sig.
<i>Sexo</i>	,995	5,731*	1	1044	,017
<i>Etapa Educativa</i>	,988	12,503**	1	1044	,000
<i>Titularidad</i>	,999	,554	1	1044	,457
<i>Tipología</i>	,995	5,193*	1	1044	,023
<i>P1. Haber oído 'coding'</i>	,991	9,913**	1	1044	,002
<i>P2. Experiencia previa 'coding'</i>	,993	7,413**	1	1044	,007
<i>P4.1 Desarrollar Pensamiento Lógico</i>	,919	92,227**	1	1044	,000
<i>P4.2 Desarrollar Creatividad</i>	,926	83,035**	1	1044	,000
<i>P4.3 Desarrollar Resolución de Problemas</i>	,934	73,789**	1	1044	,000
<i>P4.4 Desarrollar Trabajo en Equipo</i>	,969	33,362**	1	1044	,000
<i>P4.5 Desarrollar Persistencia-Perseverancia</i>	,965	38,194**	1	1044	,000
<i>P4.6 Desarrollar Capacidad de Atención</i>	,951	53,301**	1	1044	,000
<i>P5. Autoeficacia 'coding'</i>	,885	136,252**	1	1044	,000
<i>P6. Completar actividad HoC</i>	,982	19,220**	1	1044	,000
<i>P7.1 Rendimiento Matemáticas</i>	,970	32,302**	1	1044	,000
<i>P7.2 Rendimiento Lengua</i>	,993	7,129**	1	1044	,008
<i>P7.3 Rendimiento Música</i>	,997	3,183	1	1044	,075
<i>P7.4 Rendimiento Tecnología / Informática</i>	,970	32,767**	1	1044	,000
<i>P7.5 Rendimiento Plástica</i>	,997	3,275	1	1044	,071
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	,728	390,095**	1	1044	,000

* Significativo al nivel $p_{(F)} < 0,05$ ** Significativo al nivel $p_{(F)} < 0,01$

Tabla D.29. Probabilidades previas para los grupos

<i>P9. Determinación aprender 'coding' on-line</i>	Previa	Casos utilizados en análisis	
		No ponderados	Ponderados
No	,187	196	196,000
Sí	,813	850	850,000
Total	1,000	1046	1046,000

Tabla D.30. Coeficientes de función de clasificación para los grupos

Coeficientes de función de clasificación		
	<i>P9. Determinación aprender 'coding' on-line</i>	
	No	Sí
<i>P2. Experiencia previa 'coding'</i>	1,119	1,531
<i>P4.1 Desarrollar Pensamiento Lógico</i>	1,770	1,981
<i>P5. Autoeficacia 'coding'</i>	2,950	3,280
<i>P8. Motivación proseguir aprendizaje 'coding'</i>	1,006	2,465
<i>(Constante)</i>	-10,487	-15,839

Funciones discriminantes lineales de Fisher

Tabla D.31. Tabla de contingencia ¿Caso anómalo? * Sexo

			Sexo		Total
			Chico	Chica	
¿Caso anómalo del análisis discriminante?	No	Recuento	788	687	1475
		% dentro de Caso anómalo del análisis discriminante	53,4%	46,6%	100,0%
	Sí	Recuento	58	74	132
		% dentro de Caso anómalo del análisis discriminante	43,9%	56,1%	100,0%
Total	Recuento	846	761	1607	
	% dentro de Caso anómalo del análisis discriminante	52,6%	47,4%	100,0%	

Tabla D.32. Tabla de contingencia ¿Caso anómalo? * Etapa Educativa

			Etapa Educativa				Total
			Primaria	Secundaria	Bachillerato	FP	
¿Caso anómalo del análisis discriminante?	No	Recuento	481	873	142	14	1510
		% dentro de Caso anómalo	31,9%	57,8%	9,4%	0,9%	100,0%
	Sí	Recuento	24	93	15	1	133
		% dentro de Caso anómalo	18,0%	69,9%	11,3%	0,8%	100,0%
Total	Recuento	505	966	157	15	1643	
	% dentro de Caso anómalo	30,7%	58,8%	9,6%	0,9%	100,0%	

Tabla D.33. Tabla de contingencia ¿Caso anómalo? * Clúster de pertenencia

			Número de Clúster				Total
			1	2	3	4	
¿Caso anómalo del análisis discriminante?	No	Recuento	93	53	485	588	1219
		% dentro de Caso anómalo	7,6%	4,3%	39,8%	48,2%	100,0%
	Sí	Recuento	42	91	0	0	133
		% dentro de Caso anómalo	31,6%	68,4%	0,0%	0,0%	100,0%
Total	Recuento	135	144	485	588	1352	
	% dentro de Caso anómalo	10,0%	10,7%	35,9%	43,5%	100,0%	

Tabla D.34. Tabla de contingencia 'Coders vs. Scratchers' * sexo

			Sexo		Total
			Chico	Chica	
Coders vs. Scratchers	Coders	Recuento	195	199	394
		% dentro de Coders vs. Scratchers	49,5%	50,5%	100,0%
	Scratchers	Recuento	128	100	228
		% dentro de Coders vs. Scratchers	56,1%	43,9%	100,0%
Total	Recuento	323	299	622	
	% dentro de Coders vs. Scratchers	51,9%	48,1%	100,0%	

Tabla D.35. Tabla de contingencia 'Coders vs. Scratchers' * etapa educativa

			Nivel Educativo				Total
			Primaria	Secundaria	Bachillerat.	FP	
Coders vs. Scratchers	Coders	Recuento	144	209	43	4	400
		% dentro de Coders vs. Scratch.	36,0%	52,3%	10,8%	1,0%	100,0%
	Scratch.	Recuento	34	164	29	2	229
		% dentro de Coders vs. Scratch.	14,8%	71,6%	12,7%	0,9%	100,0%
Total	Recuento	178	373	72	6	629	
	% dentro de Coders vs. Scratch.	28,3%	59,3%	11,4%	1,0%	100,0%	

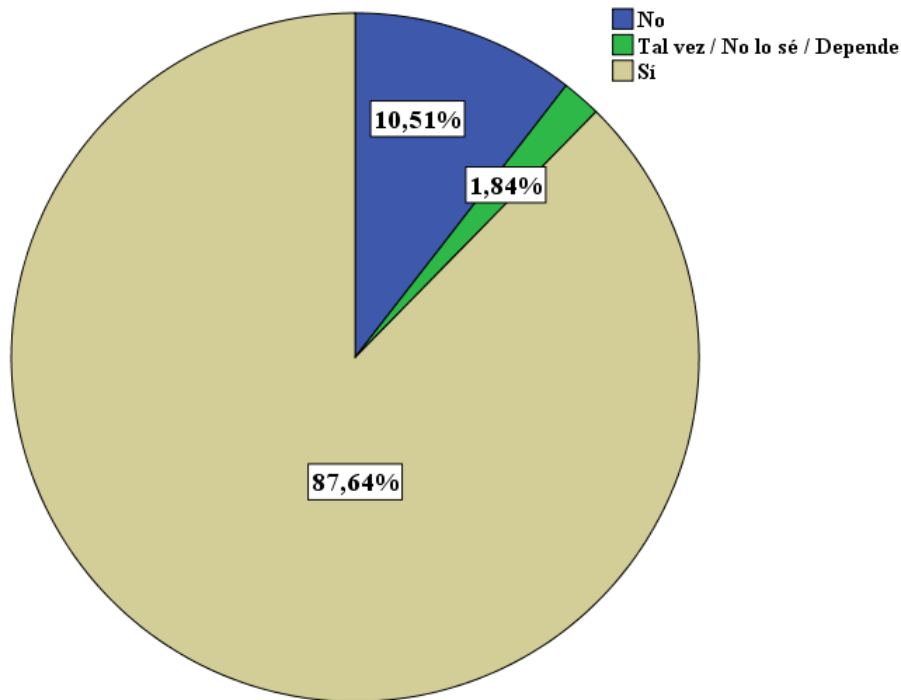


Figura D.37. Distribución de frecuencias de respuesta a la Pregunta 11 (relevancia del 'coding' para la formación personal)

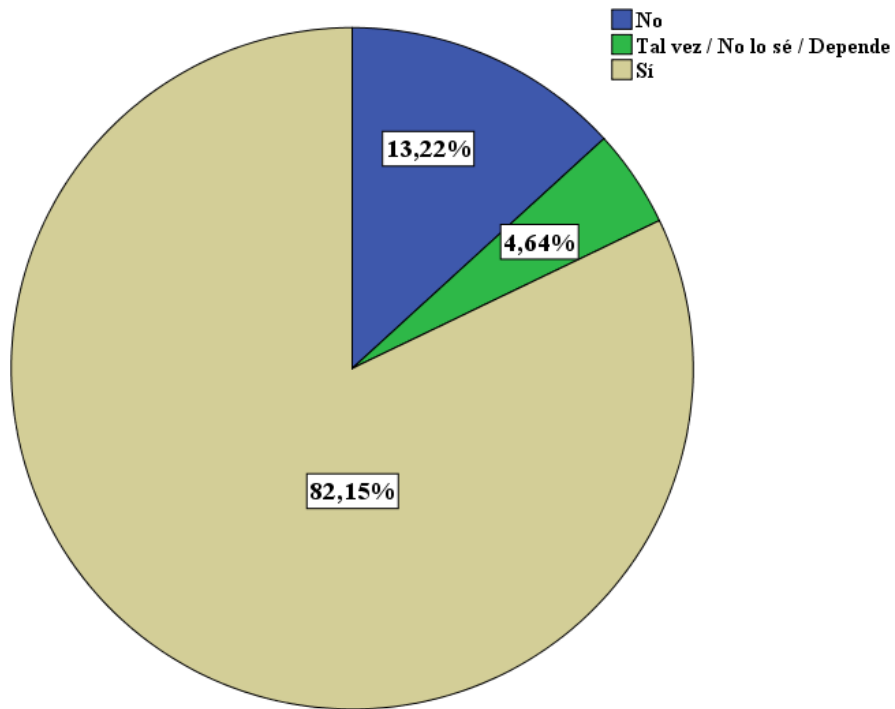


Figura D.38. Distribución de frecuencias de respuesta a la Pregunta 12 (relevancia del 'coding' para la formación profesional)

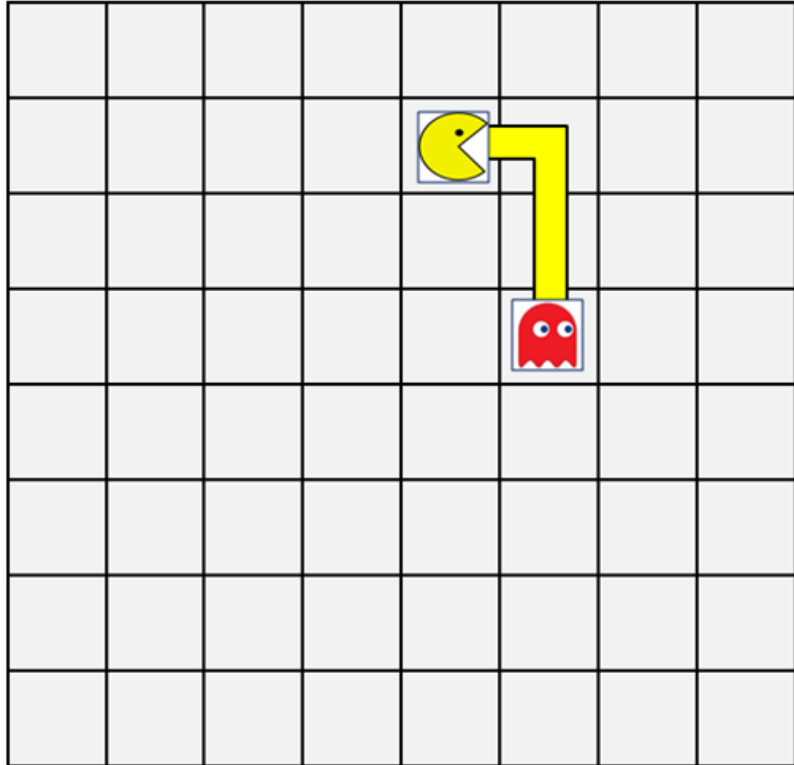
ANEXO E


Test de Pensamiento Computacional (*versión 1.0*)



	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	Opción correcta
			Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Item 1	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Secuenciación	B
Item 2	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Completamiento	C
Item 3	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Depuración	A
Item 4	Laberinto	Textual	Sí	No	No	No	No	No	No	No	No	Secuenciación	A
Item 5	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Secuenciación	D
Item 6	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Secuenciación	C
Item 7	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Completamiento	D
Item 8	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación	B
Item 9	Lienzo	Textual	Sí	Sí	No	No	No	No	No	No	No	Depuración	A
Item 10	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Completamiento	B
Item 11	Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No	No	Secuenciación	D
Item 12	Laberinto	Visual por bloques	Sí	No	Sí	No	No	No	No	No	No	Completamiento	C
Item 13	Laberinto	Textual	Sí	No	Sí	No	No	No	No	No	No	Secuenciación	A
Item 14	Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	Sí	Depuración	B
Item 15	Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	Sí	Secuenciación	C
Item 16	Laberinto	Visual por flechas	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación	D
Item 17	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación	A
Item 18	Laberinto	Textual	Sí	No	Sí	Sí	No	No	No	No	Sí	Completamiento	C
Item 19	Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	Sí	Completamiento	D
Item 20	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Depuración	C
Item 21	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación	B
Item 22	Laberinto	Textual	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación	B
Item 23	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Depuración	A
Item 24	Laberinto	Textual	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento	D
Item 25	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento	C
Item 26	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación	A
Item 27	Laberinto	Textual	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación	B
Item 28	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Depuración	D
Item 29	Laberinto	Textual	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento	A
Item 30	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento	C
Item 31	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación	B
Item 32	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento	B
Item 33	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Depuración	D
Item 34	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación	A
Item 35	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento	C
Item 36	Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Secuenciación	D
Item 37	Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Completamiento	A
Item 38	Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Depuración	D
Item 39	Lienzo	Visual por bloques	No	No	No	No	No	No	No	Sí	Sí	Secuenciación	B
Item 40	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	Sí	Sí	Completamiento	C


TPC (Versión 1.0) - ÍTEM 1


¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?



Opción A


Opción B
 

Opción C


Opción D


Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 1.0) - ÍTEM 2

¿Qué orden falta en la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Opción B

Opción C
 ✓

Opción D

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Completamiento

TPC (Versión 1.0) - ÍTEM 3

Para llevar a 'Pac-Man' hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

Opción A ✓

- avanzar
- avanzar
- avanzar
- girar a la izquierda ↻
- avanzar
- avanzar
- girar a la derecha ↻
- avanzar
- avanzar

Opción B

- avanzar
- avanzar
- avanzar
- girar a la izquierda ↻
- avanzar
- avanzar
- girar a la derecha ↻
- avanzar
- avanzar

Opción C

- avanzar
- avanzar
- avanzar
- girar a la izquierda ↻
- avanzar
- avanzar
- girar a la derecha ↻
- avanzar
- avanzar

Opción D

- avanzar
- avanzar
- avanzar
- girar a la izquierda ↻
- avanzar
- avanzar
- girar a la derecha ↻
- avanzar
- avanzar

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Depuración

TPC (Versión 1.0) - ÍTEM 4

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Avanzar 1 vez + Girar a la derecha + Avanzar 2 veces + Girar a la izquierda + Avanzar 1 vez

Opción B

Avanzar 1 vez + Girar a la derecha + Avanzar 2 veces + Girar a la derecha + Avanzar 1 vez

Opción C

Avanzar 1 vez + Girar a la derecha + Avanzar 2 veces + Girar a la izquierda

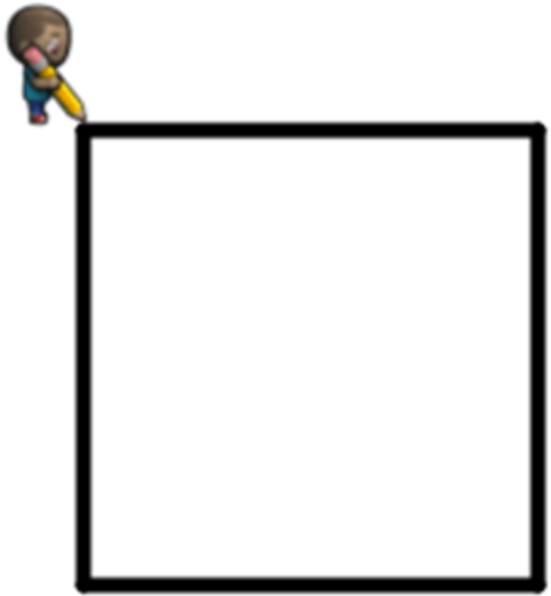
Opción D

Avanzar 1 vez + Girar a la izquierda + Avanzar 2 veces + Girar a la derecha + Avanzar 1 vez

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Textual	Sí	No	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 1.0) - ÍTEM 5

¿Qué ordenes debe ejecutar el artista para dibujar el cuadrado? Cada uno de los lados del cuadrado mide 100 píxeles.



Opción A

- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles
- girar a la izquierda por 90 grados
- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles

Opción B

- mover hacia adelante 25 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 25 píxeles
- girar a la izquierda por 90 grados
- mover hacia adelante 25 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 25 píxeles

Opción C

- mover hacia adelante 50 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 50 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 50 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 50 píxeles

Opción D ✔

- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 1.0) - ÍTEM 6

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

× 5

Opción B

× 3

Opción C

× 4

✓

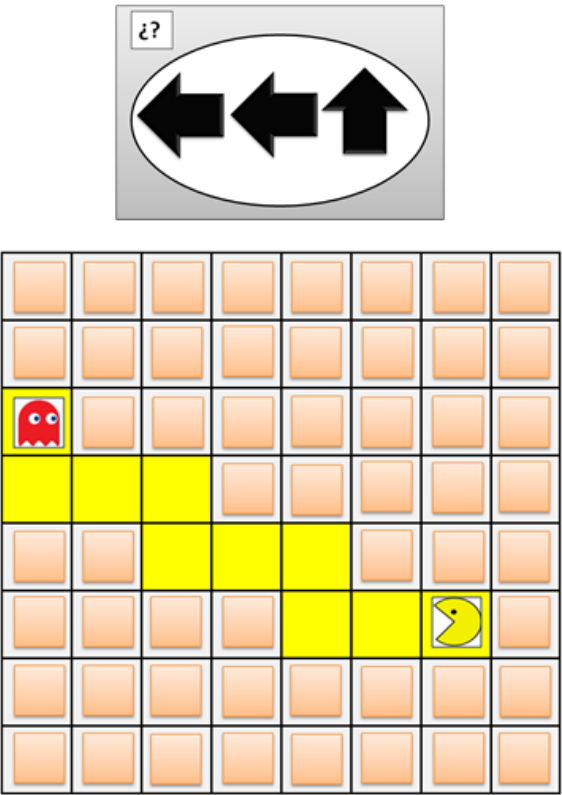
Opción D

× 2

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 1.0) - ÍTEM 7

¿Cuántas veces se debe repetir la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?




Opción A
× 2

Opción B
× 1

Opción C
× 4

Opción D
× 3



Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Completamiento

TPC (Versión 1.0) - ÍTEM 8

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

```

repetir 4 veces
  haz
    repetir 3 veces
      haz avanzar
    girar a la derecha
  avanzar
            
```

Opción B ✓

```

repetir 3 veces
  haz
    repetir 4 veces
      haz avanzar
    girar a la derecha
  avanzar
            
```

Opción C

```

repetir 3 veces
  haz
    repetir 4 veces
      haz
        avanzar
        girar a la derecha
  avanzar
            
```

Opción D

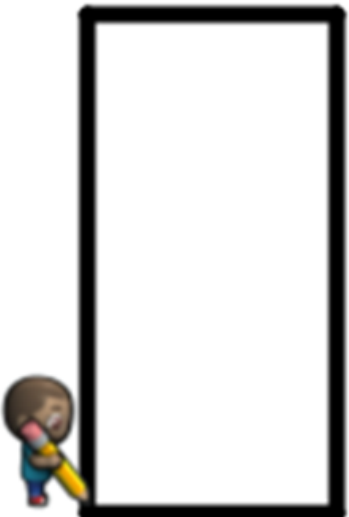
```

repetir 4 veces
  haz avanzar
repetir 3 veces
  haz
    girar a la derecha
  avanzar
            
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 9

Para que el artista dibuje una vez el siguiente rectángulo (50 píxeles de ancho y 100 píxeles de alto), ¿en qué paso de la siguiente secuencia de órdenes hay un error?



Opción A ✔

4 × [Avanzar 50 px. + Girar a la izquierda + Avanzar 100 px. + Girar a la izquierda]

Opción B

4 × [Avanzar 50 px. + Girar a la izquierda + Avanzar 100 px. + Girar a la izquierda]

Opción C

4 × [Avanzar 50 px. + Girar a la izquierda + Avanzar 100 px. + Girar a la izquierda]

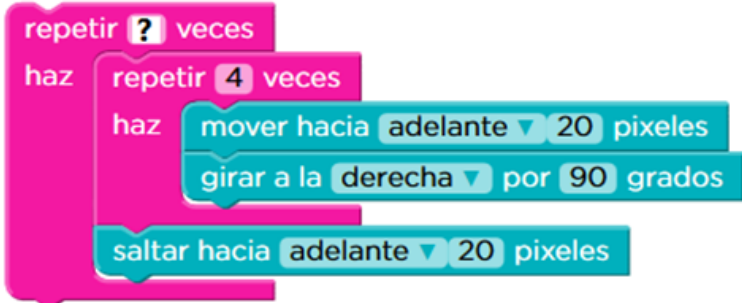

Opción D

4 × [Avanzar 50 px. + Girar a la izquierda + Avanzar 100 px. + Girar a la izquierda]

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Textual	Sí	Sí	No	No	No	No	No	No	No	Depuración

TPC (Versión 1.0) - ÍTEM 10

¿Qué número falta en la siguiente secuencia de órdenes para que el artista ejecute la figura? El ancho total de la figura es de 120 píxeles.

Opción A: 4

Opción B: 6

Opción C: 2

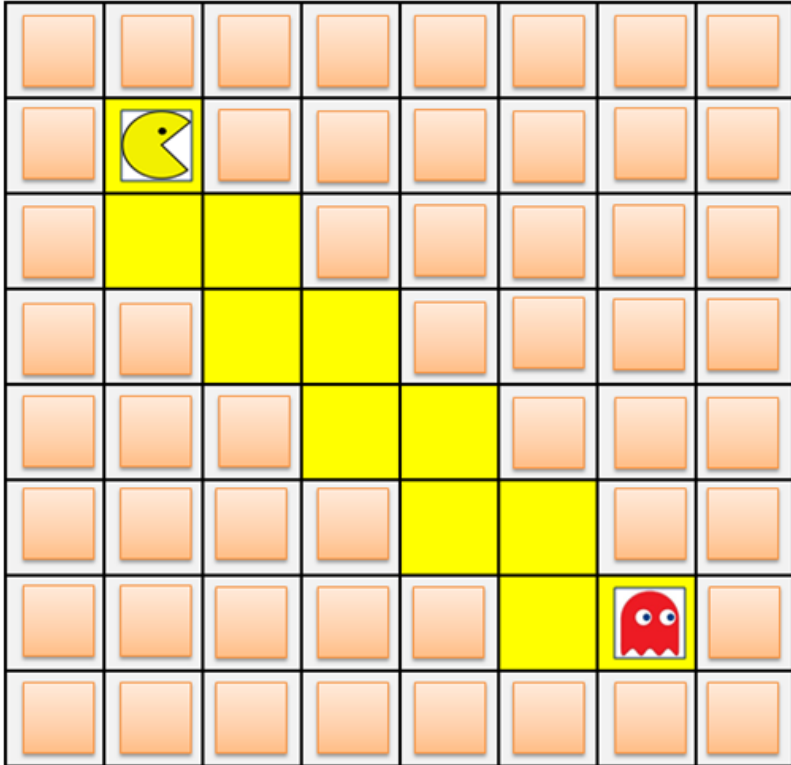
Opción D: 5

✔


Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Completamiento


TPC (Versión 1.0) - ÍTEM 11

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?





Opción A

Repetir hasta llegar a... 





Opción B


Repetir hasta llegar a... 





Opción C

Repetir hasta llegar a... 



Opción D 


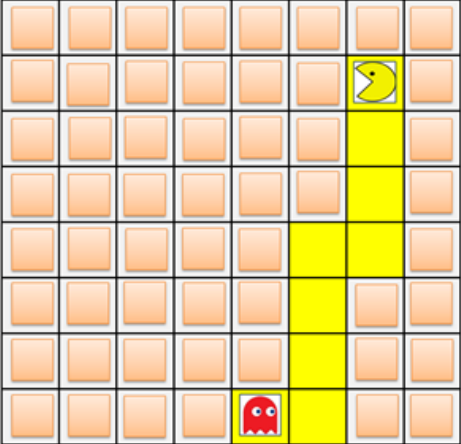
Repetir hasta llegar a... 



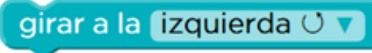
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No	No	Secuenciación

TPC (Versión 1.0) - ÍTEM 12

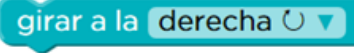
¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?





Opción A




Opción B





Opción C



Opción D

No falta ningún bloque

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por bloques	Sí	No	Sí	No	No	No	No	No	Completamiento	

TPC (Versión 1.0) - ÍTEM 13

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Repetir hasta llegar al fantasma[Avanzar 5 veces + Girar a la izquierda]

Opción B

Repetir hasta llegar al fantasma[Avanzar 5 veces] + Girar a la izquierda

Opción C

Repetir hasta llegar al fantasma[Avanzar 1 vez + Girar a la izquierda]

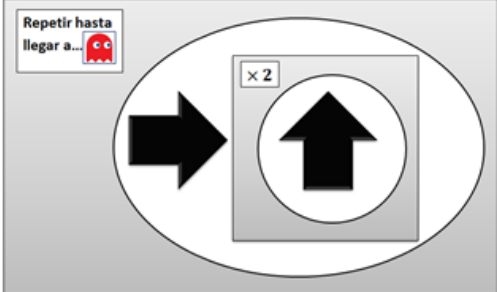
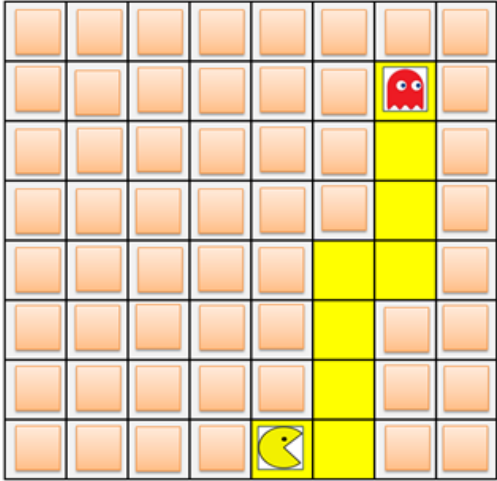
Opción D

Repetir hasta llegar al fantasma[Avanzar 1 vez] + Girar a la izquierda

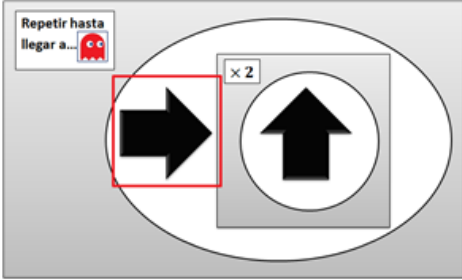
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Textual	Sí	No	Sí	No	No	No	No	No	No	Secuenciación

TPC (Versión 1.0) - ÍTEM 14

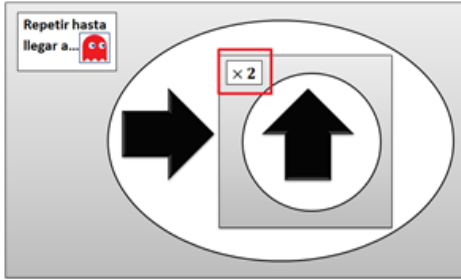
Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

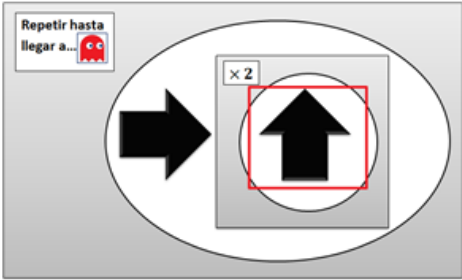
Opción A



Opción B



Opción C




Opción D

No hay ningún error en la secuencia

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	Sí	Depuración

TPC (Versión 1.0) - ÍTEM 15

¿Qué secuencia de órdenes debe ejecutar el artista para dibujar la escalera que llegue hasta la flor? Cada peldaño sube 30 píxeles



Opción A

```

Repetir hasta la flor
haz
  repetir 7 veces
  haz
    mover hacia adelante 30 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 30 píxeles
            
```

Opción B

```

Repetir hasta la flor
haz
  repetir 4 veces
  haz
    mover hacia adelante 120 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 30 píxeles
            
```

Opción C ✓

```

Repetir hasta la flor
haz
  repetir 4 veces
  haz
    mover hacia adelante 30 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 30 píxeles
            
```

Opción D

```

Repetir hasta la flor
haz
  repetir 4 veces
  haz
    mover hacia adelante 30 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 210 píxeles
            
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 16

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Opción B

Opción C

Opción D

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 17

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A ✓

```

Repetir hasta llegar a... (ghost)
  hacer
    avanzar
    si hay camino a la derecha
      hacer girar a la derecha
        
```

Opción B

```

Repetir hasta llegar a... (ghost)
  hacer
    girar a la derecha
    si hay camino a la derecha
      hacer avanzar
        
```

Opción C

```

Repetir hasta llegar a... (ghost)
  hacer
    avanzar
    si hay camino a la derecha
      hacer girar a la izquierda
        
```

Opción D

```

Repetir hasta llegar a... (ghost)
  hacer
    avanzar
    si hay camino a la izquierda
      hacer girar a la izquierda
        
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación


TPC (Versión 1.0) - ÍTEM 18

¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?

Repetir hasta llegar al fantasma [Avanzar+¿? (Girar a la izquierda)]

Opción A
Si hay camino de frente

Opción B
Si hay camino a la derecha

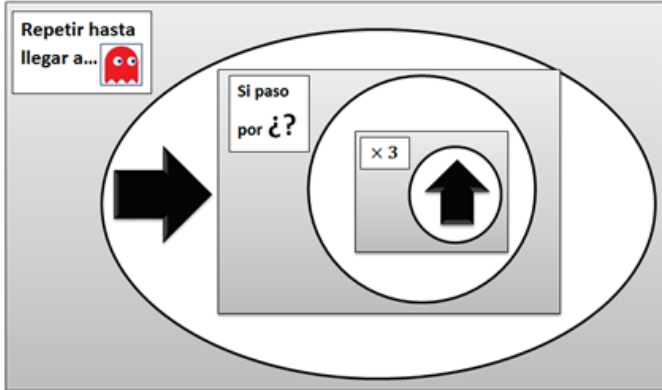
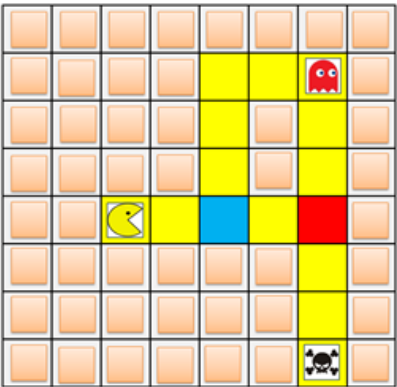
Opción C 
Si hay camino a la izquierda


Opción D
Si no hay camino de frente

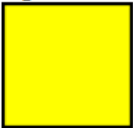
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Textual	Sí	No	Sí	Sí	No	No	No	No	Sí	Completamiento

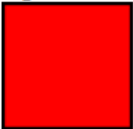
TPC (Versión 1.0) - ÍTEM 19


¿Qué falta en la siguiente secuencia de órdenes para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A


Opción B


Opción C


Opción D
Tanto la opción A como la opción C son correctas 

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 20

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

Opción A

```

Repetir hasta llegar a... [ghost]
hacer
  avanzar
  si hay camino a la izquierda
  hacer girar a la izquierda
  si hay camino a la derecha
  hacer avanzar
                
```

Opción B

```

Repetir hasta llegar a... [ghost]
hacer
  avanzar
  si hay camino a la izquierda
  hacer girar a la izquierda
  si hay camino a la derecha
  hacer avanzar
                
```

Opción C ✓

```

Repetir hasta llegar a... [ghost]
hacer
  avanzar
  si hay camino a la izquierda
  hacer girar a la izquierda
  si hay camino a la derecha
  hacer avanzar
                
```

Opción D
No hay ningún error

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Depuración

TPC (Versión 1.0) - ÍTEM 21

¿Qué órdenes llevan a Pac-Man hasta el fantasma por el camino señalado?

Opción A

```

Repetir hasta llegar a...
hacer
  si hay un camino delante
  hacer avanzar
  sino girar a la izquierda
        
```

Opción B ✓

```

Repetir hasta llegar a...
hacer
  si hay un camino delante
  hacer avanzar
  sino girar a la derecha
        
```

Opción C

```

Repetir hasta llegar a...
hacer
  si hay camino a la derecha
  hacer girar a la derecha
  sino avanzar
        
```

Opción D

```

Repetir hasta llegar a...
hacer
  si hay camino a la izquierda
  hacer girar a la izquierda
  sino avanzar
        
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 22

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A
Repetir hasta llegar al fantasma {Si hay camino delante(Avanzar) | Si no(Girar a la derecha)}

Opción B ✓
Repetir hasta llegar al fantasma {Si hay camino delante(Avanzar) | Si no(Girar a la izquierda)}

Opción C
Repetir hasta llegar al fantasma {Si hay camino a la derecha(Girar a la derecha) | Si no(Avanzar)}

Opción D
Repetir hasta llegar al fantasma {Si hay camino a la izquierda(Girar a la izquierda) | Si no(Avanzar)}

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Textual	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 23

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

Opción A ✔

Opción B

Opción C

Opción D
No hay ningún error

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Depuración

TPC (Versión 1.0) - ÍTEM 24


¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?

Repetir hasta llegar al fantasma {Si hay camino adelante(Avanzar) | Si no {Si hay camino a la izquierda(Girar a la izquierda) | Si no(?)}}

Opción A
Avanzar

Opción B
Retroceder

Opción C
Girar a la izquierda

Opción D 
Girar a la derecha

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Textual	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 25

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?

Opción A

Opción B

Opción C

Opción D

No falta ningún bloque

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de amidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 26

¿Qué órdenes llevan a 'Pac-Man' por el camino señalado hasta las fresas e indican a 'Pac-Man' que se coma el número de fresas indicado?

Opción A ✔

```

mientras haya camino delante
  hacer avanzar
repetir 3 veces
  haz Comer 1 fresa
            
```

Opción B

```

mientras haya camino delante
  hacer avanzar
repetir 4 veces
  haz Comer 1 fresa
            
```

Opción C

```

mientras haya camino delante
  hacer avanzar
repetir 5 veces
  haz Comer 1 fresa
            
```

Opción D

```

mientras haya camino delante
  hacer avanzar
    repetir 3 veces
      haz Comer 1 fresa
            
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 27

¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?

□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□
□	□	□	□	□	□	□	□

Opción A
Mientras haya camino delante {Avanzar 5 veces + [3 × (Comer 1 fresa)]}

Opción B
Mientras haya camino delante {Avanzar 1 vez + [3 × (Comer 1 fresa)]}

Opción C
Mientras haya camino delante {Avanzar 3 veces + [5 × (Comer 1 fresa)]}

Opción D
Mientras haya camino delante {Avanzar 1 vez} + [3 × (Comer 1 fresa)]

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Textual	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 28

Para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas, ¿en qué paso de la siguiente secuencia de órdenes hay un error?

Opción A

```

mientras haya camino delante
  hacer
    avanzar
    repetir 3 veces
      haz Comer 1 fresa
  girar a la izquierda
  mientras haya camino delante
    hacer
      avanzar
      repetir 4 veces
        haz Comer 1 fresa
        
```

Opción B

```

mientras haya camino delante
  hacer
    avanzar
    repetir 3 veces
      haz Comer 1 fresa
  girar a la izquierda
  mientras haya camino delante
    hacer
      avanzar
      repetir 4 veces
        haz Comer 1 fresa
        
```

Opción C

```

mientras haya camino delante
  hacer
    avanzar
    repetir 3 veces
      haz Comer 1 fresa
  girar a la izquierda
  mientras haya camino delante
    hacer
      avanzar
      repetir 4 veces
        haz Comer 1 fresa
        
```

Opción D ✓

```


mientras haya camino delante
  hacer
    avanzar
    repetir 3 veces
      haz Comer 1 fresa
  girar a la izquierda
  mientras haya camino delante
    hacer
      avanzar
      repetir 4 veces
        haz Comer 1 fresa
        
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Depuración

TPC (Versión 1.0) - ÍTEM 29

¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas?

Mientras haya camino delante {? + [Si hay una fresa(Comer 1 fresa)]}

Opción A 

Avanzar 1 vez

Opción B

Avanzar 2 veces

Opción C

Avanzar 3 veces

Opción D

Avanzar 5 veces

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Textual	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 30

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas?

Opción A

Mientras haya camino delante

Opción B

Mientras no haya camino delante

Opción C ✔

Mientras haya alguna fresa

Opción D

Mientras no haya ninguna fresa

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 31

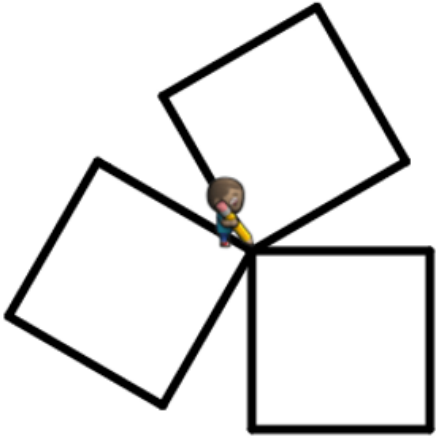
Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function':

Función

```

my function
  repetir 4 veces
  haz mover hacia adelante 100 pixeles
  girar a la derecha por 90 grados
        
```

¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? Cada uno de los lados de cada cuadrado mide 100 píxeles.



Opción A

```

repetir 3 veces
  haz my function
  girar a la derecha por 120 grados
        
```

Opción B

```

repetir 3 veces
  haz my function
  girar a la derecha por 120 grados
        
```

Opción C

```

repetir 4 veces
  haz my function
  girar a la derecha por 90 grados
        
```

Opción D

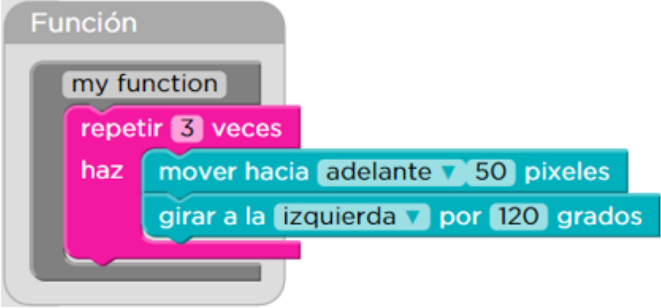
```

repetir 4 veces
  haz my function
  girar a la derecha por 90 grados
        
```


Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 32

Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function':



¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? Cada uno de los lados de cada triángulo mide 50 píxeles.



Opción A 15	Opción B 5
Opción C 4	Opción D 3

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 33

Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function':

Función

```

my function
  repetir 360 veces
  haz
    mover hacia adelante 1 pixeles
    girar a la derecha por 1 grados
    
```

Para que el artista ejecute el siguiente diseño ¿En qué paso de la secuencia de órdenes hay un error?

Opción A

```

repetir 4 veces
  haz
    my function
    girar a la derecha por 90 grados
    
```

Opción B

```

repetir 4 veces
  haz
    my function
    girar a la derecha por 90 grados
    
```

Opción C

```

repetir 4 veces
  haz
    my function
    girar a la derecha por 90 grados
    
```

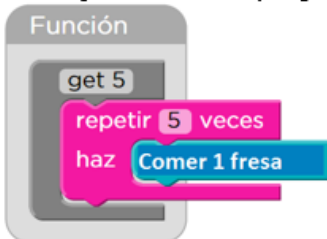
Opción D

No hay ningún error en la secuencia

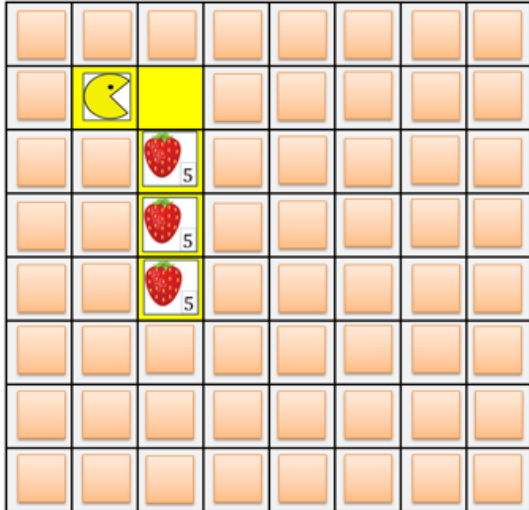
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de amidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Depuración

TPC (Versión 1.0) - ÍTEM 34

Si tenemos el siguiente conjunto de órdenes, al que llamamos 'get 5':



¿Qué órdenes llevan a 'Pac-Man' por el camino señalado hasta las fresas e indican a 'Pac-Man' que se coma el número de fresas indicado?



Opción A ✓

```

avanzar
girar a la derecha
repetir 3 veces
  haz avanzar
  get 5
    
```

Opción B

```

avanzar
girar a la derecha
repetir 3 veces
  haz get 5
avanzar
    
```

Opción C

```

avanzar
girar a la derecha
repetir 5 veces
  haz avanzar
  get 5
    
```

Opción D


```

avanzar
girar a la derecha
repetir 5 veces
  haz get 5
avanzar
    
```

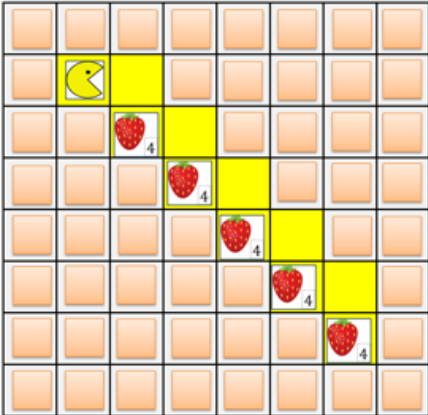
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 35

Si tenemos el siguiente conjunto de órdenes, llamado 'move and get 4':



¿Qué falta en la siguiente secuencia para llevar a 'Pac-Man' por el camino señalado hasta las fresas, comiendo el número de fresas indicado?



repetir ??? veces
haz move and get 4

Opción A

3

Opción B

4

Opción C

5

Opción D

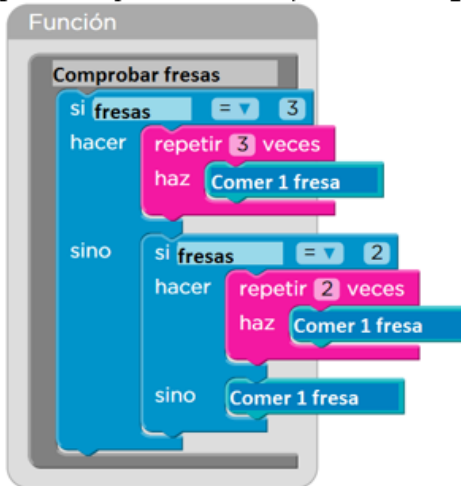
6

✓

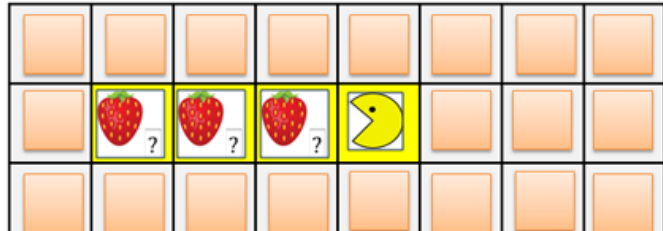
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 36


Si tenemos el siguiente conjunto de órdenes, llamado 'comprobar fresas':




¿Qué órdenes llevan a 'Pac-Man' por el camino señalado y hacen que 'Pac-Man' se coma el número de fresas indicado? **En las casillas con fresas puede haber 1, 2 ó 3 fresas.**




Opción A




Opción B



Opción C



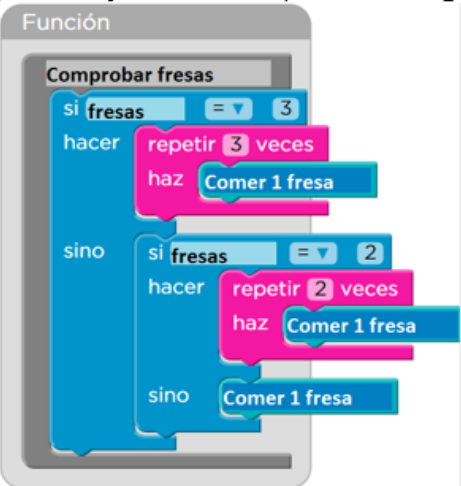
Opción D




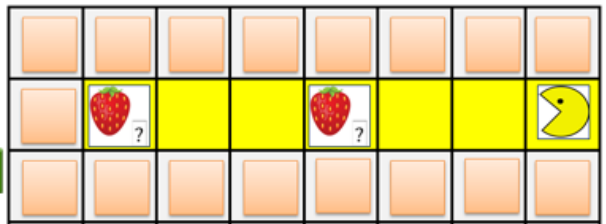
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 37

Si tenemos el siguiente conjunto de órdenes, llamado 'comprobar fresas':



¿Qué falta en la siguiente secuencia para llevar a 'Pac-Man' por el camino señalado comiendo el número de fresas indicado? En las casillas con fresas puede haber 1, 2 ó 3 fresas.

Opción A ✓

Opción B

2

3

Opción C

4

Opción D

5

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Completamiento

TPC (Versión 1.0) - ÍTEM 38

Para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas, ¿en qué paso del conjunto de órdenes 'comprobar fresas' hay un error? **En las casillas con fresas puede haber 1 ó 2 fresas**

Opción A

Opción B

Opción C

Opción D

No hay ningún error en la secuencia

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	Sí	No	No	Sí	Sí	Depuración

TPC (Versión 1.0) - ÍTEM 39

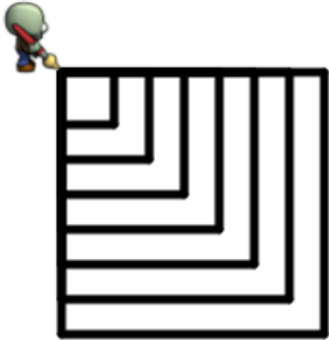
Si tenemos la siguiente orden, que dibuja un cuadrado de longitud-lado variable en función de un 'contador':

dibujar un cuadrado longitud: contador

Y el 'contador' cuenta de un número inicial a otro número final, pasando por un incremento especificado

contar con contador de a por

¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? El lado del cuadrado más pequeño es 30 píxeles, y el lado del cuadrado más grande es 150 píxeles.



Opción A

contar con contador de 30 a 150 por 10

dibujar un cuadrado longitud: contador

Opción B

contar con contador de 30 a 150 por 20

dibujar un cuadrado longitud: contador

Opción C

contar con contador de 0 a 150 por 10

dibujar un cuadrado longitud: contador

Opción D

contar con contador de 0 a 150 por 20

dibujar un cuadrado longitud: contador

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Lienzo	Visual por bloques	No	No	No	No	No	No	No	No	Sí	Sí	Secuenciación

TPC (Versión 1.0) - ÍTEM 40

¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? El lado más corto mide 20 píxeles y el lado más largo mide 200 píxeles

Opción A
10

Opción B
20

Opción C
30 ✓

Opción D
40

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	Sí	Completamiento	

ANEXO F

Test de Pensamiento Computacional (*versión 2.0*)

	Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	Opción correcta
			Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
				Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Item 1	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Secuenciación	B
Item 2	Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Completamiento	C
Item 3	Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Depuración	D
Item 4	Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Secuenciación	D
Item 5	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Secuenciación	C
Item 6	Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Completamiento	D
Item 7	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	No	Depuración	A
Item 8	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación	B
Item 9	Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No	No	Secuenciación	D
Item 10	Laberinto	Visual por bloques	Sí	No	Sí	No	No	No	No	No	No	Completamiento	C
Item 11	Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	Sí	Depuración	C
Item 12	Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	Sí	Secuenciación	A
Item 13	Laberinto	Visual por flechas	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación	B
Item 14	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación	A
Item 15	Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	Sí	Completamiento	D
Item 16	Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Depuración	D
Item 17	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación	B
Item 18	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación	A
Item 19	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Depuración	B
Item 20	Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento	C
Item 21	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación	A
Item 22	Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación	B
Item 23	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento	A
Item 24	Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento	C
Item 25	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación	B
Item 26	Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento	B
Item 27	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación	A
Item 28	Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Completamiento	C

TPC (Versión 2.0) - ÍTEM 1

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Opción B

Opción C

Opción D

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 2.0) - ÍTEM 2

¿Qué orden falta en la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Opción B

Opción C
 ✓

Opción D

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por flechas	Sí	No	No	No	No	No	No	No	No	Completamiento

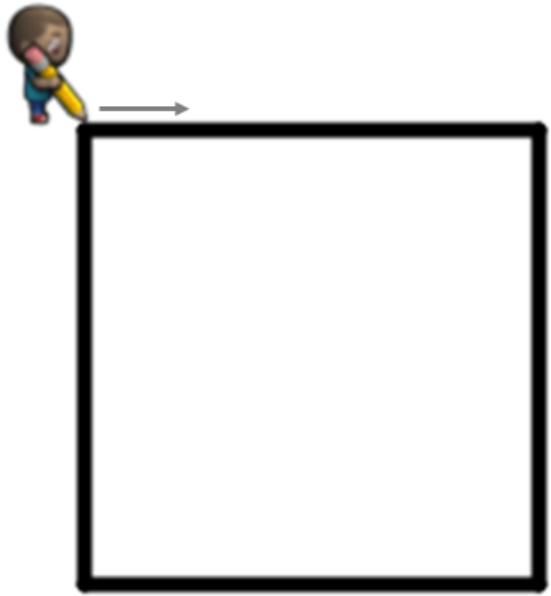
TPC (Versión 2.0) - ÍTEM 3

Para llevar a 'Pac-Man' hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Depuración

TPC (Versión 2.0) - ÍTEM 4

¿Qué órdenes debe ejecutar el artista para dibujar el cuadrado? Cada uno de los lados del cuadrado mide 100 píxeles.



Opción A

- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles
- girar a la izquierda por 90 grados
- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles

Opción B

- mover hacia adelante 25 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 25 píxeles
- girar a la izquierda por 90 grados
- mover hacia adelante 25 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 25 píxeles

Opción C

- mover hacia adelante 50 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 50 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 50 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 50 píxeles

Opción D ✔

- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles
- girar a la derecha por 90 grados
- mover hacia adelante 100 píxeles

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Visual por bloques	Sí	No	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 2.0) - ÍTEM 5

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

× 5

Opción B

× 3

Opción C

× 4

✓

Opción D

× 2

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	No	Secuenciación

TPC (Versión 2.0) - ÍTEM 6

¿Cuántas veces se debe repetir la secuencia para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A
× 2

Opción B
× 1

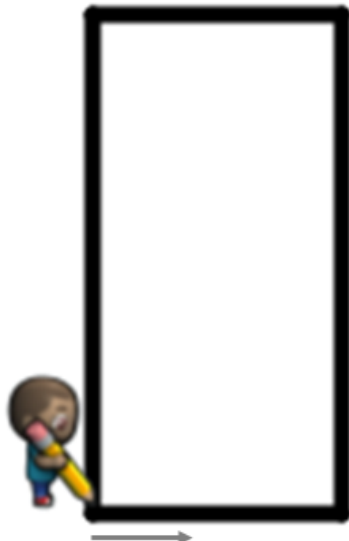
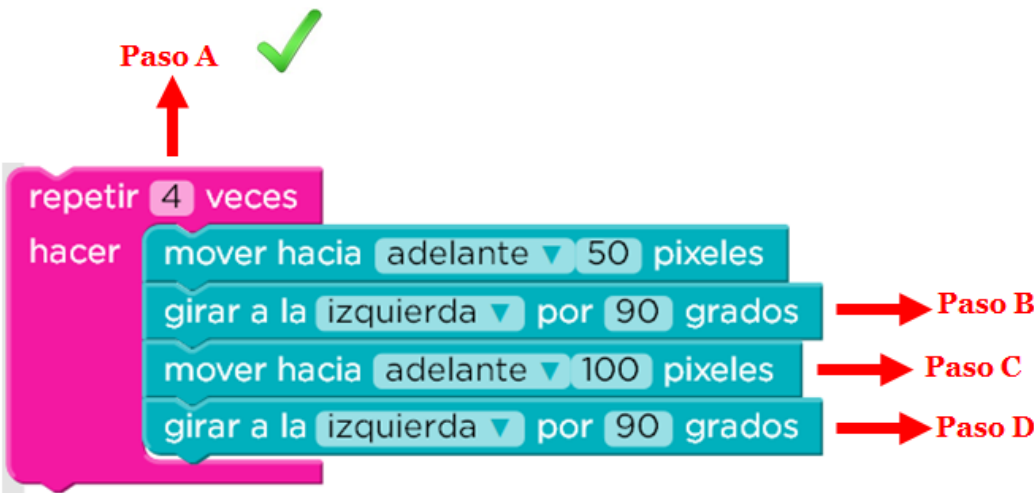
Opción C
× 4

Opción D
× 3

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por flechas	Sí	Sí	No	No	No	No	No	No	No	Completamiento

TPC (Versión 2.0) - ÍTEM 7

Para que el artista dibuje **una vez** el siguiente rectángulo (50 píxeles de ancho y 100 píxeles de alto), ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	No	No	Depuración

TPC (Versión 2.0) - ÍTEM 8

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

```

repetir 4 veces
  haz
    repetir 3 veces
      haz avanzar
    girar a la derecha
  avanzar
            
```

Opción B ✓

```

repetir 3 veces
  haz
    repetir 4 veces
      haz avanzar
    girar a la derecha
  avanzar
            
```

Opción C

```

repetir 3 veces
  haz
    repetir 4 veces
      haz
        avanzar
        girar a la derecha
    avanzar
            
```

Opción D

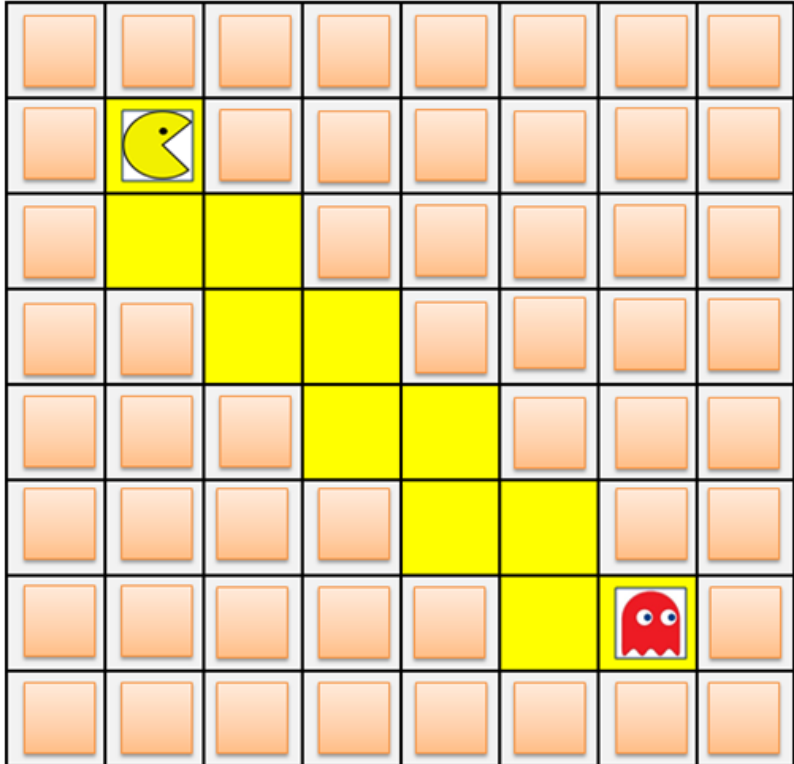
```

repetir 4 veces
  haz avanzar
repetir 3 veces
  haz
    girar a la derecha
  avanzar
            
```


Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	No	Sí	Secuenciación

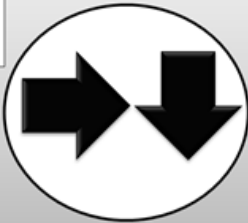
TPC (Versión 2.0) - ÍTEM 9

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?





Opción A

Repetir hasta llegar a... 




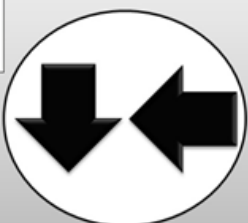
Opción B

Repetir hasta llegar a... 




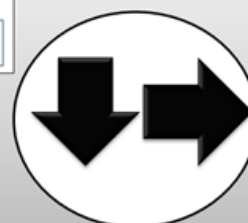
Opción C

Repetir hasta llegar a... 



Opción D ✔


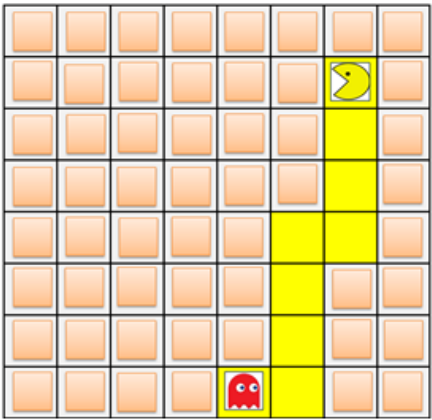
Repetir hasta llegar a... 



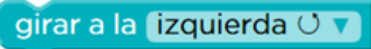
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por flechas	Sí	No	Sí	No	No	No	No	No	No	Secuenciación

TPC (Versión 2.0) - ÍTEM 10


¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?





Opción A




Opción B





Opción C



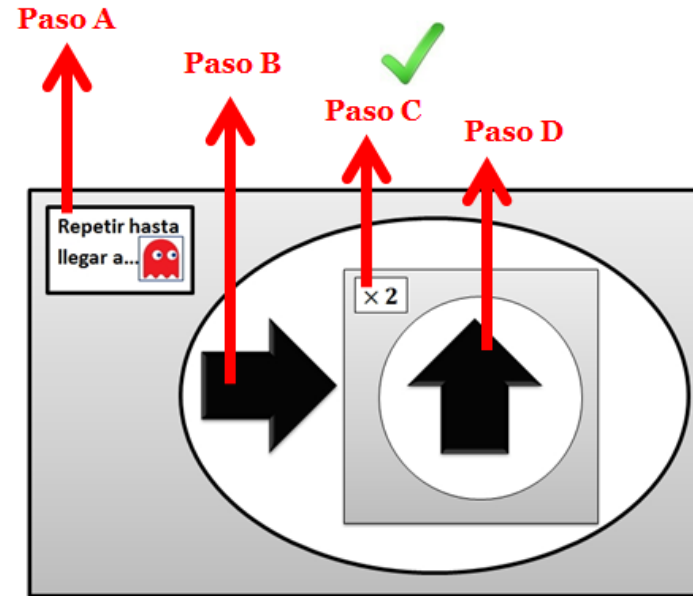
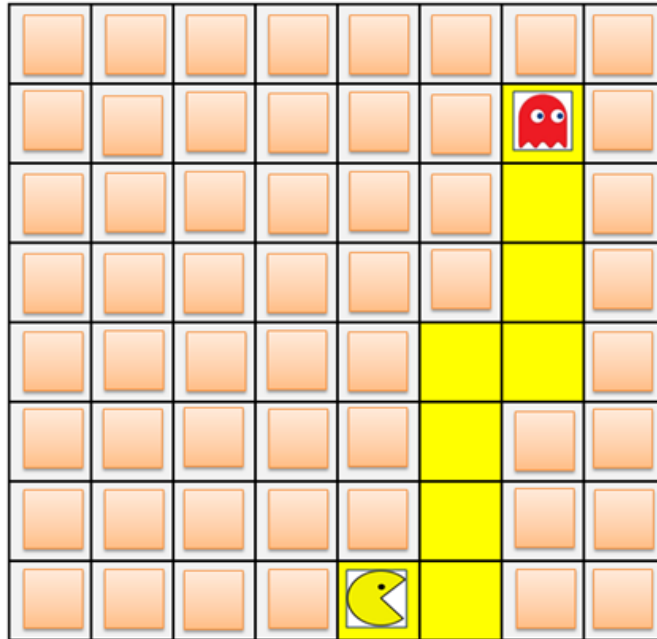
Opción D

No falta ningún bloque

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	No	No	No	No	No	Completamiento

TPC (Versión 2.0) - ÍTEM 11


Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?



Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por flechas	Sí	Sí	Sí	No	No	No	No	No	Sí	Depuración

TPC (Versión 2.0) - ÍTEM 12

¿Qué secuencia de órdenes debe ejecutar el artista para dibujar la escalera que llegue hasta la flor? Cada peldaño sube 30 píxeles



Opción A ✓

```

Repetir hasta la flor
haz
  repetir 4 veces
  haz
    mover hacia adelante 30 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 30 píxeles
            
```

Opción B

```

Repetir hasta la flor
haz
  repetir 4 veces
  haz
    mover hacia adelante 120 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 30 píxeles
            
```

Opción C

```

Repetir hasta la flor
haz
  repetir 4 veces
  haz
    mover hacia adelante 30 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 210 píxeles
            
```

Opción D

```

Repetir hasta la flor
haz
  repetir 7 veces
  haz
    mover hacia adelante 30 píxeles
    girar a la derecha por 90 grados
  saltar hacia adelante 30 píxeles
            
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Visual por bloques	Sí	Sí	Sí	No	No	No	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 13

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

Opción B ✓

Opción C

Opción D

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 14

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A ✓

```

Repetir hasta llegar a...
hacer
  avanzar
  si hay camino a la derecha
  hacer girar a la derecha
                    
```

Opción B

```

Repetir hasta llegar a...
hacer
  girar a la derecha
  si hay camino a la derecha
  hacer avanzar
                    
```

Opción C

```

Repetir hasta llegar a...
hacer
  avanzar
  si hay camino a la derecha
  hacer girar a la izquierda
                    
```

Opción D

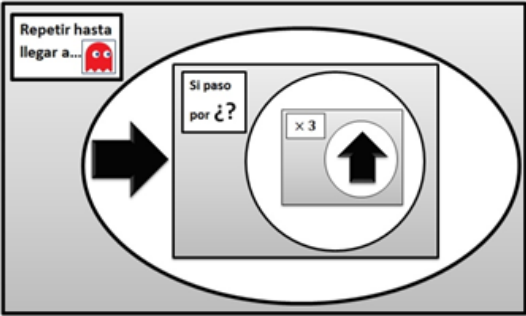
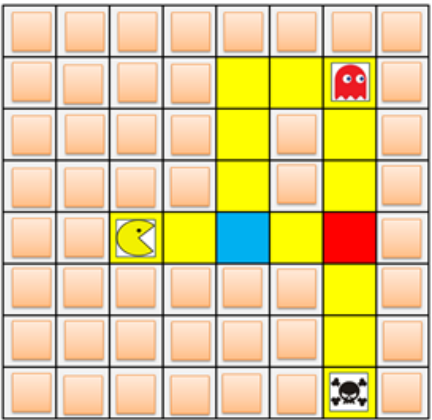
```

Repetir hasta llegar a...
hacer
  avanzar
  si hay camino a la izquierda
  hacer girar a la izquierda
                    
```


Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 15

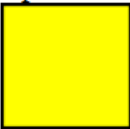
¿Qué falta en la siguiente secuencia de órdenes para llevar a 'Pac-Man' hasta el fantasma por el camino señalado?

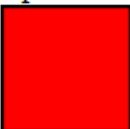
Opción A



Opción B




Opción C



Opción D

Tanto la opción A como la opción C son correctas



Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por flechas	Sí	Sí	Sí	Sí	No	No	No	No	Sí	Completamiento

TPC (Versión 2.0) - ÍTEM 16

Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	Sí	No	No	No	No	Sí	Depuración

TPC (Versión 2.0) - ÍTEM 17

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A

```

Repetir hasta llegar a...
hacer
  si hay un camino delante
  hacer avanzar
  sino girar a la izquierda
        
```

Opción B ✓

```

Repetir hasta llegar a...
hacer
  si hay un camino delante
  hacer avanzar
  sino girar a la derecha
        
```

Opción C

```

Repetir hasta llegar a...
hacer
  si hay camino a la derecha
  hacer girar a la derecha
  sino avanzar
        
```

Opción D

```

Repetir hasta llegar a...
hacer
  si hay camino a la izquierda
  hacer girar a la izquierda
  sino avanzar
        
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 18

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

Opción A ✔

```

Repetir hasta llegar a... (fantasma)
  hacer
    si hay un camino delante
      hacer avanzar
    sino girar a la izquierda
        
```

Opción B

```

Repetir hasta llegar a... (fantasma)
  hacer
    si hay un camino delante
      hacer avanzar
    sino girar a la derecha
        
```

Opción C

```

Repetir hasta llegar a... (fantasma)
  hacer
    si hay camino a la derecha
      hacer girar a la derecha
    sino avanzar
        
```

Opción D

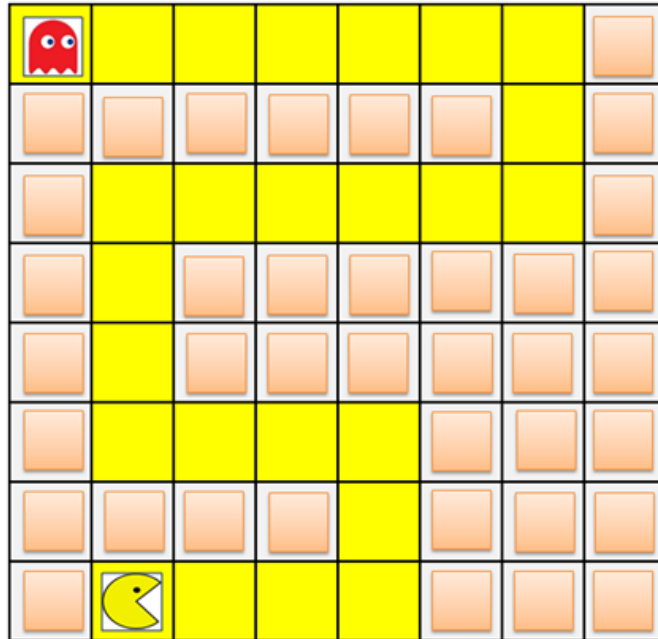
```

Repetir hasta llegar a... (fantasma)
  hacer
    si hay camino a la izquierda
      hacer girar a la izquierda
    sino avanzar
        
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 19

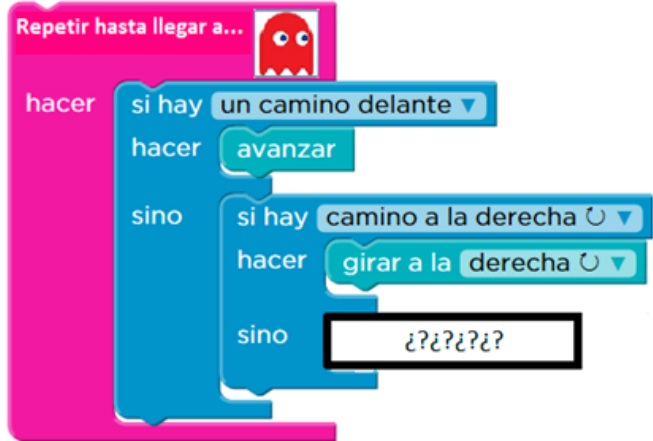
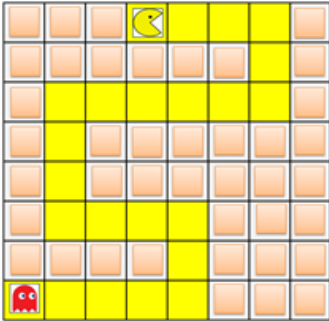
Para que 'Pac-Man' llegue hasta el fantasma por el camino señalado, ¿en qué paso de la siguiente secuencia de órdenes hay un **error**?




Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Depuración

TPC (Versión 2.0) - ÍTEM 20


¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' llegue hasta el fantasma por el camino señalado?





Opción A

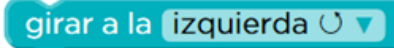


Opción B





Opción C



Opción D

No falta ningún bloque

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	Sí	No	Sí	No	No	No	Sí	Completamiento

TPC (Versión 2.0) - ÍTEM 21

¿Qué órdenes llevan a 'Pac-Man' por el camino señalado hasta las fresas e indican a 'Pac-Man' que se coma el número de fresas indicado?

Opción A ✔

```

mientras haya camino delante
hacer avanzar
repetir 3 veces
haz Comer 1 fresa
                    
```

Opción B

```

mientras haya camino delante
hacer avanzar
repetir 4 veces
haz Comer 1 fresa
                    
```

Opción C

```

mientras haya camino delante
hacer avanzar
repetir 5 veces
haz Comer 1 fresa
                    
```

Opción D

```

mientras haya camino delante
hacer avanzar
repetir 3 veces
haz Comer 1 fresa
                    
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 22

¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?

Opción A

```

mientras haya camino delante
haz
  repetir 5 veces
  hacer avanzar
  repetir 3 veces
  hacer Comer 1 fresa
        
```

Opción B ✔

```

mientras haya camino delante
hacer avanzar
  repetir 3 veces
  haz Comer 1 fresa
        
```

Opción C

```

mientras haya camino delante
haz
  repetir 3 veces
  hacer avanzar
  repetir 5 veces
  hacer Comer 1 fresa
        
```

Opción D


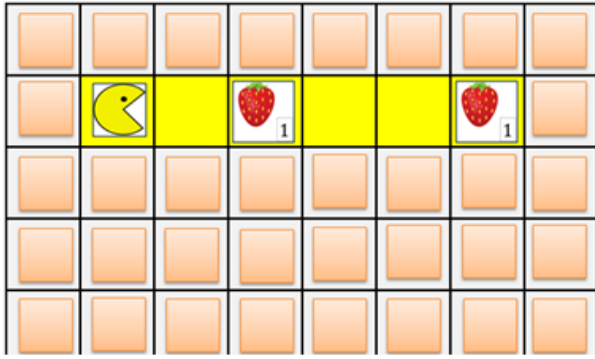
```

mientras haya camino delante
hacer avanzar
  repetir 3 veces
  haz Comer 1 fresa
        
```

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por bloques	Sí	Sí	No	No	No	Sí	No	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 23

¿Qué falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comenzando el número de fresas indicadas?

Opción A ✔
1 vez

Opción B
2 veces

Opción C
3 veces

Opción D
5 veces

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento

TPC (Versión 2.0) - ÍTEM 24

¿Qué bloque falta en la siguiente secuencia de órdenes para que 'Pac-Man' avance por el camino señalado comiendo el número de fresas indicadas (número desconocido)?

Opción A
Mientras haya camino delante

Opción B
Mientras no haya camino delante

Opción C
Mientras haya alguna fresa

Opción D
Mientras no haya ninguna fresa

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado							Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples			Funciones con parámetros
Laberinto	Visual por bloques	Sí	No	No	Sí	No	Sí	No	No	Sí	Completamiento

TPC (Versión 2.0) - ÍTEM 25

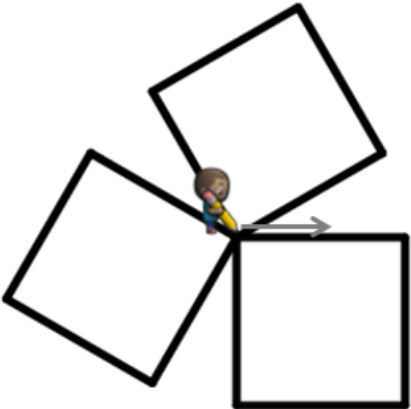
Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un cuadrado de 100 píxeles de lado:

Función

```

my function
  repetir 4 veces
  haz mover hacia adelante 100 píxeles
  girar a la derecha por 90 grados
    
```

¿Qué secuencia debe ejecutar el artista para dibujar el siguiente diseño? Cada uno de los lados de cada cuadrado mide 100 píxeles.



Opción A

```

repetir 3 veces
  haz my function
  girar a la derecha por 120 grados
    
```

Opción B

```

repetir 3 veces
  haz my function
  girar a la derecha por 120 grados
    
```

Opción C

```

repetir 4 veces
  haz my function
  girar a la derecha por 90 grados
    
```

Opción D

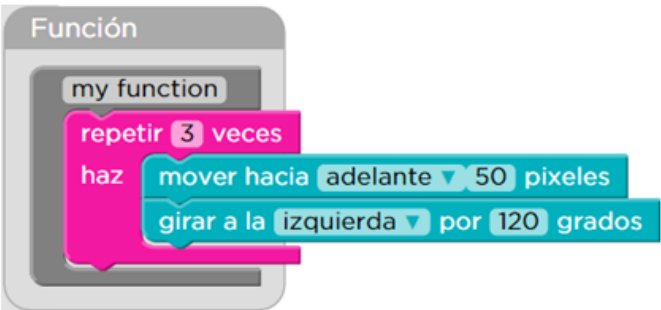
```

repetir 4 veces
  haz my function
  girar a la derecha por 90 grados
    
```


Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 26

Si tenemos el siguiente conjunto de órdenes, al que llamamos 'my function', y que dibuja un triángulo de 50 píxeles de lado:



¿Qué le falta a la siguiente secuencia para que el artista dibuje el siguiente diseño? Cada uno de los lados de cada triángulo mide 50 píxeles.




Opción A 15	Opción B 5 ✔
Opción C 4	Opción D 3

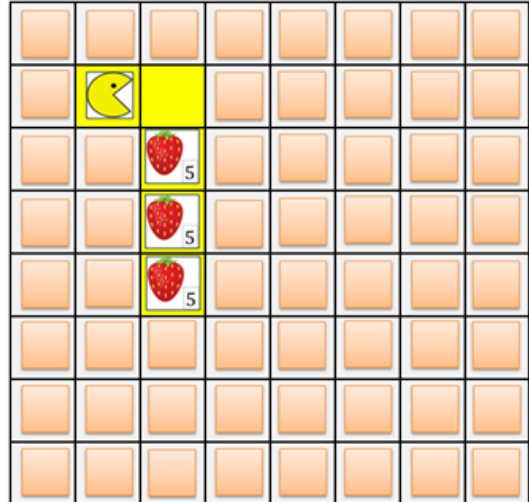
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Lienzo	Visual por bloques	Sí	Sí	No	No	No	No	No	Sí	No	Sí	Completamiento

TPC (Versión 2.0) - ÍTEM 27

Si tenemos el siguiente conjunto de órdenes, al que llamamos 'get 5':



¿Qué órdenes van llevando a 'Pac-Man' por el camino señalado e indicándole que se coma el número de fresas correspondiente?



Opción A ✔

```

avanzar
girar a la derecha
repetir 3 veces
haz
  avanzar
  get 5
        
```

Opción B

```

avanzar
girar a la derecha
repetir 3 veces
haz
  get 5
avanzar
        
```

Opción C

```

avanzar
girar a la derecha
repetir 5 veces
haz
  avanzar
  get 5
        
```

Opción D


```

avanzar
girar a la derecha
repetir 5 veces
haz
  get 5
avanzar
        
```

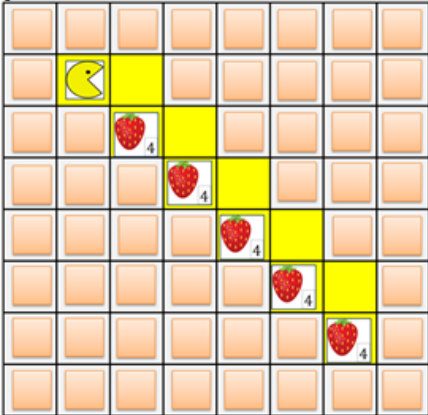
Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)			
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros		
Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	Sí	No	Sí	Secuenciación

TPC (Versión 2.0) - ÍTEM 28

Si tenemos el siguiente conjunto de órdenes, llamado 'move and get 4':



¿Qué falta en la siguiente secuencia para llevar a 'Pac-Man' por el camino señalado hasta las fresas, comiendo el número de fresas indicado?



repetir ??? veces
haz move and get 4

Opción A	Opción B
3	4
Opción C	Opción D
5	6

Entorno - Interfaz del reactivo	Estilo de las alternativas de respuesta	Concepto computacional abordado								Existencia de anidamiento	Tarea requerida	
		Direcciones	Bucles (loops)		Condicionales (conditionals)			Funciones (functions)				
			Repetir veces (repeat times)	Repetir hasta (repeat until)	Condicional simple (if)	Condicional compuesto (if/else)	Mientras que (while)	Funciones simples	Funciones con parámetros			
Laberinto	Visual por bloques	Sí	Sí	No	No	No	No	No	Sí	No	Sí	Completamiento

ANEXO G

Instrucciones, ejemplos y autoevaluación añadidos al TPC durante su virtualización

INSTRUCCIONES

El test está compuesto por 28 preguntas, distribuidas en 7 páginas con 4 preguntas en cada una de ellas.

Todas las preguntas tienen 4 opciones de respuesta (A, B, C ó D) de las cuales sólo una es correcta.

A partir de que comience el test dispones de 45 minutos para hacerlo lo mejor que puedas. No es imprescindible que contestes a todas las preguntas.

Para avanzar de una página a otra del test, en la parte inferior de la página debes pinchar sobre el botón 'Continuar'. MUY IMPORTANTE: cuando acabes o finalice el tiempo debes avanzar hasta la última página y pinchar sobre el botón 'Enviar' para que se guarden tus respuestas.

Si necesitas ampliar alguna pregunta para verla más grande, haz 'Ctrl+' con el teclado (o 'Ctrl-' para verla más pequeña)

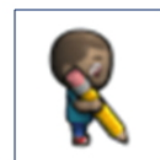
Antes de comenzar el test, vamos a ver 3 ejemplos para que te familiarices con el tipo de preguntas que te irás encontrando, y en la que aparecerán los personajes que ya te presentamos.



'Pac-Man'



Fantasma



Artista

¡ÁNIMO Y SUERTE!

EJEMPLO I

En este primer ejemplo se te pregunta cuáles son los órdenes que llevan a 'Pac-Man' hasta el fantasma por el camino señalado.

La opción correcta en este ejemplo es la B. Márcala en el botón de respuesta correspondiente, que está debajo de la pregunta.

¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

	<p>Opción A</p>
	<p>Opción B</p>
	<p>Opción C</p>
	<p>Opción D</p>

EJEMPLO II

En este segundo ejemplo se te pregunta de nuevo cuáles son los órdenes que llevan a 'Pac-Man' hasta el fantasma por el camino señalado. Pero en este caso las opciones de respuesta, en vez de ser flechas, son bloques que encajan unos con otros.

La opción correcta en este ejemplo es la C. Márcala en el botón de respuesta correspondiente, que está debajo de la pregunta.


¿Qué órdenes llevan a 'Pac-Man' hasta el fantasma por el camino señalado?

	<p>Opción A</p>	<p>Opción B</p>
	<p>Opción C</p>	<p>Opción D</p>

EJEMPLO III

En este tercer ejemplo se te pregunta qué órdenes debe seguir el artista para dibujar la figura que aparece en pantalla. Es decir, cómo debe MOVER el lápiz para que se dibuje la figura.

La opción correcta en este ejemplo es la A. Márcala en el botón de respuesta correspondiente, que está debajo de la pregunta.

<p>¿Qué órdenes debe ejecutar el artista para dibujar la figura? El lado corto mide 50 píxeles y el lado largo 100 píxeles.</p> 	<p>Opción A</p> <p>mover hacia adelante ▼ 50 píxeles</p> <p>girar a la izquierda ▼ por 90 grados</p> <p>mover hacia adelante ▼ 100 píxeles</p> <p style="text-align: right;">✔</p>	<p>Opción B</p> <p>mover hacia adelante ▼ 50 píxeles</p> <p>girar a la derecha ▼ por 90 grados</p> <p>mover hacia adelante ▼ 100 píxeles</p>
	<p>Opción C</p> <p>mover hacia adelante ▼ 100 píxeles</p> <p>girar a la izquierda ▼ por 90 grados</p> <p>mover hacia adelante ▼ 50 píxeles</p>	<p>Opción D</p> <p>mover hacia adelante ▼ 100 píxeles</p> <p>girar a la derecha ▼ por 90 grados</p> <p>mover hacia adelante ▼ 50 píxeles</p>

AUTOEVALUACIÓN

Por favor, contesta sinceramente a estas dos breves preguntas de autoevaluación. Luego pincha sobre el botón 'ENVIAR' para finalizar el test y que tus respuestas queden guardadas.

De 0 a 10, ¿cómo consideras que te ha salido el Test?

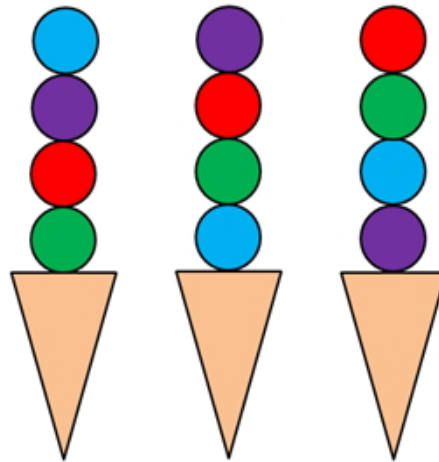
De 0 a 10, ¿cómo consideras que se te dan los ordenadores y la informática?

ANEXO H

Tareas Bebras utilizadas para la validación convergente del TPC

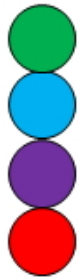
ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B1-1	<i>'Ice Cream Machine'</i> (La Máquina de Helados)	Lithuanian Bebras Contest 2013 (Dagiene & Stupuriene, 2014)	?	Detectar un algoritmo; funcionamiento de máquinas; bucles	5º y 6º de Primaria
B1-2	<i>'Dress Code for Beavers'</i> (La Vestimenta de los Castores)	Lit. Bebras Web – Task examples	ALG, STRUC	?	
B1-3	<i>'Tic-tac-toe'</i> (Las Tres en Raya)	Bebras UK-2014	?	Abstracción y pensamiento algorítmico	
B2-1	<i>'Water Supply'</i> (Suministro de agua)	Lit. Bebras Web – Task examples	STRUC	?	1º y 2º de la ESO
B2-2	<i>'Fast Laundry'</i> (Lavandería rápida)	Lit. Bebras Web – Task examples	ALG	?	
B2-3	<i>'Abacus'</i> (Ábaco chino)	Bebras UK-2014	?	Abstracción, descomposición y pensamiento algorítmico	
B3-1	<i>'Friends'</i> (La red social)	Lit. Bebras Web – Task examples	STRUC, SOC	?	3º y 4º de la ESO
B3-2	<i>'Visiting Friends'</i> (Visitando amigos)	Lithuanian Bebras Contest 2013 (Dagiene & Stupuriene, 2014)	ALG, STRUC	Conteo. Detección de patrones y periodicidad	
B3-3	<i>'Mobile Phones'</i> (Teléfonos móviles)	Bebras UK-2014	?	Abstracción, descomposición y evaluación	

La máquina de helados siempre pone las bolas en el mismo orden. En la siguiente foto, se pueden ver tres ejemplos de ello:



De las opciones que aparecen debajo, **¿cuál de ellas podría venir de la máquina de helados?**

Opción A



Opción B



Opción C



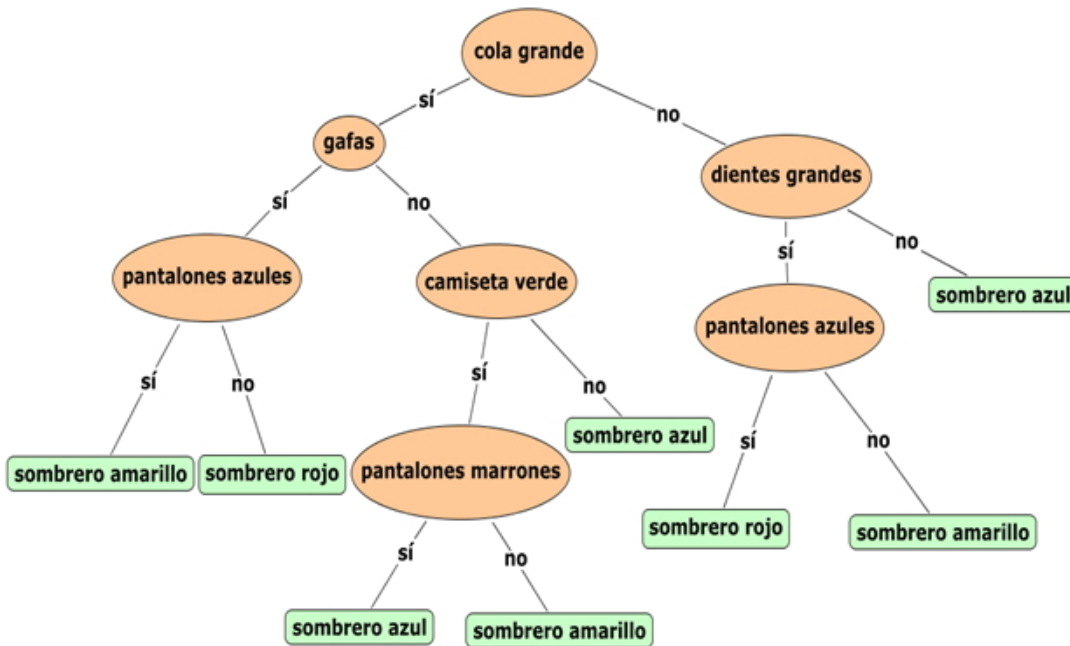
Opción D



ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B1-1	'Ice Cream Machine' (La Máquina de Helados)	Lithuanian Bebras Contest 2013 (Dagiene & Stupuriene, 2014)	?	Detectar un algoritmo; funcionamiento de máquinas; bucles	5º y 6º de Primaria

A los castores les gustan los sistemas complejos de reglas y, en consecuencia, han establecido un nuevo código de vestimenta. Usa el gráfico para determinar si un castor está vestido correctamente o incorrectamente. Este gráfico se denomina 'árbol' porque hay un único nodo raíz (el superior) con ramas de conexión a otros nodos (como un árbol real). En cada nodo tienes que decidir en qué dirección quieres ir dentro del árbol, no se puede volver a subir.

¿Qué castor no se viste según el código de vestimenta?



Opción A



Opción B



Opción C



Opción D



ID	Nombre de la Tarea	Fuente	Categoría	Descriptor	Grupo de Edad
B1-2	'Dress Code for Beavers' (La Vestimenta de los Castores)	Lit. Bebras Web – Task examples	ALG, STRUC	?	5° y 6° de Primaria

Estás jugando a las ‘Tres en Raya’ con un amigo. Primero tu amigo comienza colocando una ‘O’ en la casilla que prefiera, luego tú colocas una ‘X’ en otra casilla. Y así el juego continúa por turnos. Gana el jugador que coloca sus tres marcas formando una línea horizontal, vertical o diagonal.

Ahora es tu turno para colocar una ‘X’ en el tablero de abajo.

¿En qué casilla debes colocar tu ‘X’ para tener las máximas posibilidades de ganar el juego?

X	1	2
O	O	X
O	3	4

Opción A
Casilla 1

Opción B
Casilla 2

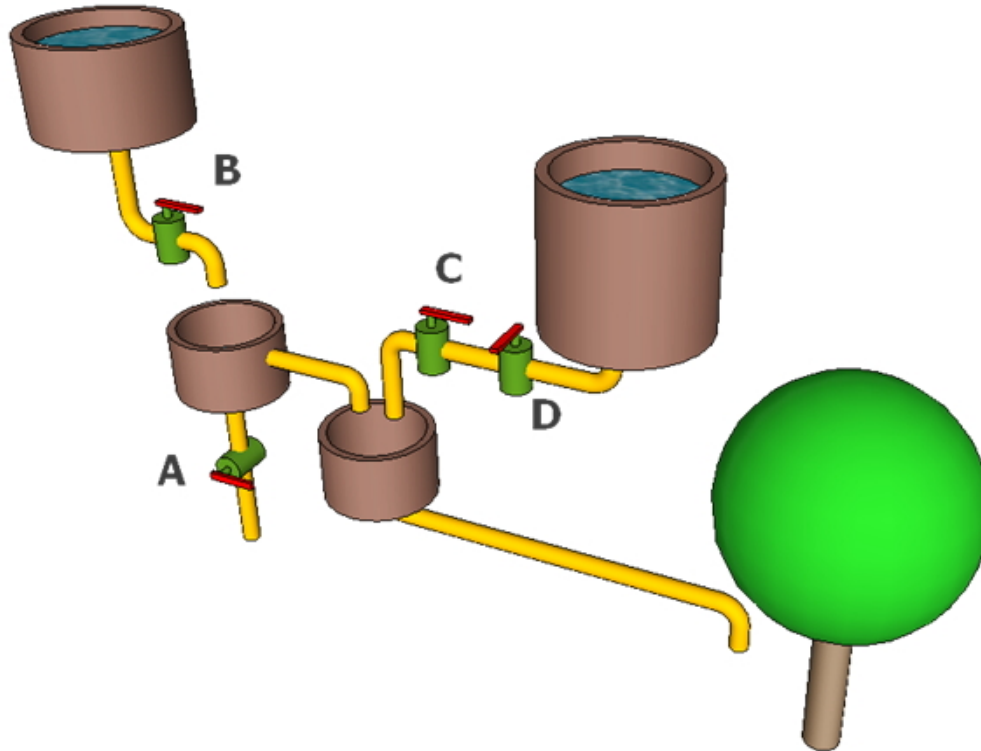
Opción C
Casilla 3

Opción D
Casilla 4

ID	Nombre de la Tarea	Fuente	Categoría	Descriptor	Grupo de Edad
B1-3	‘Tic-tac-toe’ (Las Tres en Raya)	Bebras UK-2014	?	Abstracción y pensamiento algorítmico	5° y 6° de Primaria

Hemos construido un sistema de tuberías para regar un árbol. El sistema contiene las variables A, B, C y D, que pueden ser verdaderas o falsas. Una variable tiene el valor verdadero, si el grifo en cuestión está abierto, y falso, si está cerrado.

¿En qué caso llega agua hasta el árbol?



Caso 1

A = falso, B = verdadero, C = falso, D = falso

Caso 2

A = verdadero, B = verdadero, C = falso, D = falso

Caso 3

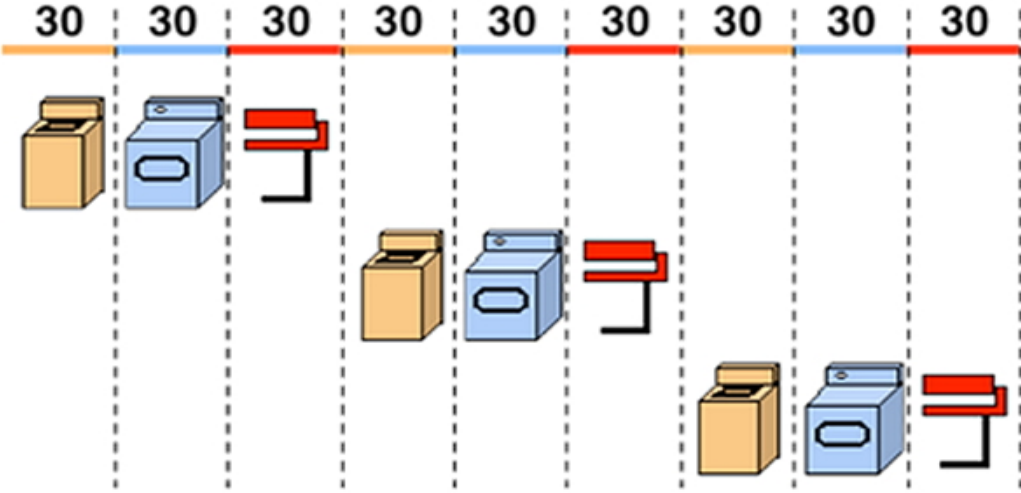
A = verdadero, B = falso, C = falso, D = verdadero

Caso 4

A = falso, B = falso, C = falso, D = verdadero

ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B2-1	'Water Supply' (Suministro de agua)	Lit. Bebras Web – Task examples	STRUC	?	1º y 2º de la ESO

Jorge ha abierto su nuevo negocio: una lavandería. Jorge tiene tres máquinas: una lavadora, una secadora y una plancha. Cada máquina tarda 30 minutos en realizar su trabajo. Así, cuando llega un cliente, éste necesita 90 minutos para completar la colada: lavado, secado y planchado. Y tres clientes que utilicen las máquinas sin solaparse necesitarán en consecuencia 270 minutos.



30 30 30 30 30 30 30 30 30

Opción A
90 minutos

Opción B
120 minutos

Opción C
150 minutos

Opción D
270 minutos

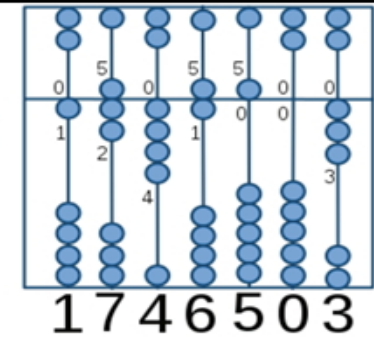
Pero entonces, aparecen 3 clientes con mucha prisa. Cada uno de ellos tiene suficiente ropa para una carga propia. Y todos están de acuerdo en que quieren terminar lo antes posible. Si optimizan el uso de las máquinas...

¿Cuántos minutos llevará que los 3 clientes consigan terminar su colada?

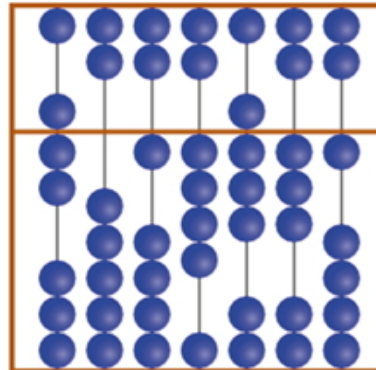
ID	Nombre de la Tarea	Fuente	Categoría	Descriptorios	Grupo de Edad
B2-2	'Fast Laundry' (Lavandería rápida)	Lit. Bebras Web – Task examples	ALG	?	1º y 2º de la ESO

En un ábaco chino, un número se representa por la posición de las bolitas. El valor de una bolita en la parte superior es de 5; el valor de una bolita en la parte inferior es de 1. El ábaco se pone a cero empujando las bolitas lejos del centro.

Así, en el ejemplo de la derecha las bolitas del ábaco están representando el número **1746503**



¿Qué número representa el siguiente ábaco chino?



Opción A 3014431	Opción B 7514831	Opción C 3514431	Opción D 7014831
----------------------------	----------------------------	----------------------------	----------------------------

ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B2-3	'Abacus' (Ábaco chino)	Bebras UK-2014	?	Abstracción, descomposición y pensamiento algorítmico	1º y 2º de la ESO

Lucia y sus amigos están registrados en una red social. Aquí están los amigos de Lucia, y los amigos de sus amigos.

En el gráfico, una línea significa amistad entre dos personas. Por ejemplo, Monica es amiga de Lucia pero Alex no es amigo de Lucia.

- Si alguien comparte una foto con alguno de sus amigos entonces esos amigos pueden hacer comentarios sobre la misma
- Entonces, si alguien comenta la foto, todos sus amigos pueden ver tanto el comentario como la foto, pero no pueden hacer comentarios sobre la misma a menos que originalmente pudieran.

Lucia ha subido una foto, ¿con qué amigos puede Lucia compartir la foto si no quiere que Jacob pueda verla?

Opción A
Dana, Michael, Eve

Opción B
Dana, Eve, Monica

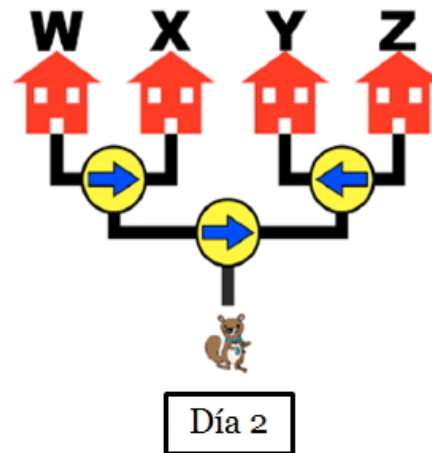
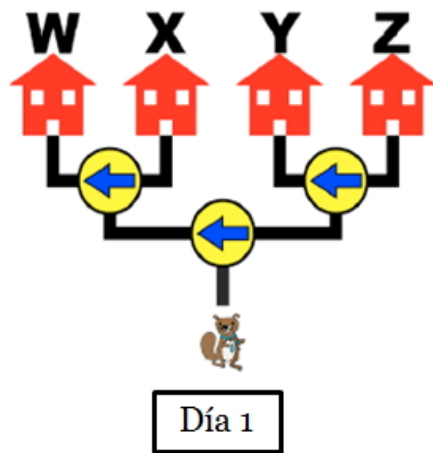
Opción C
Michael, Eve, Jacob

Opción D
Michael, Peter, Monica

ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B3-1	'Friends' (La red social)	Lit. Bebras Web – Task examples	STRUC, SOC	?	3º y 4º de la ESO

El Sr. Castor tiene 4 amigos. Cada uno de ellos vive en una casa diferente. El Sr. Castor sale cada día de su casa para visitar a un amigo. Cuando el Sr. Castor llega a un cruce en el camino sigue la flecha, pero, según pasa, deja girada la flecha para que apunte en la dirección contraria (El Sr. Castor ignora las flechas en su camino de regreso a casa)

En el día 1, el Sr. Castor visita al amigo W. En el día 2, visita al amigo Y, ya que dejó giradas dos flechas el día anterior. Etc...



¿Qué amigo visita el Sr. Castor el día 30?

Opción A

Amigo W

Opción B

Amigo X

Opción C

Amigo Y

Opción D

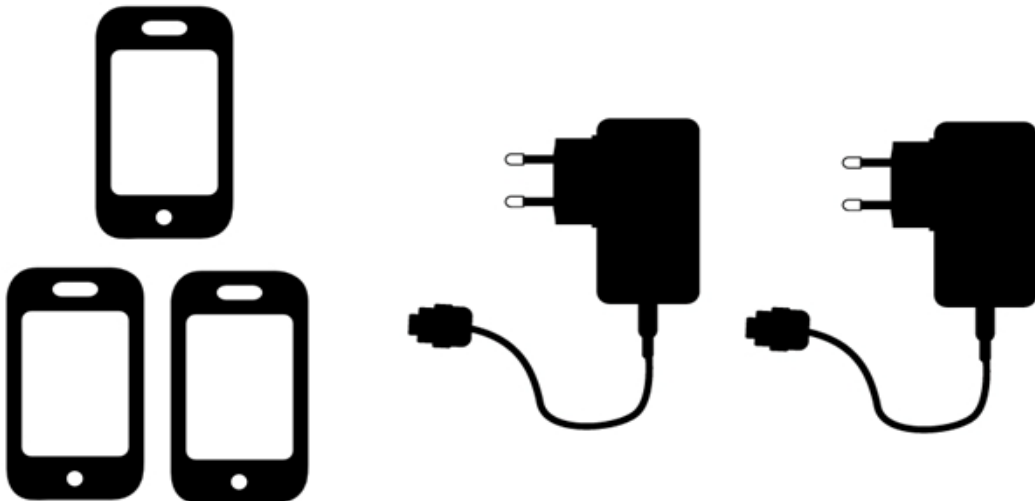
Amigo Z

ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B3-2	'Visiting Friends' (Visitando amigos)	Lithuanian Bebras Contest 2013 (Dagiene & Stupuriene, 2014)	ALG, STRUC	Conteo. Detección de patrones y periodicidad	3º y 4º de la ESO

En tu familia hay tres teléfonos móviles, pero ninguno de ellos tiene batería. Se tarda 1 hora en cargar completamente cada teléfono móvil, pero esto no tiene por qué hacerse todo seguido.

Tu familia sólo tiene dos cargadores de teléfonos móviles en casa.

¿Cuál es el menor tiempo que necesita tu familia para recargar completamente los tres teléfonos?



Opción A
3 horas

Opción B
2 horas

Opción C
1 hora y media

Opción D
1 hora

ID	Nombre de la Tarea	Fuente	Categoría	Descriptores	Grupo de Edad
B3-3	'Mobile Phones' (Teléfonos móviles)	Bebras UK-2014	?	Abstracción, descomposición y evaluación	3º y 4º de la ESO

ANEXO I

Correo para el profesor con las instrucciones de aplicación del TPC

Hola xxxxxxxx,

Te escribo con las instrucciones de aplicación de nuestro TEST DE PENSAMIENTO COMPUTACIONAL (TPC), que está alojado en la siguiente URL (puedes probarlo si lo deseas):

<http://goo.gl/IYEKMB>

Al respecto de este test:

- En la primera página cumplimentan sus datos; están unas breves instrucciones que podéis leer en alto antes de comenzar; y 3 ejemplos de preguntas con la respuesta correcta para que se familiaricen.
- A partir de la segunda página están las 28 preguntas del test (4 por página). Tienen un máximo de 45 minutos para cumplimentarlo (hemos hecho algún piloto aquí en Madrid y es tiempo suficiente)
- Es importante que cuando finalicen las preguntas o llegado el tiempo límite, avancen hasta la última página y pulsen en ENVIAR para que se guarden las respuestas

En ambas sesiones es **IMPORTANTE** que, antes de que comiencen a realizar la prueba, les des estas indicaciones generales para que se enfoquen adecuadamente en la tarea a realizar:

- Les tranquilices, pues no son exámenes de los que se deriva una calificación escolar.
- Les motives a concentrarse y hacerlo lo mejor posible.
- Les recuerdes que la prueba se realiza individualmente y en silencio.

¿OK?... Si tienes cualquier duda acerca del procedimiento, podemos hablar por teléfono cuando desees.

Un abrazo y seguimos adelante,

ANEXO J

Cuestionario para profesores sobre la implantación del Curso K-8 de Code.org

El objetivo de este cuestionario es controlar las distintas variables en la implantación del Curso K-8 de Code.org [<http://studio.code.org/s/1>] en los distintos centros de la investigación, para así poder interpretar mejor los resultados de las distintos tests aplicados a los participantes.

Nombre del profesor/a:

Apellidos del profesor/a:

Centro Educativo:

¿Cuántas semanas lectivas has dedicado en tu/s grupo/s al Curso K-8 de Code.org?

Si tienes varios grupos que hayan realizado el K-8, y cada grupo ha dedicado un número distinto de semanas, indica un valor promedio

- 5 semanas o menos
- 6 semanas
- 7 semanas
- 8 semanas
- 9 semanas
- 10 semanas o más

De las 20 etapas de las que se compone el Curso K-8 de Code.org [<http://studio.code.org/s/1>], ¿qué porcentaje aproximado has impartido en tu/s grupo/s?

Si tienes varios grupos que hayan realizado el K-8, y en cada grupo has impartido un porcentaje distinto, indica un valor promedio

- 50% de las etapas o menos
- 60%
- 70%
- 80%
- 90%
- 100% de las etapas

De las 20 etapas de las que se compone el Curso K-8 de Code.org [<http://studio.code.org/s/1>], ¿cuál es la etapa más avanzada que has llegado a impartir a alguno/s de tu/s grupo/s?:

¿Qué parte del Curso K-8 de Code.org has impartido en tu/s grupo/s?

- Sólo etapas '*on-line*'

- Sólo etapas '*unplugged*' (sin necesidad de dispositivos electrónicos)
- Tanto etapas '*on-line*' como '*unplugged*'
- Otro:

¿Qué metodología de implantación del Curso K-8 de Code.org has utilizado en tu/s grupo/s?

- Todos los estudiantes han ido avanzando al mismo ritmo por las etapas del curso, según he ido indicando en el aula
- Cada estudiante ha ido avanzando a su propio ritmo (sólo en el aula)
- Cada estudiante ha ido avanzando a su propio ritmo (tanto en el aula como en su casa)
- Otro:

¿Has tenido estudiantes muy avanzados que hayan finalizado el Curso K-8 significativamente antes que el resto de sus compañer@s?

- Sí
- No

En caso afirmativo, ¿cómo has manejado esa situación?

- Los estudiantes avanzados, una vez finalizaron el Curso K-8, se dedicaron a ayudar a sus compañer@s más lentos
- Los estudiantes avanzados, una vez finalizaron el Curso K-8, comenzaron aprendizajes de nivel superior (p.e. curso de Khan Academy, etc...)
- Otro:

Como profesor/a responsable de la implantación del Curso K-8 de Code.org, ¿cuál es tu nivel de satisfacción con el curso globalmente?

- 1 = Muy bajo
- 2
- 3
- 4
- 5 = Muy alto

Globalmente, ¿cuál crees que ha sido el grado de aprendizaje de tus estudiantes con el Curso K-8?

- 1 = Poco aprendizaje
- 2
- 3
- 4
- 5 = Mucho aprendizaje

Después de la experiencia, ¿tienes intención de impartir el Curso K-8 nuevamente el próximo año académico?

- Sí
- No

Brevemente, indica los aspectos más positivos y más negativos (puntos mejorables) del Curso K-8:

Opcionalmente, relata alguna anécdota o incidente especialmente reseñable que ilustre vuestra experiencia con el Curso K-8:

Finalmente, con los grupos que han realizado el Curso K-8 durante el 2º trimestre, ¿tienes intención de trabajar con Scratch durante el 3º trimestre?

En colaboración con Jesús Moreno, del colectivo Programamos.es [<http://programamos.es/>], estamos investigando las relaciones entre nuestro Test de Pensamiento Computacional y la calidad de los proyectos hechos con Scratch medidos con su herramienta "*Dr. Scratch*" [<http://drscratch.programamos.es/>]

- Sí
- No

En caso afirmativo, ¿deseas participar con tus grupos en esta extensión de nuestra investigación?

Esta extensión no requerirá más tests. Sólo tendríamos que aplicar la herramienta "*Dr.Scratch*" como medida de la calidad de los proyectos que hagan tus estudiantes durante el 3º trimestre ;-)

- Sí
- No

ANEXO K

Cuestionario de Satisfacción para estudiantes sobre el Curso K-8 de Code.org

El objetivo de este cuestionario es conocer tu opinión sobre el Curso “*K-8 Intro to Computer Science*” de Code.org [<http://studio.code.org/s/1>] que has estado realizando durante las últimas semanas

Nombre:

Apellidos:

Sexo:

- Chico
- Chica

Centro Educativo:

Curso:

- 5° Primaria
- 6° Primaria
- 1° ESO
- 2° ESO
- 3° ESO
- 4° ESO

¿Cuál es tu nota habitual en las siguientes asignaturas (o similares)?:

	Suspense	Aprobado	Notable Bajo	Notable Alto	Sobresaliente
Matemáticas					
Lengua					
Música					
Tecnología / Informática					
Plástica					
Ciencias					

¿Habías oído hablar de la 'programación informática' ('coding') antes de este Curso "K-8 Intro to Computer Science"?:

- Sí
- No

¿Habías 'programado' alguna vez antes de este Curso "K-8 Intro to Computer Science"?

- Sí
- No

¿En qué medida crees que el Curso "K-8 Intro to Computer Science" te ha ayudado a desarrollar cada una de las siguientes aptitudes y habilidades?

	1 = Nada	2 = Poco	3 = Algo	4 = Bastante	5 = Mucho
Pensamiento Lógico					
Creatividad					
Resolución de Problemas					
Trabajo en Equipo					
Persistencia-Perseverancia					
Capacidad de Atención					

Tras tu experiencia con el Curso "K-8 Intro to Computer Science", ¿cómo dirías que se te da la 'programación informática' ('coding')?

- 1= Se me da muy mal
- 2
- 3
- 4
- 5 = Se me da muy bien

¿Has conseguido completar el 100% de los niveles-pantallas propuestos en el Curso "K-8 Intro to Computer Science"?

- Sí
- No

Tras el Curso "K-8 Intro to Computer Science", ¿cómo de motivado te sientes para continuar aprendiendo 'programación informática' ('coding')?

- 1= Nada motivado
- 2
- 3
- 4
- 5 = Muy motivado

¿Te gustaría continuar aprendiendo 'programación informática' ('coding') a través de plataformas-tutoriales en Internet (a tu propio ritmo, de forma interactiva...)?

- Sí
- No

Globalmente, ¿qué valoración haces del Curso "K-8 Intro to Computer Science"?

- 1= No me ha gustado nada
- 2
- 3
- 4
- 5 = Me ha gustado mucho

Globalmente, ¿cuánto crees que has aprendido con el Curso "K-8 Intro to Computer Science"?

- 1= No he aprendido nada
- 2
- 3
- 4
- 5 = He aprendido mucho

De todas las etapas que tiene el Curso "K-8 Intro to Computer Science" [<http://studio.code.org/s/1>], ¿Cuál es la que más te ha gustado? ¿Cuál es la que menos te ha gustado? Explica por favor tu respuesta:

Te puedes referir tanto a etapas "on-line" (hechas en pantalla) como a etapas "unplugged" (hechas sin ordenador)

¿Consideras importante la 'programación informática' ('coding') para tu formación como ciudadano del siglo XXI? ¿Sí, No? ¿Por qué? Razona por favor tu respuesta:

ANEXO L

Testimonios gráficos del Curso K-8 de Code.org

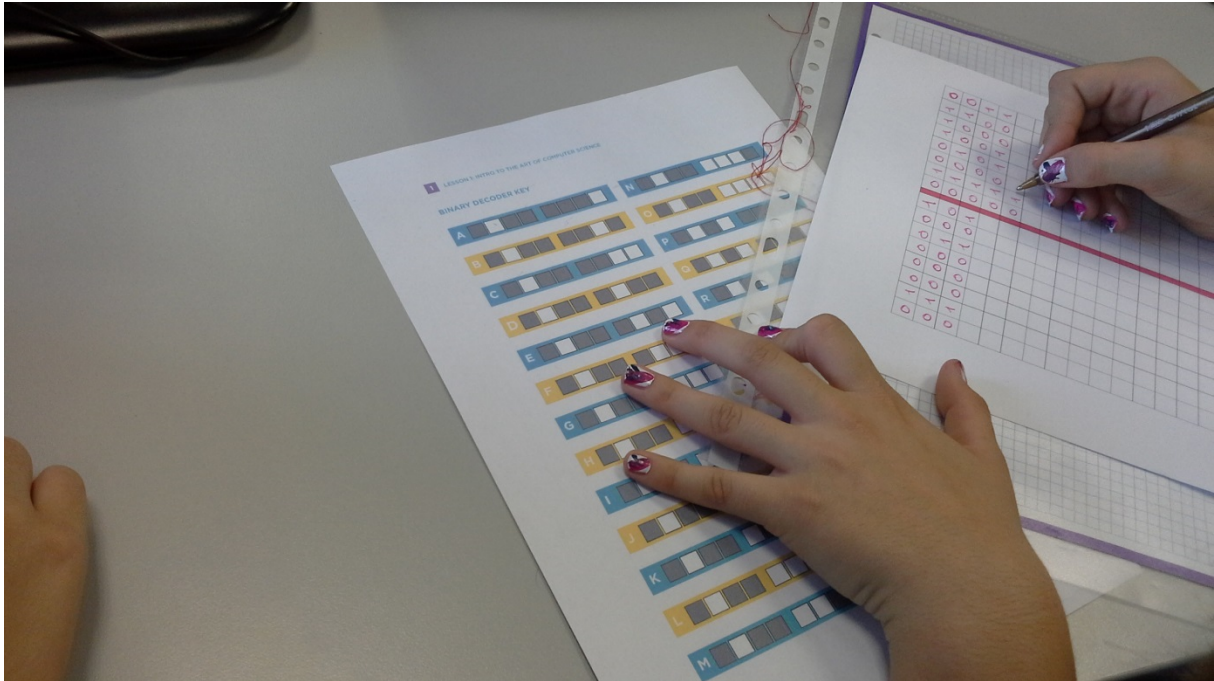


Figura L.1. Foto de la Etapa 1 ('Introducción a las Ciencias de la Computación') del Curso K-8

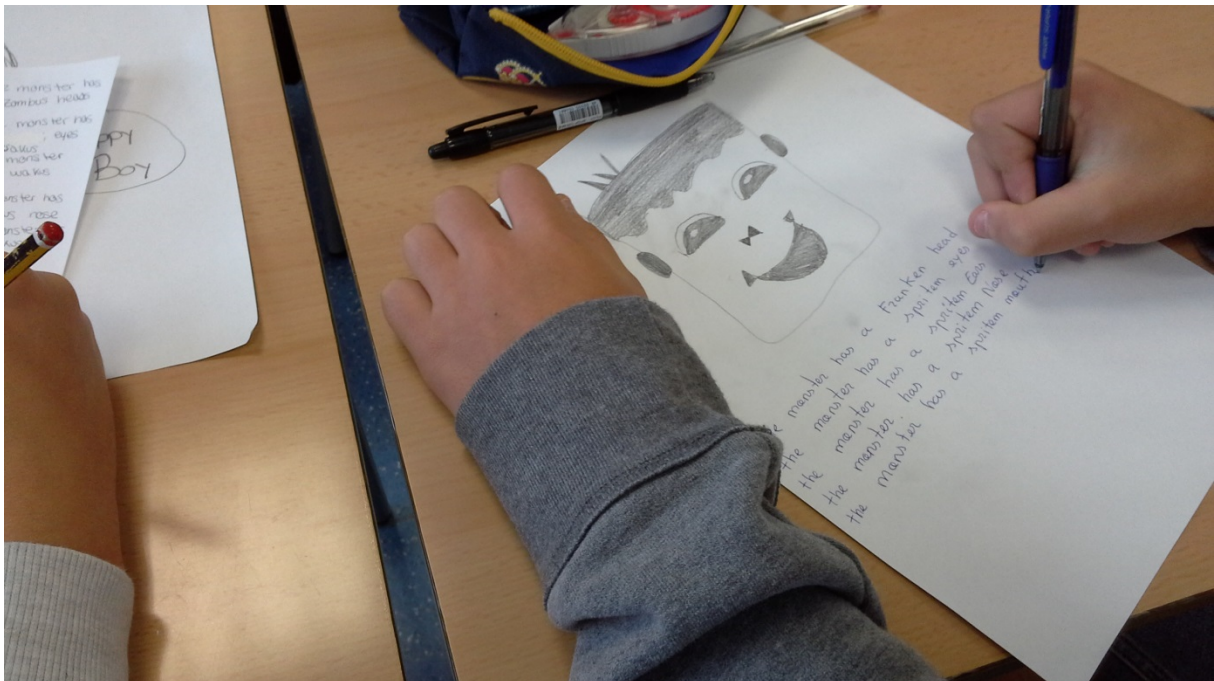


Figura L.2. Foto de la Etapa 3 ('Pensamiento Computacional') del Curso K-8

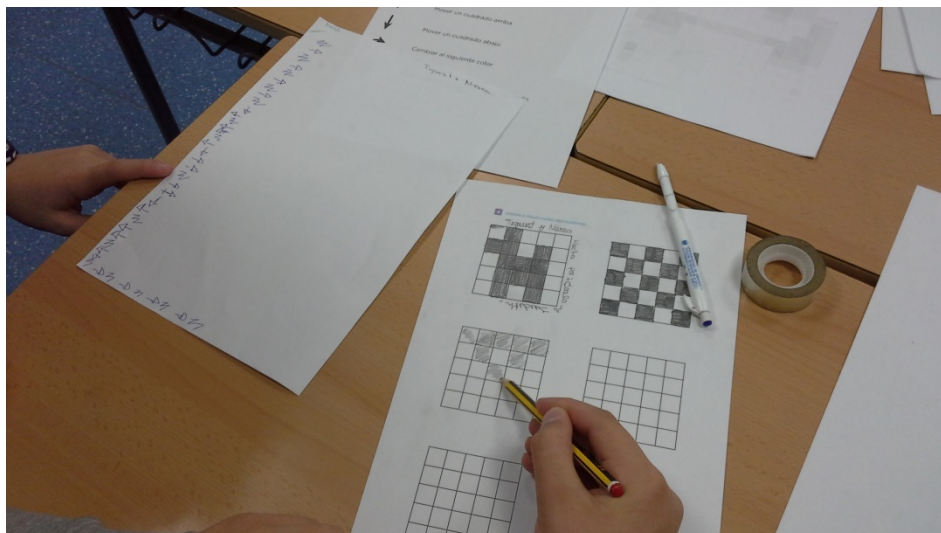


Figura L.3. Foto de la Etapa 4 ('Programación sobre hoja cuadrículada') del Curso K-8



Figura L.4. Foto de la Etapa 6 ('Algoritmos') del Curso K-8



Figura L.5. Foto de la Etapa 8 ('Funciones') del Curso K-8