

TESIS DOCTORAL

2020



Análisis del aprendizaje de los
estudiantes en un entorno educativo con
actividades de robótica

SAMANTHA ORLANDO

**PROGRAMA DE DOCTORADO EN SISTEMAS
INTELIGENTES**

DIRECTORES

Elena Gaudioso Vázquez

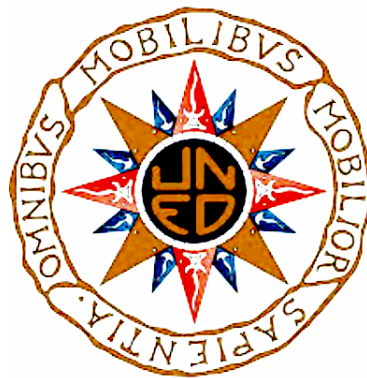
Félix de la Paz López

UNIVERSIDAD NACIONAL DE EDUCACIÓN A DISTANCIA

Escuela Técnica Superior de Ingeniería Informática

Departamento de Inteligencia Artificial

TESIS DOCTORAL



Análisis del aprendizaje de los estudiantes en un entorno
educativo con actividades de robótica

SAMANTHA ORLANDO

DIRECTORES

Elena Gaudioso Vázquez

Félix de la Paz López

Departamento de Inteligencia Artificial

ETSI Informática

Universidad Nacional de Educación a Distancia

2020

*A mis hijos Francesco y Teresa
tesoros que la vida quiso regalarme.*

Agradecimientos

Esta tesis es el resultado de mis intentos a lo largo de estos últimos años, de encontrar modalidades de enseñanza capaces de utilizar los nuevos instrumentos tecnológicos como herramienta didáctica para animar a los estudiantes en su formación y ayudar a los profesores guiar eficazmente a cada uno de sus alumnos.

En primer lugar, debo dar las gracias a los profesores del Departamento de Inteligencia Artificial de la UNED Dra. Elena Gaudioso Vázquez y Dr. Félix De la Paz, directora y codirector de esta tesis por haber confiado en mí, y por sus constantes apoyos y orientaciones, además de por su amistad.

Al Departamento de Inteligencia Artificial de la UNED, que me ha dado la oportunidad de trabajar en el laboratorio Jose Mira del departamento, permitiéndome el desarrollo de los conocimientos técnicos necesarios para complementar mi formación teórica.

Agradezco al Ministero Italiano dell'università e della ricerca (MIUR), por haberme concedido la financiación necesaria para mi formación y a la directora del Instituto IPSIA Pacinotti Maria Missaggia por la comprensión demostrada durante estos años.

Agradezco al colegio italiano de Madrid que me dio la posibilidad de hacer nuestra primera experimentación; en particular agradezco al profesor Paolo Ricci por darnos la bienvenida en su clase y por incluir nuestra actividad en el currículo de Física.

Igualmente agradezco al instituto Cabrini de Madrid, que nos abrió sus puertas para nuestra segunda experimentación. Por último, agradezco el instituto Gredos San Diego de la Suerte de Madrid que nos concedió una excelente organización en nuestra ultima experimentación.

Agradezco a todos los alumnos que han colaborado en esta investigación con su participación en las experimentaciones en clase y cuyo entusiasmo ha supuesto un estímulo importante para desarrollar el trabajo de este proyecto.

Finalmente, no puedo dejar de mencionar a mi marido y su contribución a la consecución de los objetivos de este trabajo; siempre presente, inspirador de ideas en muchas conversaciones informales y siempre capaz de motivar mi investigación.

Resumen

La evolución de la sociedad contemporánea en el campo de la tecnología ha transferido naturalmente el uso de herramientas tecnológicas a las aulas.

Las mentes de los alumnos se estimulan más mediante la activación de modernos canales de comunicación y el desarrollo de habilidades digitales que pueden emplearse en diferentes contextos reales.

La robótica educativa es una herramienta útil para favorecer los procesos cognitivos gracias al interés que los estudiantes muestran en la interacción con un robot. Además, la robótica como recurso educativo permite el desarrollo natural del conocimiento de Ciencia, Tecnología, Ingeniería y Matemáticas (STEM). En este trabajo de tesis se presenta un entorno de aprendizaje para el estudio de la física en la educación secundaria. Se propone el uso de un componente robótico para la realización de la experimentación de los conceptos aprendidos en clase. Así, el alumno puede reproducir el problema propuesto, pensar posibles soluciones, ver los resultados de sus acciones, aprender el contenido didáctico y crear relaciones con los contenidos discutidos. El alumno aprende a interactuar con el robot y con el sistema, guiado por el profesor tratando de encontrar soluciones a los problemas que encuentra. El alumno desarrolla su conocimiento al preguntarse acerca de los posibles pasos a seguir, cometiendo errores y volviéndolo a intentar.

En este trabajo, la evaluación del alumno es una herramienta fundamental para que el profesor mejore el aprendizaje del alumno; evaluar no significa únicamente asignar una calificación a cada estudiante. Significa averiguar posibles carencias de conocimiento de los estudiantes para poder enfocar mejor su aprendizaje. El sistema, gracias al uso de modelos obtenidos a partir de la interacción de los estudiantes, ayuda al profesor a seguir al alumno en su aprendizaje mediante la detección de las dificultades de aprendizaje de los estudiantes. Adicionalmente, gracias a técnicas de interpretación gráfica, el sistema proporcionará al profesor una manera de interpretar estos modelos, permitiendo visualizar las habilidades logradas y no logradas en la interacción del alumno con el sistema.

La propuesta metodológica así como su implementación, ha sido evaluada en tres institutos de Madrid. En esta tesis se recogen los resultados obtenidos y las líneas de trabajo futuro que se abren a continuación.

Abstract

The continuous advances in information technology has naturally influenced the use of technological tools in the classroom.

Students' minds are further stimulated through the activation of modern communication channels and the development of digital skills that can be used in different real-world contexts.

Educational robotics is a useful tool to promote cognitive processes because students are very motivated when interacting with a robot. Furthermore, robotics as an educational resource allows the natural development of knowledge of Science, Technology, Engineering and Mathematics (STEM).

In this thesis work, a learning environment called IDEE is presented for the study of physics in secondary education. The robotic component is proposed to be used in experimental activities. The goal of IDEE is to allow the student: to reproduce the proposed problem, think possible solutions, see the results of his actions, learn the didactic content and create relations with the discussed contents.

The student, guided by the teacher, learns by interacting with the robot and with the system, trying to find solutions to the proposed problems. When solving a problem, the student builds his knowledge by questioning about the possible steps to follow, making errors and trying again.

In this work, the evaluation of the student is a fundamental tool for the teacher; evaluation does not only mean to assign a number to find out the students who passed, but it also implies to find where the knowledge of the students is lacking in order to focus their learning.

In IDEE educational data mining models are used to analyze students' interaction data. The results are graphically shown to the teacher. The goal is to help the teacher to follow the learning process of the students to detect their learning difficulties.

The methodological proposal as well as its implementation in IDEE has been tested in three institutes in Madrid. The obtained results and the identified lines of future work are also described in this thesis.

Índice general

Resumen	I
Abstract	III
1. Introducción	2
1.1. Motivación	2
1.2. Objetivos	3
1.3. Contribuciones más relevantes	4
1.4. Estructura de la memoria	6
2. Estado del arte	7
2.1. Visión general	7
2.2. Los retos educativos del siglo XXI	8
2.3. Marco legislativo	11
2.4. Robótica educativa	11
2.4.1. LEGO Mindstorms y la robótica educativa	12
2.4.2. Instrumentos de programación en LEGO Mindstorms EV3	15
2.5. Entornos de programación	17
2.6. Minería de datos educativos	18
3. Fundamentación pedagógica	21
3.1. Introducción	21
3.2. Marco pedagógico	23
3.3. Orientaciones generales para el diseño de actividades en IDEE	23
3.4. Soporte para el profesor y el alumno	29
4. Desarrollo de IDEE	31
4.1. Visión general	31
4.2. Descripción de la Arquitectura	34
4.3. Implementación	35
4.3.1. Paradigma MTV en IDEE	37
4.3.2. Módulo del profesor	43

4.4.	Programación del robot para cada experiencia	43
4.4.1.	Área de programación en IDEE	44
4.4.2.	Creación de bloques adicionales en IDEE	46
4.4.3.	Bloques de comunicación entre el sistema y el robot	47
4.5.	Desarrollo de la interfaz del módulo del profesor	49
4.5.1.	Gráficos interactivos	49
4.6.	Diseño de actividades y contenidos	50
5.	Análisis del aprendizaje de los estudiantes en IDEE	54
5.1.	Obtención del nivel de conocimiento de los alumnos	54
5.2.	Análisis de los datos con AFM	57
5.3.	Panel del profesor	59
5.4.	Análisis de datos con CKT	63
5.5.	Adaptación de los modelos estadísticos al modelo cognitivo de IDEE	65
6.	Evaluación	69
6.1.	Objetivos	69
6.2.	Experimentación en clase con IDEE	70
6.2.1.	Evaluación de las experiencias en clase	71
6.2.2.	Análisis del aprendizaje en IDEE	73
6.3.	Valoración	81
7.	Conclusiones y trabajos futuros	83
7.1.	Conclusiones y principales aportaciones	83
7.2.	Trabajos futuros	85
7.3.	Publicaciones	86
	Bibliografía	87
A.	Códigos destacables de la aplicación	94
A.1.	Uso de Django en la creación de la base de datos	94
A.2.	Creación de actividades	95
A.3.	Creación de los bloques de comunicación entre el robot y el sistema	99
A.4.	Determinación del conocimiento del alumno	101
A.5.	Implementación de los modelos de minería de datos	102
B.	Documentos para la evaluación	109
B.1.	Pre-test presentado a los estudiantes	109
B.2.	Post-test presentado a los estudiantes	111
C.	Construcción del plano robótico con LEGO EV3	114

Índice de figuras

1.1. Niveles sobre los que se ha trabajado para la consecución de los objetivos propuestos.	5
2.1. Robótica educativa en las distintas etapas escolares	13
2.2. Juego de baloncesto programable en Scratch3 donde la pelota se mueve gracias al sensor de proximidad del kit Lego EV3	14
2.3. Ejemplo de programación del robot EV3 en Swift Playgrounds. . .	14
2.4. Resumen de los bloques del entorno de programación Lego EV3 education	16
2.5. Programa desarrollado en el nuevo entorno Lego EV3 Classroom.	16
3.1. Las actividades en IDEE, crean un puente entre el conocimiento concreto y el conocimiento abstracto teórico.	22
3.2. Skills contenidos en IDEE para conseguir los contenidos de la programación de la unidad didáctica del estudio cinemático del movimiento.	25
3.4. Imágenes de los robots utilizados en las actividades de laboratorio.	29
3.5. En la gráfica [31] se muestran las distintas áreas de investigación en minería de datos educativa.	30
4.1. Interfaz del estudiante en la actividad de laboratorio del plano inclinado.	32
4.2. Ejemplo de corrección en IDEE.	33
4.3. Modelo del sistema IDEE.	34
4.4. Arquitectura de IDEE	35
4.5. Descripción de la colaboración entre capas MTV	37
4.6. Descripción de la organización MTV del proyecto <i>web.learning</i> para la creación de IDEE	38
4.7. La imagen muestra una parte de la base de datos IDEE que incluye la estructura de las experiencias y la relación con skills y <i>score</i> . . .	39

4.8. La imagen muestra una parte de la base de datos de IDEE que incluye los modelos que permiten seguir el aprendizaje del alumno: clase <i>task</i> , clase <i>data</i> y clase <i>profile</i>	40
4.9. Diagrama de asociación UML que muestra todos los modelos creados en IDEE.	41
4.10. Vistas del proyecto <i>web.learning</i> para la creación de IDEE.	42
4.11. Plantillas del proyecto <i>web.learning</i> para la creación de IDEE.	42
4.12. Página web que permite al profesor modificar o introducir un problema en IDEE.	43
4.13. Página de trabajo para la creación de nuevos bloques.	45
4.14. Area de programación de una actividad genérica en IDEE	45
4.15. Bloques que se pueden incluir en el toolbox de las actividades de IDEE	46
4.16. Ejemplo de creación del bloque de input del sensor de contacto. En rojo un ejemplo de programación de la gráfica del bloque y en azul el código generador.	47
4.17. Ejemplo de uso de los bloques de comunicación.	49
4.18. Gráficos útiles al docente para comprender el proceso de aprendizaje de los alumnos.	50
4.19. Portal de entrada a IDEE	51
4.20. Ejemplo de actividades en IDEE.	51
5.1. Ejemplo de las respuestas de un estudiante y su seguimiento en la base de datos de IDEE	56
5.2. Área de trabajo de los estudiantes con un bajo nivel de conocimiento y las ayudas aportadas por el sistema	57
5.3. Visualización gráfica del parámetro α . Los puntos rojos representan alumnos con dificultades de aprendizaje y los puntos verde representan los alumnos significativos con buenas capacidades de aprendizaje, en los skills analizados.	60
5.4. Interfaz del docente que muestra las curvas de aprendizaje al profesor. Sobre el skill mostrado el grupo clase mejora en el aprendizaje.	60
5.5. Interfaz del docente que muestra el conocimiento acumulado de los alumnos con la visualización del <i>accumulated score</i>	61
5.6. El estudiante_1532 ha trabajado sobre el skill_12 seis veces en dos actividades del tipo problema y todas las respuestas han sido incorrectas (marcadas en rojo por el sistema)	62

5.7. Imagen del módulo del docente in IDEE. El profesor tiene acceso a diferentes informaciones utilizando el menú: (1) indicadores de aprendizaje; (2) curvas de aprendizaje, (3) gráficos y (4) tablas . . .	62
5.8. Representación del modelo tradicional CKT con descripciones de los parámetros y de los nodos.	64
5.9. Ejemplo de la Q-matriz utilizada en el AFM.	66
5.10. Ejemplo de Q-matriz utilizada en el AFM mejorado (la columna <i>Type</i> indica si la actividad es un test o un problema).	66
6.1. El gráfico muestra el porcentaje de error de los alumnos en su conocimiento inicial.	72
6.2. El gráfico muestra el porcentaje de error de los estudiantes en los diferentes temas evaluados en el pre-test.	72
6.3. El gráfico muestra el porcentaje de error en el post-test para los dos grupos.	72
6.4. El gráfico muestra la diferencia entre las medias de los dos grupos sobre el número de errores.	72
6.5. El gráfico muestra el interés del grupo clase en la asignatura de física.	73
6.6. Gráfico para determinar el grado de cumplimiento de los objetivos propuestos en esta experiencia. El alumno puede expresar su opinión respondiendo con un número entre 1 y 5.	73
6.7. Curva de aprendizaje skill_1.	76
6.8. Curva de aprendizaje skill_3.	76
6.9. Curva de aprendizaje skill_7.	76
6.10. Curva de aprendizaje skill_10.	76
6.11. Curvas de aprendizaje	76
6.12. Resumen de las actividades realizadas en IDEE por el alumno con ID 779.	76
6.13. Análisis gráfico de la puntuación del alumno en skill_3.	76
6.14. Análisis gráfico del parámetro de intercepción α del estudiante con un gráfico Polar	77
6.15. Visualización gráfica del parámetro α . Los puntos rojos representan alumnos con dificultades de aprendizaje y los puntos verdes representan los alumnos significativos con buenas capacidades de aprendizaje en los skills analizados.	80
6.16. Interfaz del docente que muestra el conocimiento acumulado de los alumnos con la visualización del <i>accumulated score</i>	80

6.17. El alumno ha trabajado en skill_12 seis veces en dos actividades tipo problema y todas las respuestas han sido incorrectas (todas están marcadas en rojo).	81
6.18. Pantalla del módulo del profesor que muestra la curva de aprendizaje para el skill_12. En esta habilidad, el alumno mejora el aprendizaje.	81
7.1. Pasos para la realización del proyecto de tesis	84
A.1. Diagrama de las funciones utilizadas para convertir los datos cuantitativos de la base de datos útiles para la aplicación de los métodos de minería de datos.	102
C.1. Plano robótico con LEGO EV3.	115
C.2. Detalles plano robótico con LEGO EV3.	116

Índice de tablas

5.1. Skills definidos en el laboratorio del plano inclinado en IDEE.	55
5.2. Ejemplo de actividades de dos estudiantes en tres actividades. Las columnas skill_1 y skill_2 muestran el valor del parámetro score de cada skill en las actividades. Las columnas student_1 y student_2 muestran si cada estudiante ha trabajado o no en cada actividad.	66
5.3. Ejemplo de desarrollo del aprendizaje en un alumno que trabaja en cuatro actividades con los modelos CKT y eCKT.	67
6.1. Indicadores del proceso de aprendizaje. La intersección de habilidades es el nivel de dificultad inicial para cada habilidad, la pendiente es la tasa de aprendizaje y la probabilidad es la probabilidad de éxito de usar una habilidad una vez que la experiencia ha terminado.	75
6.2. Resultados del modelo eCKT para el estudiante con ID 779	77
6.3. Indicadores del proceso de aprendizaje.	78

Capítulo 1

Introducción

En este capítulo se presenta el marco en el que se sitúa la presente tesis, y que constituye el referente básico de su desarrollo.

Se iniciará con la motivación y los objetivos buscados. Posteriormente se describirá la estructura de la tesis, comentando el proceso de trabajo desarrollado.

1.1. Motivación

Los estudiantes de hoy en día, por estar rodeados e inmersos constantemente en las nuevas tecnologías, piensan y procesan la información de forma diferente a las generaciones anteriores. Como consumidores y productores de contenido digital y por sus nuevas formas de comunicarse, han provocado una nueva brecha alfabeto-generacional con las generaciones anteriores [49].

Hoy en día, la mayoría de los profesores pertenecen a la generación de los llamados *inmigrantes digitales*¹, por lo que enseñan a través de un tipo de lenguaje poco atractivo para los alumnos que pertenecen a la generación de *nativos digitales*². Es necesario dotar a los profesionales de la educación de herramientas que les acompañe en el camino al uso de nuevas tecnologías, de forma progresiva y mediante intervenciones que permitan la adquisición de habilidades y conocimientos adecuados para enfrentar los nuevos desafíos educativos.

La introducción de nuevas tecnologías en el aula, requiere un modo de enseñanza activo en el que el profesor guíe a los estudiantes en su aprendizaje, los apoye cuando se detienen y al mismo tiempo les permita seguir sus propios tiempos de asimilación de los contenidos. Uno de los objetivos que los profesores del

¹Prensky define “inmigrantes digitales” las personas entre 35 y 55 años, nacidas en la era pre-digital, antes de los años 90, que han tenido que aprender a usar los medios digitales proviniendo de un mundo analógico de información.

²Prensky describe a los nativos digitales como las personas nacidas después del 1980 que, rodeadas desde temprana edad por las nuevas tecnologías desarrollan otra manera de pensar y de entender el mundo.

siglo XXI deberían proponerse, es ayudar al alumno a pensar por sí mismo. Para conseguir este objetivo, el profesor debe impulsar en los alumnos el pensamiento crítico; entendiendo éste como la capacidad de identificar, analizar, evaluar, clasificar e interpretar lo que está a nuestro alrededor.

La robótica educativa permite desarrollar un aprendizaje interdisciplinar, adquiriendo nociones sobre el uso de la tecnología y la programación. Mediante la programación, los alumnos tienen que elegir la mejor de entre las estrategias posibles, estructurando la resolución del problema en pasos lógicos; esto le permite el desarrollo del pensamiento computacional y al mismo tiempo incentiva el pensamiento abstracto y el pensamiento crítico [37]. El uso de herramientas de robótica educativa en el aula, crea un clima de colaboración entre los alumnos y hacia el profesor que favorece la motivación personal, la resolución de conflictos y el aprendizaje.

El sistema educativo debe transformarse en un nuevo espacio abierto de aprendizaje que considere natural el entorno digital, avanzando hacia una nueva forma de enseñanza.

Sin embargo, la introducción en la clase tradicional del uso de nuevas tecnologías no es fácil, ya que abre nuevos desafíos que deben abordarse y resolverse para ayudar al profesor a controlar estos nuevos entornos y aprovechar sus ventajas.

En un primer análisis identificamos las siguientes dificultades:

- las aulas se transforman en ambientes de aprendizaje no estructurados y el profesor no se siente seguro en la gestión del aula.
- El profesor no puede analizar el trabajo de los participantes y por tanto, le resulta difícil guiarlos en su aprendizaje.
- Las herramientas tecnológicas no son fáciles de usar para el profesor y las actividades a menudo no están especialmente diseñadas para un contexto educativo.
- El profesor tiene dificultades a la hora de evaluar las actividades que los alumnos han realizado en entornos tecnológicos porque no dispone de información sobre el recorrido del aprendizaje de cada alumno.

1.2. **Objetivos**

A partir de la problemática expuesta en el apartado anterior se justifica la necesidad de crear un nuevo espacio de aprendizaje que tenga en consideración por un lado el entorno digital y la nuevas formas de comunicar de los alumnos y

por el otro la necesidad didáctica del profesor de obtener información acerca del progreso del aprendizaje de los alumnos en clase.

Teniendo en cuenta todos los factores que intervienen en la educación, los objetivos principales de este trabajo son:

- Utilizar la experimentación con robots como parte integrante de una explicación en clase; guiada bajo el aspecto teórico y formativo por un sistema tutorial inteligente de manera que los alumnos aprendan siguiendo su propio ritmo de aprendizaje y sus propios conocimientos.
- Estructurar las actividades de una forma adecuada para el uso de herramientas robóticas en clase.
- Realizar un prototipo de un sistema para Educación Secundaria a través de Internet que sea útil tanto para el alumno como para el profesor y que pueda ser utilizado en clases presenciales para la consecución de los objetivos curriculares.

La idea es que el alumno pueda comunicarse con el robot de la forma más natural posible. Por tanto, un objetivo necesario de este trabajo es la construcción de bloques de programación para que el alumno pueda trabajar utilizando únicamente componentes lógicos y no conocimientos de programación.

- Dar un instrumento al profesor para que pueda desarrollar en clase una enseñanza activa, capaz de analizar los pasos que llevan al aprendizaje de los alumnos, sus fallos y sus incertidumbres. Para ello, el sistema incluirá un modelo del estudiante, en el que se guardará información sobre el progreso en el aprendizaje y las posibles dificultades que se haya podido encontrar. Para la obtención de este modelo se utilizarán técnicas de minería de datos educativas. El objetivo principal es determinar el progreso del aprendizaje de los estudiantes y representar dicha información de manera que se tenga en cuenta que el profesor deberá interpretar los resultados que proporcione el sistema sin conocimiento previo de minería de datos o programación.
- Utilizar en el aula el sistema creado para poder evaluar su aplicación en un contexto educativo real.

1.3. Contribuciones más relevantes

Para la realización de este proyecto de tesis se ha trabajado en distintos niveles:



Figura 1.1: Niveles sobre los que se ha trabajado para la consecución de los objetivos propuestos.

- Nivel pedagógico: definición de los modelos pedagógico y didáctico más adecuados.
- Nivel técnico informático; diseño del prototipo IDEE (siglas del nombre del sistema en inglés *Integrated Didactic Educational Environment*).
- Nivel de investigación aplicada: definición del modelo del estudiante determinando qué información se obtendrá a partir del análisis de las interacciones del alumno con el sistema.
- Nivel de investigación teórica: mejora del análisis del aprendizaje del alumno, conforme al modelo pedagógico implementado.

El objetivo principal de este trabajo es lograr un seguimiento del proceso de aprendizaje de los estudiantes en un contexto de aula que siga una modalidad didáctica inclusiva y participativa.

La implementación del prototipo ha sido necesaria para poder crear un sistema que pudiera adecuarse a la propuesta pedagógica y educativa de este trabajo. Para lograr este objetivo, se ha trabajado en mejorar la definición de los modelos teóricos que se han utilizado para analizar el aprendizaje de los estudiantes. De esta manera se busca obtener una evaluación del estudiante más detallada.

En la Figura 1.1, se sintetizan los distintos niveles sobre los que se ha trabajado para el consecución de los objetivos de esta tesis.

1.4. Estructura de la memoria

La motivación y los objetivos de este trabajo se han descrito en este capítulo 1 de introducción. El capítulo 2 incluye una revisión del estado del arte de las áreas implicadas en la realización de esta tesis. En el capítulo 3 se enmarca pedagógicamente el trabajo y se describe cómo las actividades didácticas propuestas se ajustan al modelo pedagógico definido. El capítulo 4 se centra en la descripción del sistema IDEE e incluye la descripción de su arquitectura y las herramientas utilizadas para su desarrollo. El capítulo 5 describe el modelo de estudiante que se ha definido y las técnicas de minería de datos utilizadas para analizar el aprendizaje de los estudiantes. Así mismo, se describen las herramientas utilizadas para que el profesor pueda interpretar los resultados del modelo de forma fácil.

El capítulo 6 describe las experiencias llevadas a cabo en dos institutos de Madrid y los resultados obtenidos a partir del análisis de los datos recogidos. El capítulo 7 describe las conclusiones y las líneas de trabajo futuro que se abren a continuación. Por último, se han incluido cuatro apéndices adicionales para describir el código más significativo en el desarrollo de IDEE (apéndice 1); los datos utilizados en la evaluación (apéndice 2); un manual de uso de IDEE (apéndice 3) y las instrucciones de construcción del robot utilizado en la experimentación (apéndice 4).

Capítulo 2

Estado del arte

Hoy en día se ha demostrado la utilidad de la introducción de instrumentos robóticos en clase para ayudar a los alumnos en el desarrollo del pensamiento crítico. En este capítulo se revisará la situación actual del uso de estas tecnologías en el aula.

Así, en este capítulo se proporcionará: i) una visión general de los entornos de aprendizaje que se han ido desarrollando para permitir el uso de instrumentos robóticos en contextos educativos; ii) una descripción de los avances tecnológicos de los kit robóticos educativos LEGO MINDSTORMS [®] EV3 y iii) un análisis de las técnicas de minería de datos que pueden ser útiles en el contexto educativo para comprender el proceso de aprendizaje de los alumnos.

2.1. Visión general

Los nuevos desarrollos en robótica educativa están haciendo posible más que nunca, que los profesores utilicen instrumentos tecnológicos en el aula, principalmente en las asignaturas enmarcadas en el área de Ciencia, Tecnología, Ingeniería y Matemáticas (STEM) [9, 29, 40, 48].

Con el rápido desarrollo de la tecnología informática, la importancia del pensamiento computacional ha sido reconocida en los sistemas educativos actuales. De hecho, las nuevas leyes educativas introducen las nuevas tecnologías en el contexto educativo español con el objetivo de que los estudiantes obtengan un conocimiento básico de programación y robótica(T.P.R).

Aunque varios estudios confirman resultados positivos de plantear actividades experimentales [8, 60], la robótica educativa sigue siendo una actividad desarrollada como extracurricular, generalmente aislada del plan de estudios oficial. Para poder utilizar de manera eficaz herramientas de robótica en entornos de aprendizaje en el aula, es importante que el profesor pueda identificar sobre qué conceptos los estudiantes tienen más dificultades en el aprendizaje. Para ello, es necesario

apoyar a los profesores, proporcionando herramientas para el seguimiento del aprendizaje de los estudiantes [4, 26, 33].

2.2. Los retos educativos del siglo XXI

Los requisitos del sistema educativo del siglo XXI han cambiado. Ahora se busca dotar a los alumnos de una educación personalizada (adaptada y centrada en cada alumno) donde cada estudiante disponga de un entorno personal con herramientas que le permitan gestionar sus avances y comunicarse con otros durante el proceso. Es necesario un nuevo tipo de enseñanza, basado en metodologías activas (construccionista, para “aprender haciendo”), que utilice las tecnologías como herramientas y una evaluación auténtica, para el aprendizaje.

La educación en el aula, debe tener en cuenta las diferentes formas de inteligencia expresadas por Gardner ¹ y, en consecuencia, las diferencias en la forma de aprender que estas formas de inteligencia determinan [21].

Dos aspectos importantes de la teoría de la inteligencia múltiple son que todos los estudiantes tienen:

- diferentes inteligencias gracias a las cuales aprenden. Por lo tanto, cada profesor, puede elegir enseñar abordando inteligencias específicas, desarrollándolas y teniendo en cuenta su existencia en el uso de materiales educativos significativos;
- un perfil de inteligencia diferente, ya que no todos tienen las mismas experiencias de vida.

Todo esto se traduce, en la enseñanza, en un enfoque dirigido a mejorar el potencial de cada estudiante, identificable a través de una observación sistemática y realizado con criterios y herramientas científicamente validados.

La existencia de diferencias individuales requiere que el profesor use una variedad de estrategias de enseñanza. De esta manera, al alternar formas de tratar con el contenido, siempre habrá un momento en que la actividad en el aula involucrará las inteligencias más desarrolladas de cada alumno. En esta perspectiva, los profesores están llamados a enriquecer su repertorio con una amplia gama de métodos, materiales y estrategias para motivar a los estudiantes en clases cada vez más heterogéneas.

El construccionismo es una variante del constructivismo que comparte con este último la idea de que el alumno construye estructuras de conocimiento, pero

¹Gardner introdujo la llamada Teoría de las Inteligencias Múltiples en el mundo científico y académico, según la cual no existe una facultad común de inteligencia, pero distintas formas de ella, cada una independiente de las demás.

también especifica que la construcción del conocimiento es mucho más significativa en un contexto donde el sujeto que aprende se dedica a la construcción de algo concreto que se pueda compartir.

Las actividades de aprendizaje basadas en el construccionismo se han estudiado ampliamente en la educación formal e informal [44].

Los propósitos de un entorno de aprendizaje construccionista son los siguientes:

- favorecer la construcción del conocimiento, no su reproducción;
- preferir un tipo de aprendizaje basado en el estudio de caso;
- contextualizar siempre el conocimiento que se está estudiando;
- limitar el uso de secuencias de instrucción predeterminadas;
- ofrecer múltiples puntos de vista de la realidad;
- promover el aprendizaje cooperativo.

Las técnicas para alcanzar estos objetivos son algunas herramientas mentales típicas de las Tecnologías de la Información y la Comunicación (TIC), a saber, hojas de cálculo, kits robóticos, bases de datos, hipermedia y otros programas que permiten al alumno llevar a cabo análisis, evaluaciones, síntesis, resolución de problemas y reflexiones destinadas a construir su propia visión de la realidad. Esto no siempre debe implicar necesariamente considerar únicamente el autoaprendizaje: se trata de ofrecer una experiencia de aprendizaje entendida como un reajuste flexible del conocimiento preexistente de acuerdo con las necesidades de una nueva situación de capacitación cada vez. Además, desde un punto de vista construccionista, el trabajo en grupo y el aprendizaje cooperativo permiten aumentar la capacidad del individuo para encontrar soluciones eficientes, mejorando así sus habilidades para resolver problemas y establecer, de hecho, una actitud construccionista en el aula.

Uno de los principales exponentes, especialmente por las contribuciones proporcionadas a las tecnologías de enseñanza y aprendizaje, es Seymour Papert, colaborador de Piaget (de 1958 a 1963) en el instituto de epistemología genética en Ginebra, y desde 1964, investigador en el Instituto de Tecnología de Massachusetts (MIT) en Boston, donde fundó el Laboratorio de Inteligencia Artificial junto con Marvin Minsky.

Según Papert, el construccionismo es una palabra con dos aspectos. Uno se refiere a la teoría constructivista de Piaget que considera el aprendizaje como una reconstrucción y no como una mera transmisión de conocimiento. El otro extiende

el concepto de materiales manipulativos afirmando que la construcción y, por lo tanto, el aprendizaje, es más eficaz y completo cuando no solo es mental, sino que está respaldado por una construcción real, por una actividad como la construcción de un proyecto significativo [46]. Las ideas de Seymour Papert manifestadas en su libro “Mindstorms: Children, Computers and Powerful Ideas (1980)” [45] se convirtieron en los principios de los modelos de aprendizaje de las llamadas CITAM o STEAM (Ciencia, Ingeniería, Tecnología, Arte y Matemáticas).

De hecho, Seymour Papert fue uno de los primeros en identificar el potencial revolucionario de los ordenadores en la educación. El área favorecida por Papert fue la creación de entornos de aprendizaje que utilizan tecnologías que hacen que el estudiante y las actividades concretas que generan aprendizaje, se vuelvan centrales. Con esa idea, en 1963 Papert creó “LOGO”, un lenguaje y entorno de programación especialmente pensado para niños. Posteriormente desarrolló la extensión del conjunto de construcción LEGO® a un conjunto de robótica para poner a disposición de los niños herramientas para concretar el pensamiento abstracto y herramientas para crear y explorar incluso criaturas artificiales. Los niños usaban LOGO para programar los movimientos de una *tortuga* ya sea en la forma de un pequeño robot mecánico o un objeto gráfico en la pantalla del ordenador.

Existen fuertes argumentos para que los jóvenes aprendan a codificar con el apoyo del enfoque constructorista; la integración de la codificación en contextos pedagógicos mejora la lógica, el pensamiento crítico, la resolución de problemas y las matemáticas [27]. Los jóvenes necesitan adquirir habilidades del siglo XXI, fortaleciéndose con las competencias requeridas relacionadas con la digitalización de nuestra sociedad.

Así, uno de los lenguajes de programación por bloques más utilizados en entornos educativos, Scratch, se basa en la idea constructorista de LOGO; con Scratch3 ² se incluye la posibilidad de conectar robots y programarlos. También las plataformas: Swift Playgrounds³, Blockly ⁴ o App Inventor ⁵ fomentan el aprendizaje según la metodología constructorista. Estos entornos no requieren ninguna experiencia especial, pero requieren un pensamiento crítico que permita al usuario indicar a la computadora qué debe hacer.

²<https://scratch.mit.edu>

³<https://www.apple.com/swift/playgrounds/>

⁴<https://developers.google.com/blockly>

⁵<https://appinventor.mit.edu>

2.3. Marco legislativo

Hoy en día, las leyes en educación incluyen disposiciones que establecen la introducción de la robótica en los programas educativos. En la Ley Orgánica 2/2006, de 3 de mayo, de Educación (L.O.E.) según el REAL DECRETO 1631/2006, de 29 de diciembre, se establecen las enseñanzas en la educación española. Con la nueva Ley Orgánica 8/2013, de 9 de diciembre, para la mejora de la calidad educativa (LOMCE) se promueve, entre otros cambios importantes, la ciencia y las nuevas herramientas tecnológicas como competencias necesarias a incluir en los programas educativos españoles.

A modo de ejemplo, los contenidos que se incluyen en el cuarto curso en educación secundaria en la materia de Tecnología son:

- Experimentación con sistemas automáticos, sensores, actuadores y aplicación de la reglamentación en dispositivos de control.
- Diseño y construcción de robots.
- Uso del ordenador como elemento de programación y control. Trabajo con simuladores informáticos para verificar y comprobar el funcionamiento de los sistemas diseñados.

También se han incrementado las actividades extraescolares de robótica con el objetivo de desarrollar las competencias y habilidades necesarias para poder solucionar problemas. La idea es acercar a los niños a la resolución de problemas de forma lúdica, incorporando la componente de programación. Los niños aprenden jugando a comprender problemas, a encontrar posibles soluciones con las herramienta a su disposición y a solucionar eventuales errores. La programación les permite desarrollar temas de control, de funcionamiento autónomo y de percepción remota. Claramente los contenidos y los objetivos deben estar adaptados a cada franja de edad según las capacidades de los niños.

2.4. Robótica educativa

El interés sobre el uso de robots en la escuela ha ido aumentando desde que Seymour Papert presentara el lenguaje de programación LOGO y su famosa tortuga en los años 60. La robótica educativa está demostrando ser un instrumento innovador en el aula.

Podemos hablar de tres paradigmas de aprendizaje relacionados con la robótica educativa [22]:

- *aprender robótica*; aquel en el que los estudiantes usan los robots como plataforma para aprender conocimientos propios de ingeniería, relacionados con mecánica, electrónica o programación;
- *aprender con la robótica*; cuando los robots son utilizados como soporte en el proceso de aprendizaje;
- *aprender de la robótica*; cuando a través de la robótica, los estudiantes aprenden contenidos y destrezas de diferentes disciplinas, a la vez que desarrollan competencias transversales.

En este último enfoque, también conocido como *robotic-based instruction* en la literatura, el robot actúa como herramienta activa intermediaria entre estudiante, profesor y materia escolar. En este enfoque las tareas basadas en dichos entornos permiten el desarrollo y adquisición de competencias por parte del alumno, no solo referentes a ámbitos técnicos, sino también a otras áreas. Por ejemplo en [39] se usa la robótica educativa para apoyar la enseñanza de la lengua materna, proponiendo dos actividades con LEGO MINDSTORMS® EV3 que puede realizar el profesor. En [25] los autores, proponen el uso de Bee-Bots® en una unidad de un curso sobre seguridad vial.

La robótica educativa se ha utilizado también en la educación en ciencias. Por ejemplo, se ha utilizado un robot social educativo para motivar a los estudiantes a leer textos de ciencias [38]. En [18] los autores proponen el uso de un robot MARRtino para enseñar conceptos Newtonianos. En [7] se les pide a los estudiantes que construyan y usen un robot rescatador para aplicar los conocimientos sobre electricidad. Todos estos enfoques evalúan el impacto del uso de la robótica educativa comparando el nivel de conocimiento de los estudiantes antes y después de la experiencia en robótica.

En la Figura 2.1 se pueden observar distintos robots que hoy en día se vienen utilizando en las distintas etapas escolares.

2.4.1. LEGO Mindstorms y la robótica educativa

En 1998, LEGO ®⁶ presentó de manera comercial su primer robot programable de la serie Mindstorms (el ladrillo RCX). Su salida al mercado se produjo tras bastantes años de trabajo con el MIT, institución a la que LEGO financiaba investigaciones acerca de cómo aprenden los niños. A cambio, la empresa de juguetes obtenía ideas para sus futuros productos en estos campos.

Desde entonces, la serie Mindstorms ha tenido tres generaciones, la última, LEGO Mindstorms Education EV3, presentada hace ya más de 7 años.

⁶Por legibilidad, en el resto del trabajo se omitirá el símbolo de marca registrada.

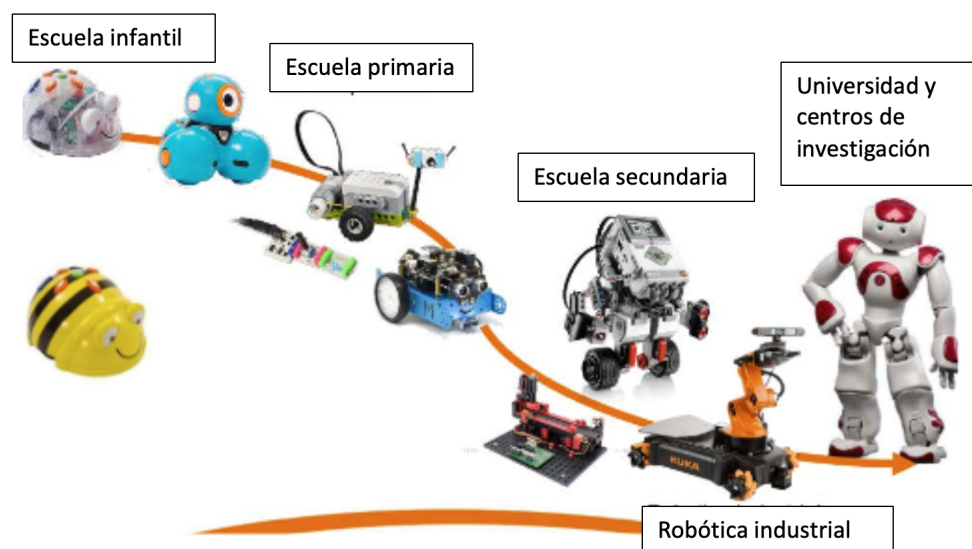


Figura 2.1: Robótica educativa en las distintas etapas escolares

En todas las versiones de robots de LEGO la idea de funcionamiento es la misma: los robots se construyen utilizando los bloques LEGO, se añaden ruedas y otras piezas específicas además de sensores y actuadores. Estos sensores y actuadores se controlan con un mini computador, el llamado *ladrillo*, que se debe programar usando un lenguaje visual basado en iconos. Estos robots no son asequibles a nivel doméstico pero han tenido muy buena acogida en entornos educativos ya que aúnan calidad y permiten crear un proyecto educativo completo.

Con el auge de Scratch y Python, se permitió la posibilidad de programar el ladrillo utilizando directamente Scratch 3.0 o instalando MicroPython en una tarjeta SD.

En la Figura 2.2 se visualiza la interfaz de Scratch3 para poder programar un juego de baloncesto, donde la pelota se mueve gracias al sensor de proximidad del kit lego EV3.

La programación de los robots Mindstorms EV3 puede realizarse desde diferentes sistemas operativos y dispositivos y está basada en iconos y programación por bloques. Como en otros lenguajes visuales como Scratch, la programación se vale de códigos de colores para la identificación rápida de funciones y modos de uso.

Más recientemente, en noviembre de 2019, LEGO lanzó una nueva versión del entorno de programación EV3 llamada LEGO Mindstorms Education EV3 Classroom que utiliza un lenguaje de codificación basado en Scratch.

El kit LEGO Mindstorms Education EV3 es ampliamente utilizado en contextos educativos [57] debido a su versatilidad, popularidad entre los estudiantes y su contenido, que incluye sensores, motores y un dispositivo programable que

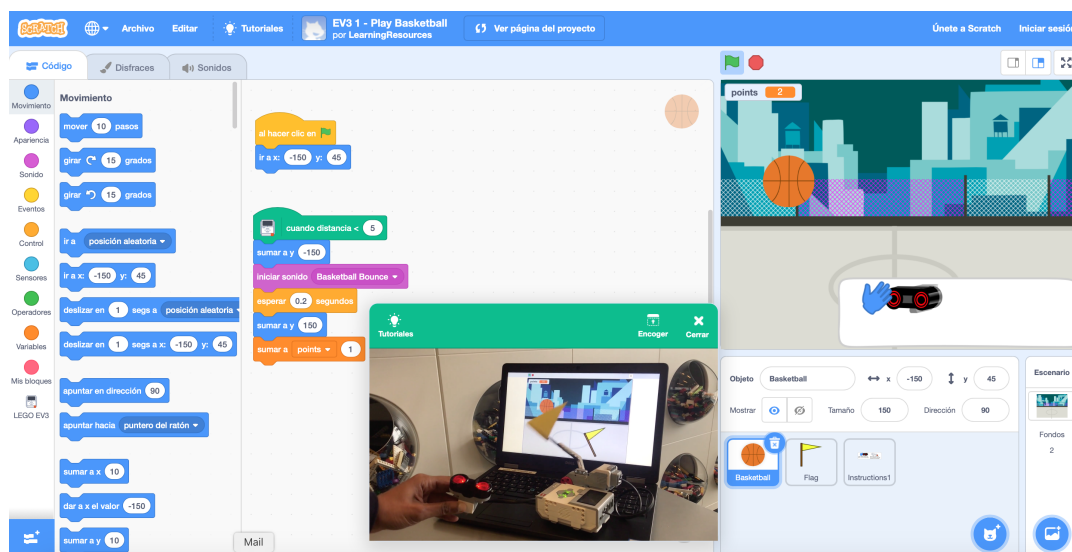


Figura 2.2: Juego de baloncesto programable en Scratch3 donde la pelota se mueve gracias al sensor de proximidad del kit Lego EV3

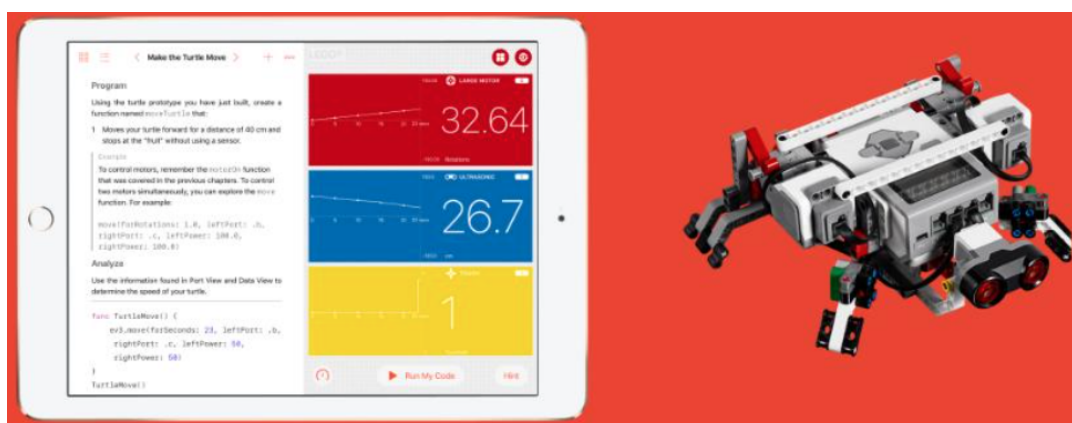


Figura 2.3: Ejemplo de programación del robot EV3 en Swift Playgrounds.

controla el movimiento y la reacción del robot [1].

También Apple con su plataforma Swift Playgrounds, el sistema ideado por Apple para que los estudiantes aprendan a programar en Swift en iPads, permite ya la programación del robot EV3 de LEGO. En la Figura 2.3 se puede ver un ejemplo de programación en Swift Playgrounds.

El kit robótico LEGO Mindstorms EV3 ofrece las siguientes ventajas a la hora de trabajar en contextos educativos:

- todos los alumnos tienen familiaridad con las construcciones, así que este tipo de herramienta le resultará familiar.
- El kit resulta resistente y compacto y no requiere conocimientos eléctricos ni electrónicos, los alumnos lo pueden manejar fácilmente.
- El Kit LEGO permite conseguir varios niveles de complejidad, dando la

posibilidad al profesor y a los alumnos de incrementar sus conocimientos y sus aplicaciones.

- Cada alumno puede construir el modelo que le guste, razonando sobre las características necesarias para poder desarrollar el trabajo; no hay límite bajo el aspecto de la construcción.
- El kit LEGO Mindstorms EV3 es programable con un lenguaje propio basado en bloques, con Scratch3 y con Python; esto permite desarrollar en los estudiantes diferentes niveles de contenidos de programación con el mismo kit educativo.
- LEGO Mindstorms ha sido creado siguiendo las ideas pedagógicas del constructivismo de Papert.

2.4.2. Instrumentos de programación en LEGO Mindstorms EV3

Los bloques de programación que se utilizan en el entorno de programación LEGO EV3 se dividen en categorías según el tipo, lo que facilita encontrar el bloque que se necesita en cada operación. A los bloques usuales, presentes en cualquier lenguaje de programación (bloques de gestión de datos, bloques matemáticos y bloques de control de flujo), se añaden los bloques para gestionar el robot y bloques más avanzados. Por ejemplo, aquellos que añaden la posibilidad de interacción entre diferentes robots EV3, la visualización gráfica del análisis de datos o la creación de bloques propios para ser utilizados en futuros programas dentro del mismo proyecto.

En la Figura 2.4 se muestra un resumen de los bloques del entorno de programación LEGO EV3.

En la nueva versión LEGO Mindstorms Education EV3 *Classroom* todos los bloques habituales están disponibles salvo la posibilidad de analizar datos. Además, la conversión de los programas creados con la versión anterior al nuevo entorno de programación no es automática ya que algunos bloques necesitan ser adaptados a la nueva programación.

En la Figura 2.5 se muestra un programa desarrollado en el nuevo entorno LEGO MINDSTORMS Education EV3 Classroom.

El trabajo de actualización del entorno de programación LEGO y las múltiples posibilidades de programación de los robots LEGO, ofrece muchas potencialidades en el contexto educativo.

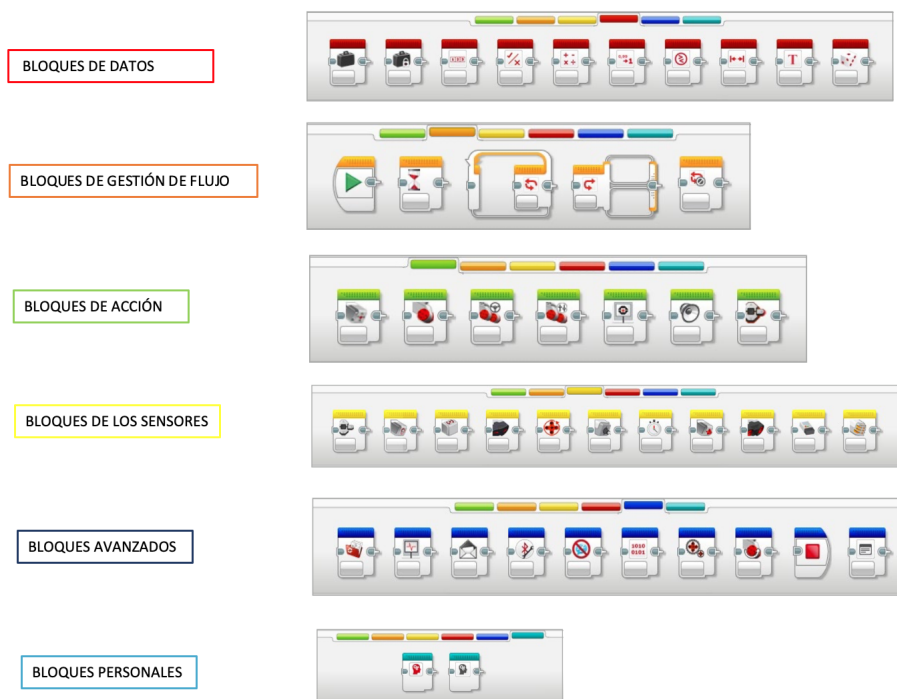


Figura 2.4: Resumen de los bloques del entorno de programación Lego EV3 education

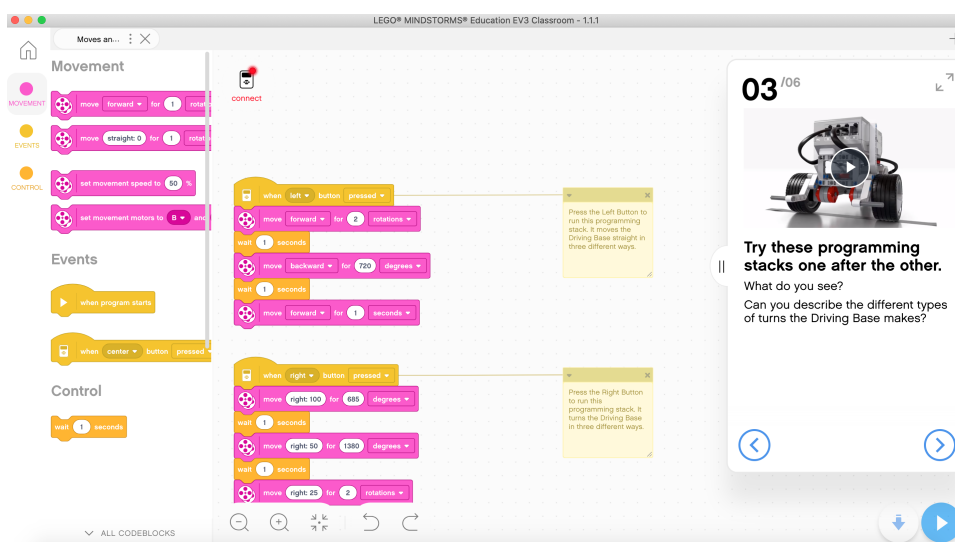


Figura 2.5: Programa desarrollado en el nuevo entorno Lego EV3 Classroom.

2.5. Entornos de programación

Para hacer que el procedimiento de programación sea más amigable y agradable para los estudiantes, se han ido desarrollando distintos entornos para el aprendizaje de programación. Por ejemplo, en [55] se presenta cómo enseñar competencias de programación a través del uso del lenguaje Visual Basic. En mayo de 2007 se lanzó Scratch [6], mencionado en la sección anterior con la idea de proporcionar un nuevo entorno más social y experimental que otros entornos de programación.

En [20] se presenta Snap!, un entorno de aprendizaje para la materia de Introducción a la Computación. Los autores proponen un lenguaje de programación de bloques basado en Scratch. Otra propuesta para ayudar a los estudiantes en la programación con Scratch, es el entorno de aprendizaje iSnap [50]. iSnap implementa un mecanismo de ayuda a los estudiantes basados en pistas que se dan cuando se detecta que los estudiantes están teniendo problemas. Por su parte, el laboratorio abierto Roberta Fraunhofer IAIS [28] ofrece una plataforma para permitir el uso de diferentes robots: LEGO Mindstorms, NAO y otros sistemas de hardware programables, como Arduino, BBC micro: bit, Calliope mini. Py-BoKids [59] es un proyecto para enseñar robótica en la escuela secundaria que se basa en el lenguaje de programación Python y donde los robots usan un microprocesador Arduino. Incluye una infraestructura de software y una colección de ejercicios prácticos dirigidos a estudiantes de secundaria. En [2] se utiliza un robot LEGO Mindstorms para proporcionar comentarios y estímulo en las tareas de programación, para conseguir así que los estudiantes reflexionen más sobre sus conceptos erróneos y persistan más en la actividad.

Sin embargo, debido a la naturaleza impredecible de una clase que usa robótica donde los estudiantes pueden interactuar de diferentes maneras [52], hay pocas propuestas para apoyar a los profesores en el seguimiento de los alumnos que utilizan instrumentos robóticos [24, 62].

La digitalización de los procesos didácticos, permite obtener datos significativos sobre la forma en que los estudiantes llevan a cabo las actividades propuestas por el docente, y, de esta manera, contribuir en el avance de la evaluación de la calidad de los procesos educativos proporcionados. El área de investigación de minería de datos educativos (sus siglas EDM vienen del término inglés *Educational Data Mining*) ofrece métodos para analizar datos de entornos de aprendizaje para comprender mejor los resultados del aprendizaje o la participación de los estudiantes [11].

2.6. Minería de datos educativos

El área de investigación de minería de datos educativos ha utilizado técnicas de aprendizaje automático para analizar los datos que se recogen de las interacciones y acciones que los estudiantes realizan en las plataformas educativas [16]. Para comprender la aplicación de estas técnicas en diferentes sistemas, a continuación se presenta una breve introducción a las diferentes técnicas de aprendizaje automático.

Los modelos de aprendizaje automático se pueden dividir en dos grandes familias: aprendizaje supervisado y aprendizaje no supervisado. En el aprendizaje supervisado cada instancia tiene asociada una etiqueta o clase que el algoritmo aprenderá a predecir mientras que en el aprendizaje no supervisado, las instancias no tienen etiqueta o clase asociada y se agrupan en función a su similitud.

Existen, a su vez, dos tipos de aprendizaje supervisado [5]:

- **Regresión:** La Regresión permite estudiar la dependencia funcional entre una variable dependiente categórica y un conjunto de variables independientes que pueden ser cuantitativas o categóricas.
- **Clasificación:** la clasificación tiene por objetivo clasificar los elementos en diferentes grupos. Cuando se utiliza un problema de clasificación, la variable dependiente es cualitativa y se corresponde a un aprendizaje supervisado, puesto que los datos se encuentran previamente clasificados.

Estos dos tipos principales de aprendizaje supervisado, clasificación y regresión, se distinguen por el tipo de variable objetivo. En los casos de clasificación, es de tipo categórico, mientras que, en los casos de regresión, la variable objetivo es de tipo numérico.

Los algoritmos más habituales que se aplican para el aprendizaje supervisado son: árboles de decisión, clasificación de Bayes, regresión por mínimos cuadrados, regresión logística, maquinas de soporte vectorial [5].

A diferencia del aprendizaje supervisado, en el no supervisado (o clustering) las instancias no están etiquetadas. Su función es la agrupación, por lo que el algoritmo agrupa por similitud. Por ejemplo, las tareas de agrupación, buscan grupos basados en similitudes, pero nada garantiza que éstas tengan algún significado o utilidad.

Como se ha comentado anteriormente, las técnicas de aprendizaje automático utilizan los datos recogidos de entornos educativos para construir modelos que predigan o caractericen determinados aspectos del proceso de aprendizaje de los estudiantes.

Hay muchas aplicaciones y tareas en entornos educativos que pueden resolverse con técnicas de minería de datos:

- mejora del modelo del estudiante y del dominio,
- predicción del comportamiento de los estudiantes,
- mejora del rendimiento de aprendizaje del estudiante,
- adaptación del curso y recomendaciones de aprendizaje basadas en los comportamientos de aprendizaje de los estudiantes.

Estudios recientes han aplicado técnicas de aprendizaje automático a los datos recopilados en actividades de programación para ayudar al profesor a seguir el aprendizaje de los alumnos. Por ejemplo, en [17] los autores muestran cómo las técnicas de aprendizaje automático pueden usarse para pronosticar el éxito de los estudiantes en un entorno de programación basado en bloques. En [34] los autores presentan cómo estimar las dificultades de los estudiantes en problemas de programación abiertos. En [10] se aplican técnicas de agrupamiento no supervisadas (métodos estadísticos de Hopkins) para identificar automáticamente las capacidades de programación de los alumnos. En [19] se utiliza el algoritmo k-means en los datos generados por los clics de los estudiantes en Scratch, con el objetivo de clasificar el aprendizaje de los alumnos en las actividades de programación. En [58] se utiliza la regresión logística para evaluar el aprendizaje de los estudiantes durante un aprendizaje con *Massive Open Online Courses*. En [53], con una modificación software en el robot Lego Mindstorms EV3, se identifica el proceso de aprendizaje del estudiante en el desarrollo de actividades de robótica educativa.

Los datos educativos se pueden analizar para obtener informaciones útiles sobre el proceso de aprendizaje de los estudiantes pero también para poder ayudar al profesor a centrar las acciones de recuperación según las capacidades individuales de los alumnos [31]. En [3], los autores analizan las interacciones de los estudiantes para predecir la motivación de los estudiantes. En [23] los autores aplican técnicas de aprendizaje no supervisado para agrupar estudiantes con patrones de interacción similares. Los grupos se muestran al profesor, que puede proporcionar algunas recomendaciones adicionales únicamente a los grupos que considere oportunos.

El profesor juega un papel fundamental en el proceso de enseñanza aprendizaje del alumnado; por esta razón, es fundamental que el profesor obtenga informaciones proporcionadas por las actividades desarrolladas por cada alumno, así como las dificultades y los progresos realizados por cada estudiante. Para realizar el análisis, los profesores necesitan herramientas que sean fáciles de utilizar y cuyos

resultados sean fácilmente interpretables [61]. En concreto, la visualización de información utiliza técnicas gráficas para ayudar a las personas a comprender y analizar los datos [36]. Las representaciones visuales permiten a los usuarios ver, explorar y comprender grandes cantidades de información a la vez. Por ejemplo, en [14], se han empleado técnicas de aprendizaje supervisado para la construcción de modelos predictivos y proponen a los profesores un tablero de instrumentos (en inglés *dashboard*) para inspeccionar dichos modelos.

En esta tesis, las técnicas de minería de datos que se se han considerado útiles para conocer el rendimiento de los estudiantes son: regresión logística, y redes bayesianas. Los modelos de regresión logística extienden la teoría de respuesta al ítem, incluyendo los componentes del conocimiento. Las redes bayesianas modelan el aprendizaje de los estudiantes capturando la dinámica del conocimiento de manera probabilística.

En este trabajo se utiliza como modelo de regresión logística el *Additive Factor Model*(AFM) para detectar el nivel de conocimiento de los alumnos y los contenidos significativos en su conocimiento y como modelo de redes Bayesiana el modelo *Conjunctive Knowledge Tracing*(CKT) para detectar dinámicamente la evolución del aprendizaje del estudiante.

Los modelos utilizados se detallarán en el capítulo 5, donde se explica cómo se lleva a cabo el análisis de los datos de los estudiantes.

Capítulo 3

Fundamentación pedagógica

En este capítulo se describirán los diferentes principios pedagógicos en lo que se basa este trabajo, y cómo estos principios se concretan en las actividades incluidas en el proyecto.

La estructura y diseño que se consideran fundamentales en el sistema objeto de este trabajo responden a una concepción contruccionista en un marco de desarrollo de inteligencias múltiples para conseguir en los alumnos su máxima capacidad de desarrollo.

En este contexto, la evaluación es parte integrante de la acción didáctica y no pura atribución de un número. Por tanto, debería ser considerada como una ayuda al alumno en su crecimiento formativo.

3.1. Introducción

Con el objetivo de superar la tradicional clase magistral y crear un clima de aula propicio para el aprendizaje y la comprensión, se ha creado IDEE (*Integrated Educational Didactic Environment*) [41, 42]; un prototipo de ambiente de aprendizaje que permite el uso de actividades diferenciadas y diversificadas, el desarrollo de estrategias y metodologías de enseñanza alternativas y la creación de diferentes métodos de verificación, evaluación y retroalimentación por el profesor.

IDEE es una herramienta educativa creada para fomentar un enfoque integrado e inclusivo que no cambie los contenidos tradicionales, pero que favorezca su adquisición por parte de los alumnos.

El alumno utiliza IDEE en clase para poder aprender contenidos curriculares de Física de una forma distinta a la clase magistral tradicional.

La idea es que el alumno pueda aprender contenidos curriculares a través de:

- la experiencia (con el laboratorio);

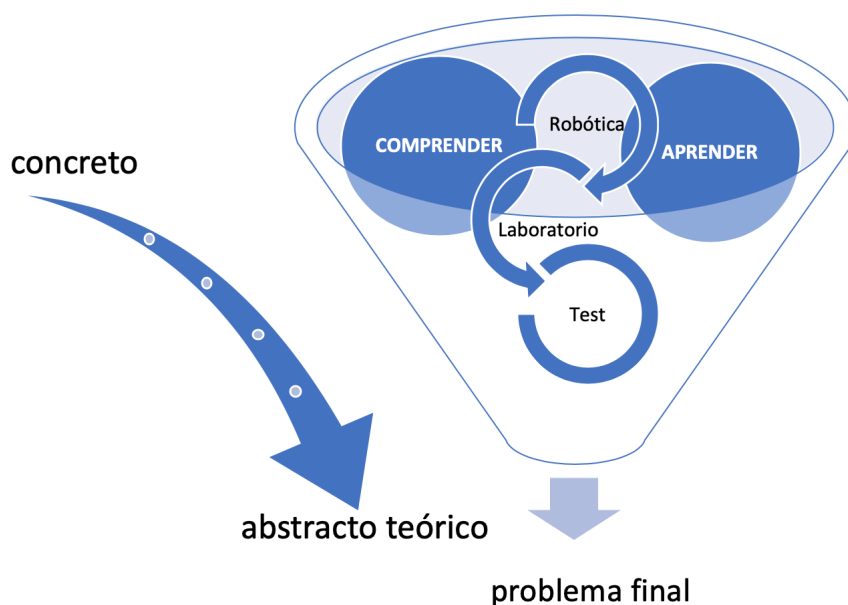


Figura 3.1: Las actividades en IDEE, crean un puente entre el conocimiento concreto y el conocimiento abstracto teórico.

- el análisis de datos e interpretación de éstos (con tests y gráficos);
- la formalización del problema y la teorización de la ley que regula el fenómeno.

El laboratorio en IDEE es un medio para comprender el fenómeno físico. Las actividades de laboratorio generan datos que deben ser adquiridos, procesados, analizados e interpretados por parte de los alumnos. Lograr esto requiere análisis y evaluación, que son destrezas de pensamiento que en IDEE se quieren desarrollar gracias a los tests; la interpretación de gráficos y las preguntas que ayudan a los alumnos a comprender la relación existente entre los datos recopilados en la actividad de laboratorio y las medidas físicas, llegando así a la formalización de la ley matemática que regula el fenómeno. Las actividades en IDEE guiadas por el profesor, crean un puente entre el conocimiento concreto y el conocimiento abstracto teórico. Se pretende que los alumnos lleguen a desarrollar habilidades que les permitan utilizar el conocimiento adquirido como una herramienta para resolver problemas.

En la Figura 3.1 se muestra el modelo de la construcción del conocimiento del alumno implementado en IDEE.

En conclusión, se intenta superar el modelo de transmisión del conocimiento basado en la idea en que los alumnos aprenden únicamente escuchando al profesor, que se convierte en el centro del modelo de aprendizaje, pasando a un modelo activo donde el alumno se convierte en el centro de su propio aprendizaje.

El modelo educativo subyacente en IDEE está basado en criterios pedagógicos

y de diseño que pueden ser aplicados a otras asignaturas, como matemáticas, química o biología.

3.2. Marco pedagógico

La introducción de la robótica en IDEE como herramienta de laboratorio sigue un enfoque claramente constructorista. En IDEE la idea es enseñar con una variedad de actividades educativas que satisfagan la diversidad en los procesos de aprendizaje de cada alumno (laboratorio, actividad de programación, tests y problemas, abstracción teórica, gráficos). Otro aspecto importante en IDEE es la evaluación individualizada, la verificación y la retroalimentación.

Este enfoque favorece la enseñanza basada en la experiencia para aprender, donde lo fundamental es “no aprender a aplicar, sino hacer para aprender” [46].

En IDEE el rol de los profesores y alumnos cambia; los profesores son ahora diseñadores de experiencias de aprendizaje (en vez de proveedores de contenido) y guías a través de esas experiencias. Por su parte, los alumnos disponen de más autonomía para crear el diseño de sus propios itinerarios de aprendizaje (y no únicamente consumiendo los contenidos incluidos en un currículum rígido).

3.3. Orientaciones generales para el diseño de actividades en IDEE

IDEE es un entorno de aprendizaje para la física centrado en los conceptos que los estudiantes deben adquirir siguiendo las normas oficiales ¹. La planificación didáctica en la que se basa IDEE está centrada en competencias, estas competencias se concretan mediante objetivos, cuyo cumplimiento indicará el nivel de aprendizaje del alumno. La competencia, según la resolución del Parlamento Europeo y del Consejo del 23 de abril de 2008, se describe como *probada capacidad para utilizar los conocimientos y habilidades personales, sociales y/o metodológicas en los contextos más diversos de la vida, el estudio y el trabajo*.

Enseñar por competencias no significa abandonar los contenidos, ya que representan el campo de experiencia en el que se ejercitan las habilidades y competencias. Sin embargo, deben ser cuidadosamente seleccionados; se deben proponer los contenidos esenciales y fundamentales y la enseñanza debe hacer todo lo posible para transformarlos en conocimiento, es decir, en el patrimonio permanente del alumno.

¹En 2006, el Parlamento Europeo recomendó un método de enseñanza basado en competencias que estructura la programación didáctica de acuerdo con diferentes unidades de aprendizaje

Una de las herramientas más completas para llevar a cabo la enseñanza por competencias es la denominada unidad didáctica de aprendizaje. Representa un segmento, más o menos grande y complejo, del plan de estudios, que tiene como objetivo hacer que los estudiantes alcancen aspectos de competencia (y, por supuesto, de sus articulaciones en habilidades y conocimientos), a través de la acción y la experiencia.

Cada unidad didáctica se propone conseguir un tema de estudio e incluye las competencias y los contenidos que se quieren desarrollar en los alumnos.

IDEE se propone trabajar los contenidos de la unidad didáctica y medir el grado de conocimiento de los alumnos sobre estos contenidos gracias a algunos parámetros que en IDEE se denominan *skills* (habilidades).

Cada experiencia se estructurará según un modelo que se concreta en estas actividades:

- Introducción del robot: para que el alumno pueda conocer las características del robot construido para el desarrollo del laboratorio.
- Actividad de laboratorio: los estudiantes tienen que programar el robot para poder recopilar datos.
- Test: los alumnos, guiados por el profesor, deberán contestar a preguntas que le harán reflexionar sobre las medidas físicas que intervienen en el fenómeno físico y como éstas se relacionan entre ellas.
- Problema final: puede considerarse como el típico problema que se encuentra en los libros de texto de los estudiantes y que el alumno tiene que resolver de manera autónoma sin la ayuda del robot.

Cada experiencia propuesta en IDEE analiza un fenómeno físico. El estudiante analiza el fenómeno, como un verdadero científico, siguiendo los pasos del método científico de Galileo Galilei. Inicialmente los estudiantes observan el fenómeno y recopilan datos gracias al desarrollo del experimento. Los laboratorios implementados en IDEE incluyen una tabla para poder introducir los datos que devuelve el robot, una representación gráfica de estos datos y un área de trabajo donde el alumno programa el robot junto con un toolbox con los bloques que los alumnos van a necesitar en la programación.

La recogida de los datos permite un análisis crítico de lo que está ocurriendo en el desarrollo del laboratorio; los errores ayudan al alumno a entender lo que está haciendo mal mientras que la representación gráfica de los datos permiten al alumno crear relaciones entre las herramientas matemáticas que necesitan utilizar y el fenómeno que se está estudiando. Gracias a los resultados recopilados durante

UNIDAD DIDACTICA	ESTUDIO CINEMATICO DEL MOVIMIENTO		
COMPETENCIA	Capacidad en describir el movimiento rectilíneo de un cuerpo utilizando las ecuaciones que relacionan espacio, velocidad y tiempo.		
CONTENIDOS	Movimiento rectilíneo uniforme	Movimiento rectilíneo uniformemente acelerado	
	Velocidad	Caída libre	
	Distancia y desplazamiento	Aceleración	
SKILL (HABILIDAD)	Interpretación grafica (skill_1)	Sistema de referencia (skill_2)	Relacionar medidas (skill_3)
	Operar matemáticamente (skill_7)	Movimiento uniformemente acelerado (skill_8)	Velocidad (skill_10)
	Tiempo (skill_11)	Conocimiento robótico (skill_12)	Movimiento uniformemente acelerado (skill_13)

Figura 3.2: Skills contenidos en IDEE para conseguir los contenidos de la programación de la unidad didáctica del estudio cinemático del movimiento.

la actividad de laboratorio, los estudiantes pueden formular hipótesis sobre las medidas que regulan el fenómeno y sus relaciones. Como último paso, los alumnos comprueban con una ley matemática las hipótesis sacando sus conclusiones, llegando al conocimiento teórico y abstracto del fenómeno que están analizando.

Actualmente IDEE contiene 4 experiencias:

- movimiento rectilíneo uniforme,
- movimiento rectilíneo uniformemente acelerado,
- caída libre,
- plano inclinado.

Las cuatro experiencias están orientadas a la comprensión del tema del estudio cinemático del movimiento rectilíneo.

Un ejemplo de *skills* en una programación didáctica de física sobre el tema del estudio del movimiento rectilíneo en un instituto de educación secundaria se puede visualizar en la Figura 3.2.

Las experiencias que didácticamente se consideran fundamentales a la hora de presentar en clase la unidad didáctica del estudio cinemático del movimiento son:

- el conocimiento de cómo se caracteriza el movimiento rectilíneo uniforme y el movimiento uniformemente acelerado;
- El análisis del movimiento en caída libre y del plano inclinado como casos de estudio y de análisis del movimiento uniformemente acelerado.

La organización de las experiencias y los objetivos que proponen se detallan a continuación.

Movimiento rectilíneo uniforme y movimiento rectilíneo uniformemente acelerado

El objetivo de las experiencias del movimiento rectilíneo uniforme y del movimiento uniformemente acelerado es comprender las características cinemáticas del movimiento y comprender las leyes matemáticas que regulan estos fenómenos.

Con las actividades de laboratorio los alumnos pueden investigar como varía el espacio en función del tiempo en estos dos casos diferentes.

Una vez que el alumno haya observado el fenómeno, el sistema IDEE le permite visualizar en una gráfica los datos que ha recopilado para observar la curva matemática que mejor representa los datos.

Gracias a las preguntas presentes en los tests, el alumno puede reflexionar sobre las relaciones entre las medidas físicas que regulan el fenómeno:

- en el movimiento rectilíneo uniforme el estudiante tiene que experimentar y descubrir la proporcionalidad directa entre espacio y tiempo.
- En el movimiento uniformemente acelerado el estudiante tiene que experimentar y descubrir la relación cuadrática entre espacio y tiempo. Además, gracias al problema final, el alumno puede aplicar la ley matemática que regula estos dos fenómenos físicos.

En estas dos experiencias se utiliza como instrumento robótico de laboratorio el robot-máquina cuyas instrucciones están incluidas en el kit educativo LEGO Mindstorms® EV3. En el experimento del movimiento rectilíneo uniforme al robot-máquina se le ha añadido un brazo que permite trazar puntos en un papel en su recorrido cada segundo. El alumno puede observar que la distancia entre los puntos trazados es siempre la misma.

En el experimento del movimiento rectilíneo uniformemente acelerado se ha incluido el sensor ultrasónico que permite al robot medir de forma autónoma las distancias, respecto a un sistema de referencia definido, en su recorrido rectilíneo con aceleración constante. La introducción del sensor es una herramienta útil para que los alumnos realicen experimentos con robots autónomos. Los datos que se recopilan se visualizan gráficamente y permiten al estudiante visualizar la relación de proporcionalidad directa con la recta y la relación cuadrática con la parábola.

Plano inclinado

El objetivo de la experiencia del plano inclinado es analizar el fenómeno de caída de un cuerpo sobre un plano inclinado. Fue Galileo quien descubrió que la caída de un objeto sigue la ley de un movimiento uniformemente acelerado.

Gracias a esta experiencia, los estudiantes pueden adquirir conocimientos sobre:

- la relación entre espacio y tiempo en el movimiento de un objeto que se desliza sobre un plano inclinado.
- Cómo se relaciona el movimiento uniformemente acelerado con el movimiento de un cuerpo que se desliza sobre un plano inclinado y con el movimiento en caída libre.
- Cómo la velocidad no es una función de la trayectoria, pero sí de la altitud desde la que cae el objeto.

Con la actividad de laboratorio se investiga sobre cómo varía la velocidad en función del tiempo. Así, los alumnos pueden comprobar que la bola no recorre espacios iguales en tiempos iguales (no sigue las leyes de un movimiento rectilíneo uniforme); la bola recorre espacios que son proporcionales al cuadrado de los tiempos. El alumno verifica gráficamente que los datos, recopilados en la tabla, representan una parábola.

Los alumnos, al realizar los tests, reflexionarán sobre el fenómeno y los datos recogidos; reflexionando sobre las leyes universales, que pueden deducirse independientemente de las obvias diferencias experimentales (relación espacio/tiempo, velocidad/tiempo y velocidad/altura).

Adicionalmente, el problema permite el análisis teórico del fenómeno. Con este ejercicio, se pone en evidencia que la velocidad final depende de la altura, pero no es una función de la masa y la trayectoria (los planos de distinta inclinación, pero de igual altura obtendrán la velocidad final sin cambios).

Caída libre

El objetivo de la experiencia de la caída libre es que los alumnos comprendan que la masa no influye en el tiempo de caída de un objeto. Con la experiencia de laboratorio, el alumno mide tiempos de caídas de objetos con masas distintas y observa que, considerando el contexto son aproximadamente iguales. Con el problema final, el alumno se da cuenta del hecho de que en la fórmula matemática, la masa no influye en el resultado.

Se muestra en la Figura 3.3 la interfaz de las actividades de laboratorio incluidas en IDEE y en la Figura 3.4 las imágenes de los robots utilizados.

LEY HORARIA DE UN MOVIMIENTO UNIFORME



- Respuesta incorrecta
- Respuesta correcta
- Sin respuesta
- Error en las medidas

Considera una potencia de 5 y utiliza el robot para determinar el espacio recorrido con el que has hecho el experimento. Repite el experimento considerando una potencia de 10 y de 15. Observa y cc



Actualiza gráfico

Tiempo	Espacio(cm)
Tiempo 1s	
Tiempo 2s	
Tiempo 3s	
Tiempo 4s	

Velocidad (cm/s)

Submit the answer

▶ Resultado:

```

Ev3
Math
start motor B+C speed 5
wait 2 seconds
set step to 0
repeat until step == 4
do
  change step by 1
  drive A time 0.05
  reverse speed 100
  drive A time 0.05
  forward speed 100
stop motors B+C brake
    
```

(a) Interfaz del laboratorio del movimiento rectilíneo uniforme

LEY HORARIA DE UN MOVIMIENTO UNIFORMEMENTE ACCELERADO



- Respuesta incorrecta
- Respuesta correcta
- Sin respuesta
- Error en las medidas

El robot se moverá con un movimiento uniformemente acelerado con una potencia inicial 5. Escribir en la tabla el espacio recorrido por el robot (Calcula la media sobre 5 misuraciones. Observa gráficamente la relación del espacio recorrido en función del tiempo.



Actualiza gráfico

Tiempo	Espacio
Tiempo 0s	
Tiempo 1s	
Tiempo 2s	
Tiempo 3s	
Tiempo 4s	
Tiempo 5s	
Tiempo 6s	
Tiempo 7s	
Tiempo 8s	
Tiempo 9s	
Tiempo 10s	

Invia la respuesta

▶ Result:

```

Ev3
Math
Connection EV3 "acelerado"
power
    
```

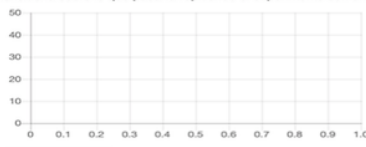
(b) Interfaz del laboratorio del movimiento uniforme acelerado.

GALILEO Y SUS IDEAS PRUEBA22



- Respuesta incorrecta
- Respuesta correcta
- Sin respuesta
- Error en las medidas

Reflexionamos sobre la relación entre el espacio recorrido por la pelota y el tiempo d Programa el robot para que tenga una inclinación de 15 grados. Calcula el tiempo de Reflexiona sobre lo que pasaría repitiendo el experimento con una inclinación de 10



Actualiza gráfico

Incógnita	Tiempo de caída
Espacio1	
Espacio2	
Espacio3	
Espacio4	

Submit the answer


▶ Resultado:

```

Ev3
Math
answer
message name "TimePlane"
Connection EV3 "angulo"
angle
Connection EV3 "PlanoInclinado"
message "Tiempo_Caida"
    
```

(c) Interfaz del laboratorio del plano inclinado.

¿UNA PELOTA DE ALGODÓN Y UNA PELOTA DE ACERO LLEGARÁN AL MISMO TIEMPO?



- Respuesta incorrecta
- Respuesta correcta
- Sin respuesta
- Error en las medidas

Utiliza el robot para determinar el tiempo de caída de pelotas de masa distinta; con cada pelota ca

Pelota	Tiempo(media)
Pelota1	
Pelota2	

Invia la respuesta

▶ Result:

```

Ev3
answer
    
```

(d) Interfaz del laboratorio de la caída.

Figura 3.3: Interfaces de las actividades de laboratorio.



Figura 3.4: Imágenes de los robots utilizados en las actividades de laboratorio.

3.4. Soporte para el profesor y el alumno

La evaluación es una parte integral de la programación didáctica, acompaña y regula la acción didáctica con respecto a las habilidades y resultados detectados en el grupo de clase, apoya los procesos de aprendizaje y maduración del alumno.

Evaluar significa:

- acompañar, guiar y apoyar al estudiante en su proceso de aprendizaje a lo largo de toda la ruta educativa para hacerle responsable de los objetivos esperados.
- Promover la autoevaluación del alumno en términos de conciencia de los resultados alcanzados y de sus habilidades.
- Llevar a cabo una función reguladora de los procesos de enseñanza para ayudar a mejorar la calidad de la enseñanza.

Gracias a los datos almacenados en la base de datos, en IDEE es posible:

- determinar el nivel de conocimiento del alumno sobre los contenidos de las actividades propuestas. De esta manera, es posible ayudar a los estudiantes a aprender según su ritmo de aprendizaje, proporcionándoles problemas de acuerdo con su nivel de aprendizaje.
- Ayudar al alumno en su aprendizaje informándole de inmediato de sus errores y sus aciertos.

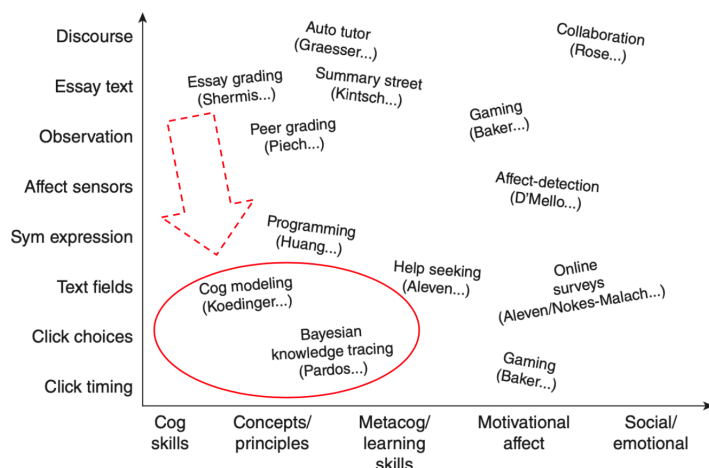


Figura 3.5: En la gráfica [31] se muestran las distintas áreas de investigación en minería de datos educativa.

- Proporcionar un análisis estadístico de los datos utilizando el *Additive Factor Model* (AFM)[12] y el *Conjunctive Knowledge Tracing* (CKT) [32]. Finalmente, es posible facilitar estos resultados de forma gráfica al profesor para que él pueda tener información sobre la ruta de aprendizaje de sus alumnos.

Las actividades en IDEE están relacionadas con *skills*, por este motivo se ha decidido utilizar modelos que trabajen sobre *skills* múltiples, utilizando las informaciones de los modelos como instrumentos útiles para la determinación del conocimiento del alumno.

En la Figura 3.5 se muestran las distintas áreas de investigación de minería de datos educativa para poder sacar informaciones útiles sobre el aprendizaje. En el eje vertical se indican los diferentes tipos de datos que pueden ser obtenidos con diferentes recursos tecnológicos y en el eje horizontal se encuentran las estructuras exploradas con estos tipos de datos. En rojo se marca el área que nos ha sido útil para conseguir nuestros propósitos en IDEE.

Gracias a las informaciones proporcionadas por IDEE:

- el alumno puede reflexionar sobre sus respuestas y encontrar de forma autónoma los recursos para poder solucionar el problema.
- El profesor puede saber si los alumnos han trabajado en clase, dónde han tenido muchos fallos o sobre qué *skills* necesitan mejorar. Con esta información y con el conocimiento propio del profesor, de su grupo clase y de sus alumnos, se considera correcto desde un punto de vista metodológico, dejar al profesor la tarea de evaluación final que engloba los conocimientos adquiridos por parte del alumno en su formación cognitiva.

Capítulo 4

Desarrollo de IDEE

A lo largo de este capítulo se describirán los elementos empleados para la realización del entorno de aprendizaje IDEE.

IDEE es un entorno de aprendizaje de física que utiliza la robótica como instrumento para desarrollar laboratorios. Las actividades incluidas en IDEE están diseñadas para alumnos de secundaria. Los alumnos en IDEE tienen que construir pequeños robots que les permitan realizar las tareas de laboratorio. Posteriormente, podrán profundizar con los datos obtenidos con la experiencia práctica, mediante ejercicios, preguntas y problemas.

4.1. Visión general

Los estudiantes utilizan IDEE a través de una interfaz que les permite programar el robot, solucionar problemas y ver las soluciones de las actividades propuestas. Una vez finalizada la actividad, las interacciones y las soluciones aportadas por los estudiantes se guardan en la base de datos. Estos datos se analizan para permitir que el profesor pueda seguir el aprendizaje del alumno en clase y su evolución.

La interfaz del estudiante está compuesta por los siguientes elementos: declaración del problema, tabla donde el estudiante introduce las respuestas, visualización gráfica de las respuestas del estudiante, botón de activación del robot, área donde aparecen los datos obtenidos con el robot, toolbox con los bloques que serán necesarios para la programación del robot o para la resolución de la actividad propuesta y el área de programación para el estudiante. En la Figura 4.1 se visualiza la interfaz del estudiante en la actividad de laboratorio del plano inclinado.

Mientras el alumno trabaja en IDEE, el sistema apoya al alumno con las herramientas de autoevaluación y el profesor guía al grupo clase en la comprensión dejando libre a los alumnos para construir sus conocimientos.

The screenshot shows the IDEE student interface for a laboratory activity. At the top left, there is a small image of Galileo Galilei and a legend with four colored boxes: red for 'Respuesta incorrecta', green for 'Respuesta correcta', yellow for 'Sin respuesta', and blue for 'Error en las medidas'. Below this is the 'Texto del problema' (Problem text) which asks the student to calculate the time of fall of a ball on an inclined plane of 15 degrees and to reflect on what would happen with a 10-degree incline. A 'Gráfico' (Graph) is provided with a grid for plotting. Below the graph is a 'Respuesta alumno' (Student answer) field with four input boxes labeled 'Espacio1' through 'Espacio4'. To the right of the answer field is a 'Botón activación Robot' (Robot activation button) with a red toggle switch. Below the answer field is the 'Resultado Robot' (Robot result) section. At the bottom is the 'Área de programación' (Programming area) with a block-based programming interface showing blocks for 'message name', 'TimePlane', 'angulo', 'PianoInclinado', and 'Tiempo_Caida'. A 'Bloques' label points to this area. A 'Submit the answer' button is also visible.

Figura 4.1: Interfaz del estudiante en la actividad de laboratorio del plano inclinado.

El sistema IDEE ayuda al alumno corrigiendo de forma automática sus errores y proporcionándole problemas según su nivel de aprendizaje [43].

Se considera fundamental la autocorrección por parte de los alumnos; gracias a los errores, el alumno puede reflexionar sobre sus respuestas. Para cada ejercicio, el alumno puede realizar tres intentos; en cada intento, las respuestas correctas se marcan en verde, las incorrectas en rojo, las preguntas no contestadas en amarillo y en azul las respuestas correctas numéricamente pero con medidas incorrectas. En las actividades de laboratorio, se considera un margen de error del 3% a la hora de dar una respuesta del alumno por válida.

En la Figura 4.2 se muestra un ejemplo de corrección en IDEE. El profesor, terminada la clase, necesita conocer qué alumnos han podido tener dificultades en el aprendizaje del tema propuesto y necesita individualizar las causas de estas dificultades. Para ello, debe tener en cuenta que cada estudiante aprende con ritmos distintos y que siempre es difícil identificar los puntos de dificultad de cada alumno en una clase no estructurada. IDEE tiene como objetivo fundamental ayudar al profesor en clase. Para ello, se ha introducido en IDEE un panel de control o dashboard en el que el profesor puede consultar información acerca del proceso de aprendizaje de los estudiantes en los contenidos propuestos y para centrar la atención en aquellos alumnos que el sistema indica que pueden estar

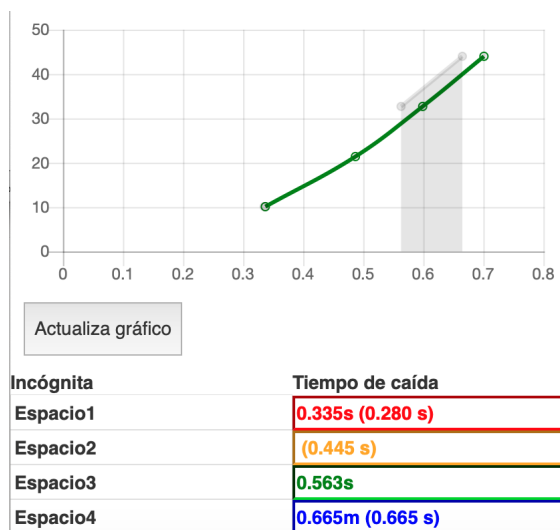


Figura 4.2: Ejemplo de corrección en IDEE.

teniendo problemas.

Para ello, la arquitectura de IDEE se ajusta a la estructura de un sistema tutorial inteligente [47]. Esta estructura se puede visualizar en la Figura 4.3 y está compuesta por:

- interfaz del estudiante: contiene las actividades que se presentan al estudiante.
- modelo del alumno: contiene el conocimiento que el alumno tiene sobre el dominio del tema a tratar. Este modelo, recoge el comportamiento evolutivo del alumno durante el trabajo, modela el conocimiento del alumno y a partir de esta información adapta los contenidos que ofrece el sistema.
- modelo pedagógico: contiene una representación del conocimiento experto en los ámbitos relativos a procesos de evaluación, enseñanza-aprendizaje y metodología de enseñanza.
- modelo didáctico: contiene información para decidir qué tareas se le presentan al estudiante de acuerdo con los objetivos de aprendizaje del módulo pedagógico.
- modelo robótico: permite la comunicación del sistema con el robot LEGO® EV3.
- dashboard del docente: proporciona al docente información acerca del progreso de aprendizaje de sus estudiantes.
- modelo del docente: contiene los instrumentos que permiten detectar el progreso del conocimiento del alumno.

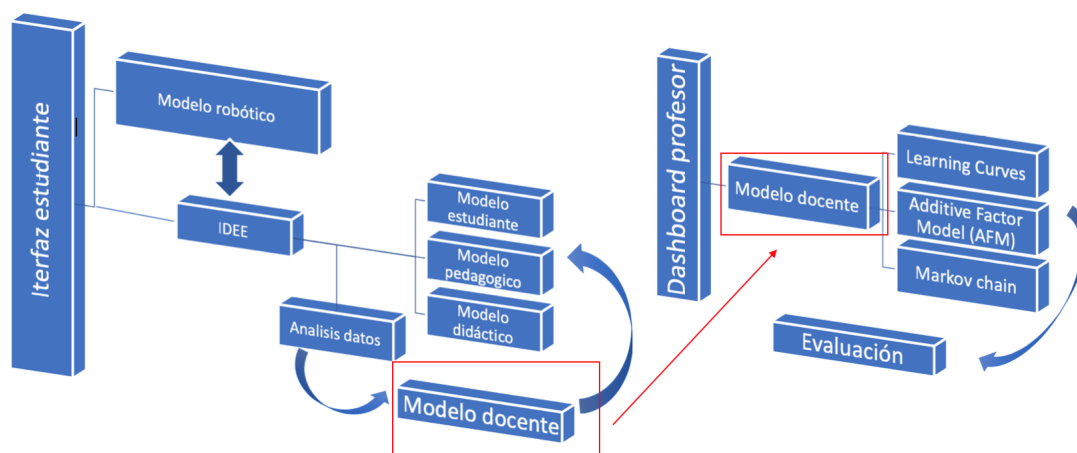


Figura 4.3: Modelo del sistema IDEE.

El entorno de aprendizaje profesor/alumno propuesto IDEE, proporciona ayudas en la resolución de los ejercicios a los estudiantes que detecta que están teniendo problemas.

Como se verá más adelante en la memoria, el análisis del aprendizaje de los alumnos se realiza en IDEE utilizando algoritmos estadísticos adaptados al contexto educativo.

En el capítulo 5 de la tesis se detalla la descripción de los modelos estadísticos utilizados y los contenidos del panel de control que ayudan al profesor a seguir el aprendizaje de los alumnos.

4.2. Descripción de la Arquitectura

La arquitectura de IDEE se muestra en la Figura 4.4 y se basa en los siguientes elementos:

- Django¹: es un framework que permite la creación y gestión de aplicaciones web. La gestión de usuarios y la interacción con la base de datos está integrada en el propio framework. Puesto que Django está implementado en Python, ese es el lenguaje que se debe emplear en la lógica del servidor. Adicionalmente, en la implementación de IDEE ha sido necesario emplear otros lenguajes como Javascript, HTML, CSS y también PostgreSQL para la gestión de la base de datos.
- Blockly²: es un entorno de programación visual de código abierto desarrollado por Google. Se utiliza para permitir que los estudiantes implementen

¹<https://www.djangoproject.com>

²<https://developers.google.com/blockly/>

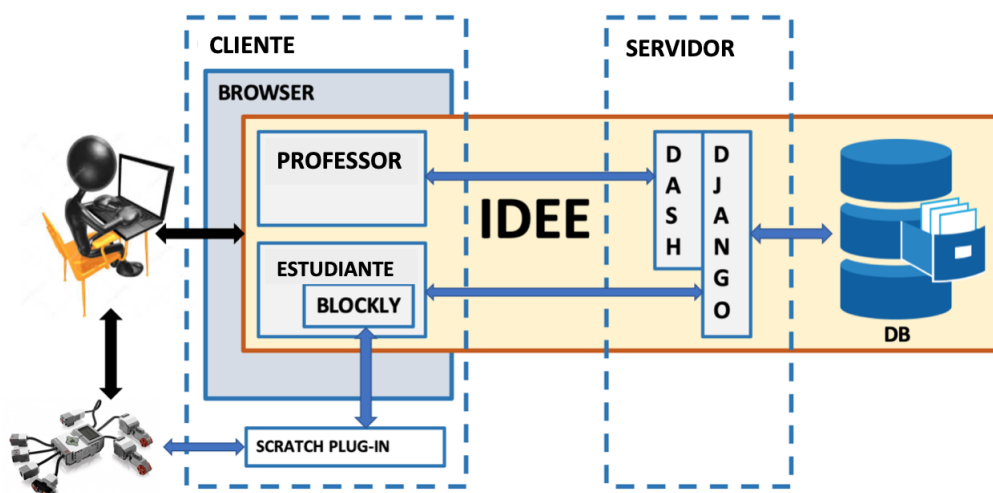


Figura 4.4: Arquitectura de IDEE

programas durante las actividades de aprendizaje. Blockly es uno de los entornos más utilizados para este tipo de aplicaciones.

- Scratch Plug-in³: permite la comunicación entre el programa Blockly y el robot LEGO® EV3 a través de una conexión bluetooth.
- Dash⁴: se utiliza para desarrollar el escritorio o *dashboard* del profesor.
- Base de datos PostgreSQL: se utiliza para almacenar y recuperar los datos registrados durante las actividades.

4.3. Implementación

Para la implementación de IDEE se ha considerado oportuno crear una aplicación web interactiva, que permita al alumno trabajar en páginas dinámicas y al profesor la visualización de los resultados de aprendizaje. Para el desarrollo de la arquitectura de IDEE se ha decidido utilizar Django. Como se ha comentado anteriormente, Django es un framework web de alto nivel que se caracteriza por su sencillez, rapidez y su reutilización de código. Las ventajas del uso de Django son:

- Orden: Django, de acuerdo con la filosofía de Python, ayuda (a veces obliga) a ser ordenado. Todo debe estar en su lugar y por esta razón moverse entre el código del proyecto se convierte en una simple acción.

³https://scratch.mit.edu/info/ext_download/

⁴<http://www.dash-project.org>

- Gratis y de código abierto: esto significa basar su software en plataformas no vinculadas por contratos de terceros.
- Optimización de los tiempos de desarrollo: todos o casi todos los aspectos del desarrollo web están automatizados: sesiones de inicio de sesión, mapas del sitio, patrones de URL, validación de formularios; todo se gestiona de una manera muy sencilla.
- ORM (del inglés *Object-Relational Mapping*) para bases de datos: en Django se utiliza una metodología extremadamente simple para la gestión de datos. Cada tabla se convierte en un objeto (o clase) cuyas propiedades se corresponden con las diferentes columnas de la tabla. Así, cada registro en la tabla será una instancia de dicho objeto.
- Seguridad: en Django todo se gestiona de forma sencilla y segura, las sesiones de inicio de sesión y el ORM son un ejemplo.
- Completo: Django incluye todo lo necesario para crear un sitio web. Entre otras funcionalidades, proporciona: modelos html, ORM para bases de datos, administración de sesiones y un panel de control.
- Lenguaje de programación: Django usa Python, un lenguaje que se está convirtiendo de facto en un estándar en el desarrollo de aplicaciones web. Esto permite disponer de un gran número de módulos y bibliotecas que ofrecen una gran diversidad de funcionalidades.

Más concretamente, Django está basado en el paradigma MTV (del inglés *Model-Template-View*). El modelo hace referencia al acceso a la capa de datos y la vista se refiere a la parte del sistema que selecciona qué mostrar y cómo mostrarlo. Finalmente la plantilla implica la parte del sistema que decide qué vista usar, dependiendo de la entrada del usuario, accediendo al modelo si es necesario.

El modelo define en forma de clases de Python la estructura de los datos almacenados, cada tipo de dato que debe ser almacenado se encuentra en una variable con ciertos parámetros. Una vez que hayamos decidido cuáles serán nuestros modelos y sus campos, debemos pensar en la relación que existe entre ellos. Django le permite definir relaciones de uno a uno (*OneToOneField*), de uno a muchos (*ForeignKey*) y de muchos a muchos (*ManyToManyField*). Cada modelo está compuesto por una serie de atributos, que representan un campo de la tabla correspondiente. Django ofrece automáticamente una API Python que proporciona acceso a las bases de datos. Gracias a esta API, Django ejecuta código SQL y

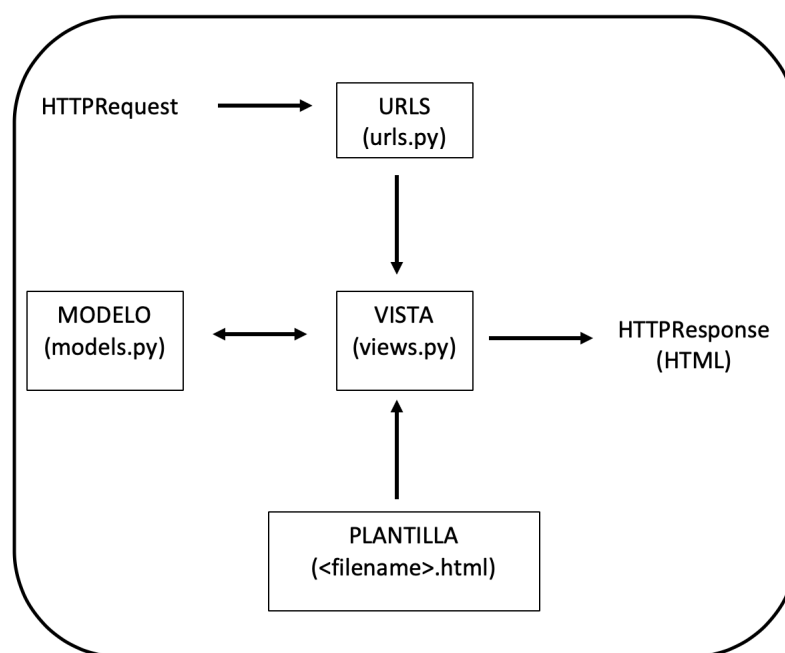


Figura 4.5: Descripción de la colaboración entre capas MTV

devuelve estructuras de datos Python que representan filas de las tablas en la base de datos.

La vista se presenta en forma de funciones en Python y su propósito es determinar qué datos se mostrarán. Las vistas en Django reciben una petición Web de un usuario, que será un objeto `HttpRequest`, y devuelve la respuesta mediante un objeto `HttpResponse`. De esta forma, se presentan al usuario páginas con los datos solicitados y no solo páginas estáticas.

La presentación de los datos es tarea de la plantilla (*template*), que básicamente es una página HTML.

En la Figura 4.5 se muestra la arquitectura de Django.

4.3.1. Paradigma MTV en IDEE

El proyecto Django para la creación de IDEE se llama *web_learning*; la estructura MTV creada se muestra en la parte izquierda de la Figura 4.6.

En IDEE se han creado distintos modelos para garantizar la introducción de las experiencias y el correspondiente almacenamiento de los datos necesarios para el seguimiento del aprendizaje de los estudiantes.

Se ha creado el modelo *experiments*, que incluye las experiencias creadas en IDEE donde cada experiencia describe un fenómeno físico.

Cada experimento está relacionado con una relación uno a muchos con el modelo *elements*; los elementos pueden ser problemas o tests. Los problemas pueden tener distintas versiones y los tests pueden tener distintas preguntas (en

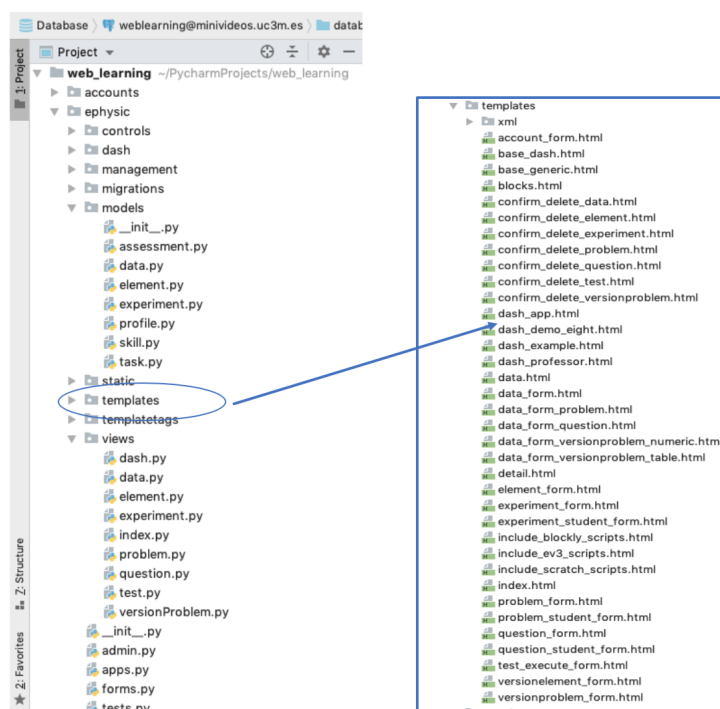


Figura 4.6: Descripción de la organización MTV del proyecto *web_learning* para la creación de IDEE

el modelo, las versiones y las preguntas se encuentran en el modelo *assessment*).

Cada *assessment* está relacionado con una relación de muchos a muchos con los skills que el alumno tiene que conseguir en las actividades (los skills se definen en el modelo *skills*). En la creación de cada actividad se define el parámetro *score* para cada skill; que representa el nivel de conocimiento al que contribuye dicho skill. Este parámetro es la clave para permitir al sistema obtener el conocimiento adquirido por los estudiantes durante las actividades y personalizar los problemas.

La estructura de las experiencias en IDEE y la relación con los skills que el alumno debe adquirir durante la actividad se muestran en la Figura 4.7.

Para garantizar la evaluación de los estudiantes, se han implementado otros modelos en IDEE:

- modelo *task*: para tener conocimiento de las interacciones del alumno en el sistema. En la clase *task* hay un campo que almacena el momento en que cada alumno empieza cada actividad y el momento en que termina.
- modelo *data*: el sistema almacena las actividades que va realizando el alumno guardando la respuesta del alumno y el momento en que se da esa respuesta. Cada respuesta está relacionada con la actividad desarrollada por el alumno. También se almacena la respuesta correcta para poder evaluar los errores de los estudiantes.

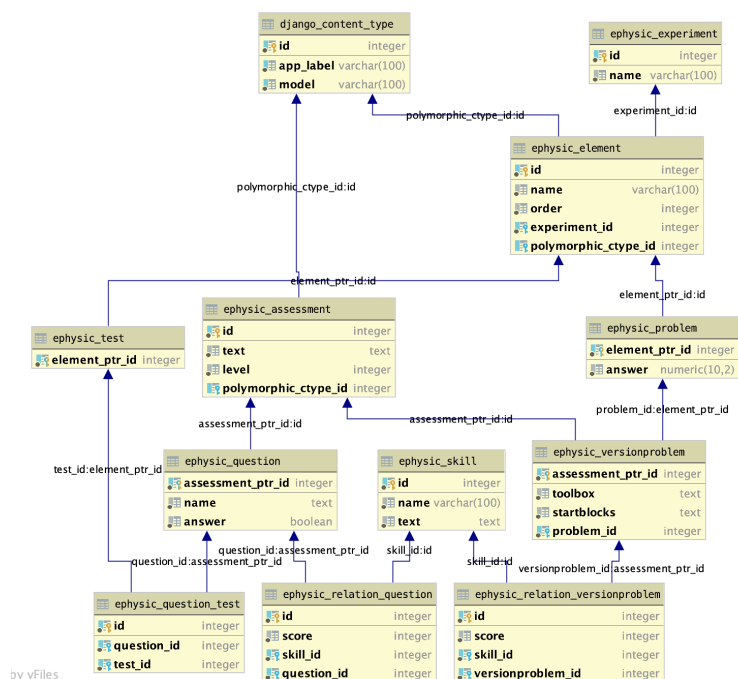


Figura 4.7: La imagen muestra una parte de la base de datos IDEE que incluye la estructura de las experiencias y la relación con skills y *score*.

- modelo *profile*: IDEE puede almacenar el progreso del aprendizaje del estudiante gracias al modelo *profile* que relaciona, la actividad y la habilidad (skill), con el conocimiento que los estudiantes adquieren sobre cada actividad y cada skill en la que trabaja el estudiante.

En la Figura 4.8 se muestran las tablas de los modelos mencionados anteriormente y cómo están relacionados en la base de datos.

En la Figura 4.9 se visualiza el diagrama de asociación UML que muestra la estructura completa de la base de datos en IDEE.

Adicionalmente, para cada actividad, se ha definido una vista que permite crear las actividades; una que permite mostrarlas, cancelarlas o modificarlas y una vista *dash* que permite las visualizaciones al profesor del análisis de los datos de aprendizaje de los estudiantes.

Gracias a las vistas y plantillas creadas en IDEE:

- el alumno puede moverse y trabajar en las distintas páginas web de IDEE;
- el profesor puede visualizar cómo el alumno trabaja y los objetivos de aprendizaje alcanzados;
- el profesor puede incluir contenidos didácticos en el sistema, para poder desarrollar su clase.

En la Figura 4.10 se muestran las vistas creadas en IDEE.

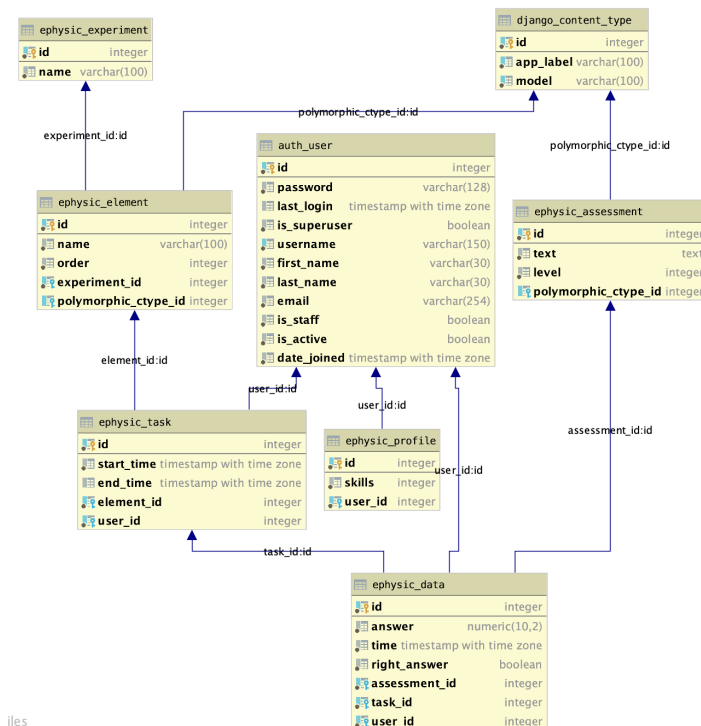


Figura 4.8: La imagen muestra una parte de la base de datos de IDEE que incluye los modelos que permiten seguir el aprendizaje del alumno: clase *task*, clase *data* y clase *profile*.

Para la implementación de las plantillas, se ha generado un nuevo directorio en el proyecto: *Templates* (ver la Figura 4.6). En este directorio será donde se incluyan las distintas plantillas generadas para la aplicación. Además, se ha utilizado la herencia de plantillas y se ha creado una plantilla base (denominada *base_generic.html*), que sirve como raíz del resto de plantillas. Esta plantilla definirá las partes comunes en IDEE. En concreto, define la barra de navegación superior y la parte inferior de la página. Adicionalmente, se han creado: plantillas que estructuran los elementos de las páginas y plantillas que permiten al profesor editar los contenidos en IDEE.

Se muestra en la Figura 4.11 un diagrama de las plantillas que se utilizan en IDEE.


```

8 def experiment_show(request, experiment_id):
9     experiment = Experiment.objects.get(pk=experiment_id)
10    context = {
11        'response': experiment
12    }
13    return render(request, 'detail_experiment.html', context)
14
15
16 def experiment_student_show(request, experiment_id):
17    experiment = get_object_or_404(Experiment, pk=experiment_id)
18    block_element = Element.objects.filter(experiment=experiment_id)
19    context = {
20        'experiment': experiment,
21        'block_element': block_element
22    }
23    return render(request, 'experiment_student_form.html', context)
24
25
26 def experiment_create(request, template_name='experiment_form.html'):
27    form = ExperimentForm(request.POST or None)
28    if form.is_valid():
29        form.save()
30        return redirect('index')
31    return render(request, template_name, {'form': form})
32
33
34 def experiment_update(request, experiment_id):
35    experiment = get_object_or_404(Experiment, pk=experiment_id)
36    block_element = Element.objects.filter(experiment=experiment_id)
37    form = ExperimentForm(request.POST or None, instance=experiment)
38    context = {
39        'form': form,
40        'experiment_id': experiment_id,
41        'block_element': block_element
42    }
43    if form.is_valid():
44        form.save()
45        return redirect('index')
46    return render(request, 'experiment_form.html', context)
47
48
49 def experiment_delete(request, experiment_id, template_name='confirm_delete_experiment.html'):
50    experiment = get_object_or_404(Experiment, pk=experiment_id)
51    if request.method == 'POST':
52        experiment.delete()

```

Figura 4.10: Vistas del proyecto *web_learning* para la creación de IDEE.

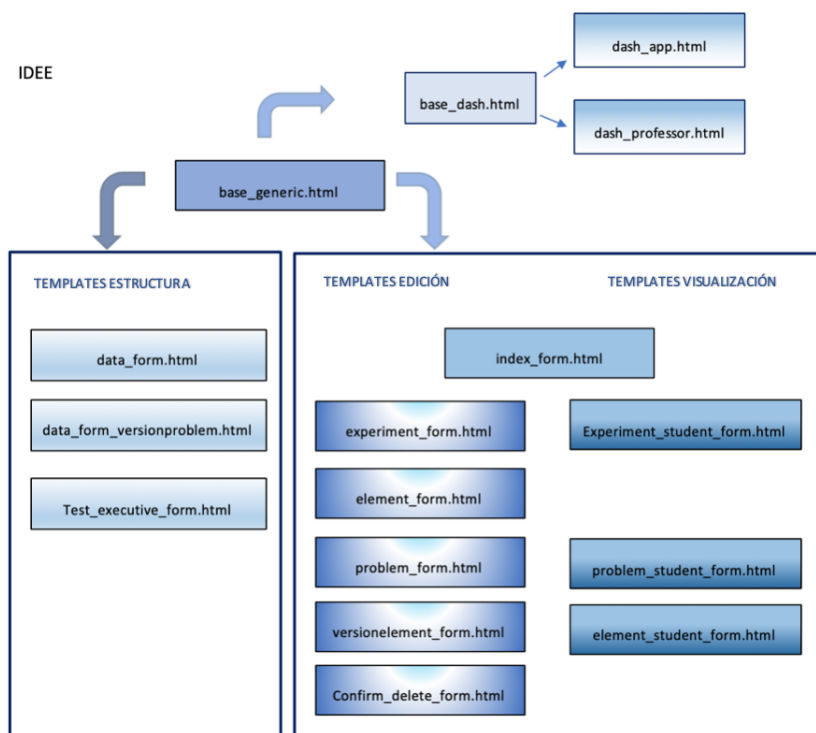


Figura 4.11: Plantillas del proyecto *web_learning* para la creación de IDEE.

4.3.2. Módulo del profesor

En el desarrollo de un sistema que se quiera utilizar en clase es importante que los contenidos sean adaptables a las necesidades didácticas del profesor. Por tanto, se ha procurado que el profesor pueda elegir entre utilizar el material didáctico incluido en IDEE o introducir sus nuevas experiencias adaptando los contenidos a su grupo clase. Para lograr esta funcionalidad, se hace uso de los formularios (*forms*) que proporciona Django.

El profesor para incluir una actividad puede escribir el texto del problema, los bloques de programación en el toolbox y relacionar la actividad con los skills que incluye y el parámetro *score* que le corresponde. En la Figura 4.12 se visualiza la página web *versionproblem_form* a través de la cual el profesor define un problema.

Welcome to Physics World!
[Home](#) [Logout](#)

Name:
 Movimiento Plano Inclinado

Answer:
 {("type": "table", "col_names": ["Incógnita", "Tiempo de caída"], "rows": [{"Espacio1": "0.280 s"}, {"Espacio2": "0.445 s"}, {"Espacio3": "0.563 s"}, {"Espacio4": "0.665 s"}])}

Text:
 GALILEO Y SUS IDEAS PRUEBA2

Reflexionamos sobre la relación entre el espacio recorrido por la pelota y el tiempo de caída reproduciendo el mismo experimento que hizo Galileo Galilei.

Toolbox:
 b{<category id="%{BKY_CATEV3}" name="Ev3" colour="%{BKY_EV3_HUE}"><block type="ev3_read_message"><value name="name"><block type="text"><field name="TEXT">TimePlane</field></block></value></block><block type="ev3_send_message_angle"><value name="name"><block type="text"><field

Startblocks:
 b{<code></code>

Level: 1

- skill2 0
- skill3 8
- skill4 0
- skill5 0
- skill6 0
- skill8 0
- skill9 5
- skill10 0
- skill11 0
- skill12 0
- skill13 10
- skill14 0
- skill15 0
- skill16 0
- skill17 0
- skill18 0
- skill19 0

Figura 4.12: Página web que permite al profesor modificar o introducir un problema en IDEE.

4.4. Programación del robot para cada experiencia

En la creación del entorno IDEE se ha presentado la necesidad de elegir el lenguaje de programación más apto para permitir a los alumnos programar el robot y desarrollar los laboratorios.

Los lenguajes visuales de programación actualmente son ampliamente utilizados en el ámbito de la enseñanza para los niños y adolescentes, siendo un tipo de programación intuitiva y accesible [51]. El uso de bloques de colores permite superar a los estudiantes uno de los dos desafíos a los que deben enfrentar todos los principiantes de programación: el de transformar sus ideas en declaraciones lógicas manteniendo la sintaxis correcta. Para los adolescentes que se acercan a la programación por primera vez, esta es una ventaja decisiva.

Los principales lenguajes de programación visual disponibles hoy en día son: Scratch⁵, Snap!⁶ y Blockly⁷.

Snap! fue desarrollado en la universidad de Berkeley con la intención de poder enseñar conocimientos de programación a cualquier persona independientemente de su nivel y su formación.

Blockly tiene como objetivo proporcionar una caja de herramientas que pueda incluirse en cualquier sistema web cuyo objetivo sea proporcionar un entorno de programación por bloques.

El código desarrollado en Blockly puede exportarse a diferentes lenguajes: Java, Python, PHP y Dart. De manera similar a Scratch, Blockly permite crear aplicaciones moviendo bloques de código análogo a piezas de Lego, sin tener que escribir complicadas líneas de código.

Blockly proporciona la herramienta Blockly Developer Tools alojado en la página web ⁸ que puede ayudar en la creación de nuevos bloques. Este editor proporciona las herramientas para crear la forma gráfica de los bloques, exportarlos e incluirlos en una aplicación externa.

En la Figura 4.13 se muestra cómo se presenta la página de Blockly.

Tanto Scratch como Snap! ofrecen un sitio web en el que poder desarrollar proyectos utilizando sus herramientas. Por el contrario, Blockly puede incluirse en el propio sistema que se esté creando. Esta característica nos ha hecho elegir Blockly para nuestro proyecto, ya que en nuestro entorno IDEE necesitábamos implementar un módulo que permitiera a los estudiantes utilizar programación por bloques.

4.4.1. Área de programación en IDEE

Estructuralmente hablando, el área de programación de IDEE se compone de dos partes:

- Toolbox: define el conjunto de bloques organizados por categorías que se

⁵<https://scratch.mit.edu>

⁶<https://snap.berkeley.edu>

⁷<https://developers.google.com/blockly>

⁸<https://blockly-demo.appspot.com/static/demos/blockfactory/index.html>

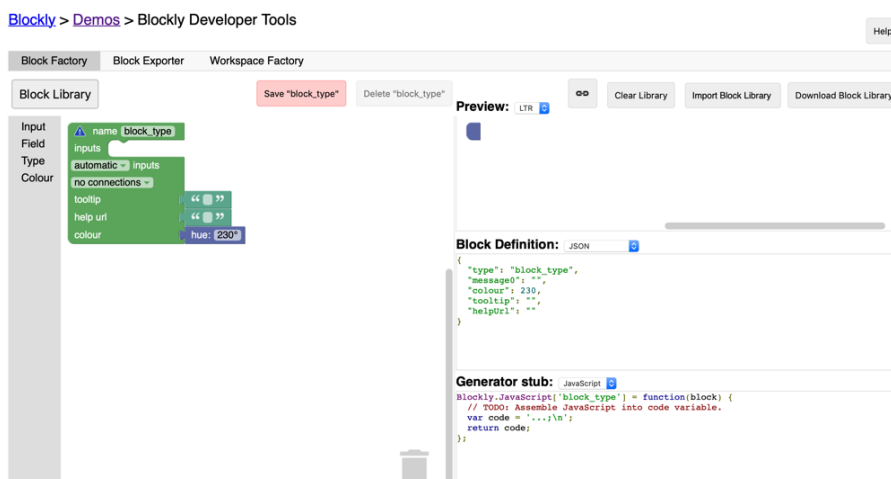


Figura 4.13: Página de trabajo para la creación de nuevos bloques.

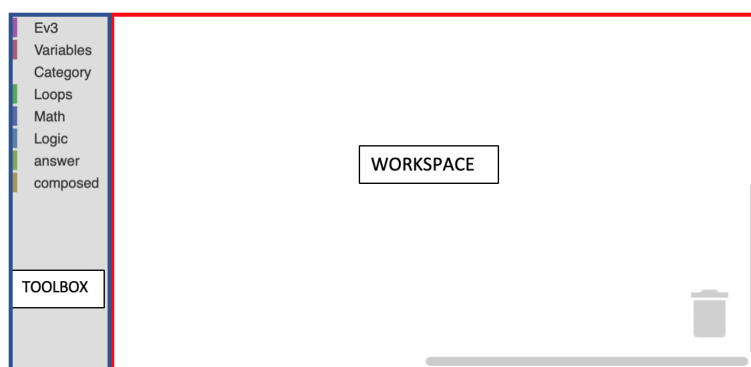


Figura 4.14: Área de programación de una actividad genérica en IDEE

podrán conectar entre sí; el toolbox es el área donde se encuentran los bloques que el estudiante puede utilizar para programar el robot. Esta área solo contendrá aquellos bloques que le sean de utilidad al estudiante en función del robot diseñado y de la actividad que está desarrollando.

- Workspace: área donde se pueden insertar los bloques para generar el código.

En la Figura 4.14 se visualiza el área de programación en IDEE que permite a los estudiante resolver problemas y programar el robot.

Los bloques implementados en IDEE se han diferenciado en:

- bloques de input: bloques que trabajan con la introducción de datos, como los bloques que gestionan los sensores;
- bloques de output: bloques que restituyen datos, como los bloques de acción;
- bloques matemáticos para el desarrollo de las operaciones matemáticas;
- bloques de control de flujo: para poder trabajar con ciclos y repeticiones;

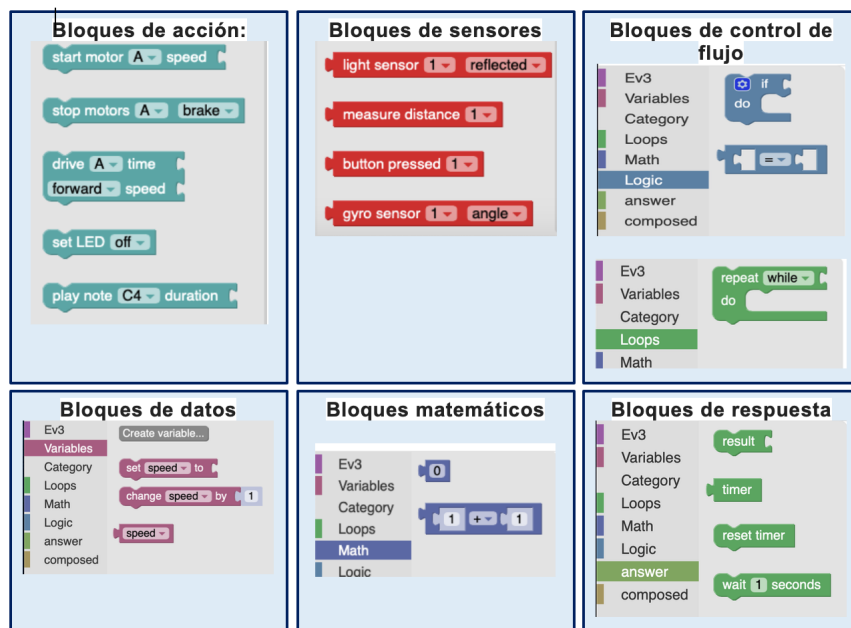


Figura 4.15: Bloques que se pueden incluir en el toolbox de las actividades de IDEE

- bloques de datos: para trabajar con variables;
- bloques de respuesta: necesario para devolver el resultado de la solución propuesta por el estudiante para que pueda ser almacenada en IDEE. Además, permite devolver el tiempo que ha tardado en ejecutarse el experimento.

En la Figura 4.15 se visualizan los bloques creados en IDEE.

4.4.2. Creación de bloques adicionales en IDEE

El conjunto de bloques disponibles en Blockly en ocasiones, no son suficientes para la resolución de los ejercicios propuestos en IDEE. Es necesario, por tanto, combinar diferentes bloques para realizar una única operación. No obstante, el objetivo de IDEE no es que el estudiante aprenda a programar si no que utilice los bloques para manejar el robot del experimento. Por tanto, se ha visto la necesidad de crear nuevos bloques para facilitar el uso de la toolbox al estudiante y permitir que se centre en la resolución del ejercicio y no en la programación. A continuación, se muestra cómo se ha realizado la implementación de nuevos bloques en IDEE.

El primer paso para poder construir un bloque que se quiere incluir en el toolbox es definir la forma, el campo y los puntos de conexión. Una vez creada la gráfica del bloque, el segundo paso es la creación del código generador. Para ello, se ha incluido en IDEE una librería JavaScript ⁹ que permite la comunicación

⁹Desarrollada por Ken Aspeslagh en 2015: <https://github.com/LLK/scratchx/wiki>

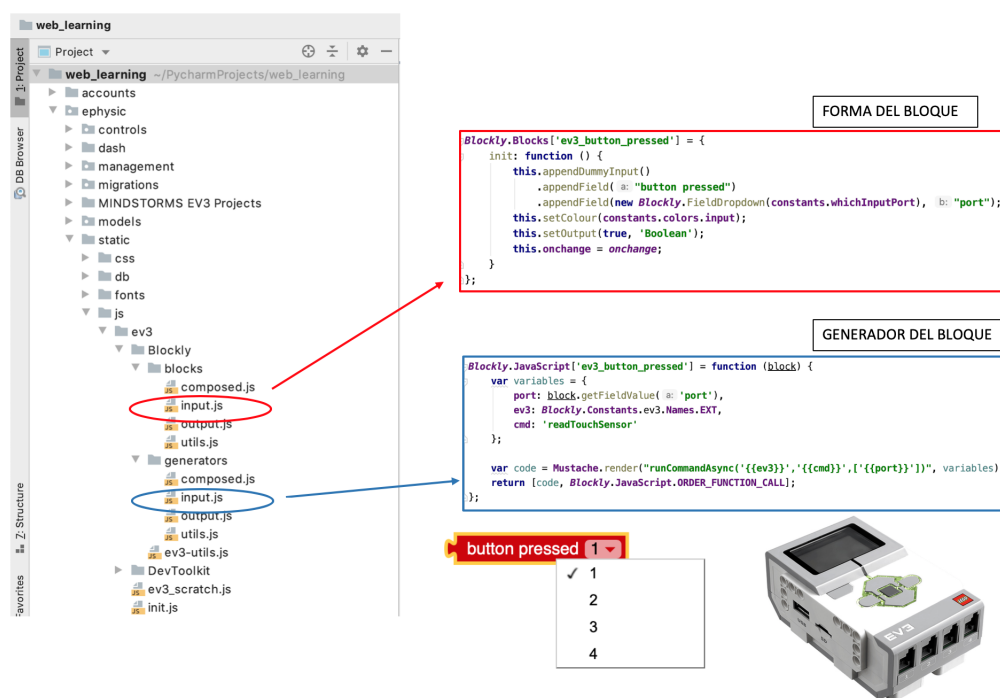


Figura 4.16: Ejemplo de creación del bloque de input del sensor de contacto. En rojo un ejemplo de programación de la gráfica del bloque y en azul el código generador.

con el robot EV3 utilizando el plug-in de Scratch; la comunicación se establece vía Bluetooth sin necesidad de modificar el firmware.

En IDEE en la carpeta *blocks* se ha definido la estructura de las gráficas de los bloques y en la carpeta *generators* se ha creado, para cada bloque definido, su generador. Para el código de generación de los bloques se ha utilizado el lenguaje de programación Python y para el código de definición de los bloques se ha utilizado el lenguaje JavaScript.

En la Figura 4.16 hay un ejemplo del bloque de input del sensor de contacto.

4.4.3. Bloques de comunicación entre el sistema y el robot

Una actividad de laboratorio robótico que se utilice en un contexto educativo y en un sistema con autoevaluación, debe garantizar que:

- los experimentos aseguren siempre los mismos resultados, en las mismas condiciones iniciales.
- El nivel de la programación necesario para el desarrollo de las actividades debe estar adaptado al nivel de conocimientos informáticos de los alumnos.

Para ello, se han creado en IDEE los bloques necesarios para la comunicación entre el robot y el sistema; un bloque permite comunicar con el robot que ejecuta

el programa en el *ladrillo* y gracias a otro bloque los datos con los resultados se visualizan en la interfaz del sistema.

Los dos bloques implementados son:

- un bloque de comunicación entre IDEE y el ladrillo EV3 que hemos llamado *connection EV3*.

Gracias a este bloque el sistema IDEE comunica con el robot, que se pone a la espera de recibir los parámetros necesarios para ejecutar el programa. Este programa se puede generar utilizando el lenguaje de bloques incluido en la web del LEGO® Mindstorms EV3.

- Un bloque de comunicación entre el robot y el sistema; que hemos llamado *message name*.

En este caso el robot envía un mensaje al sistema con el resultado que el robot ha calculado. Gracias a este bloque el alumno puede visualizar el resultado en la interfaz del sistema IDEE y no solo en el robot.

Gracias a estos bloques es posible:

- utilizar el trabajo existente sobre los bloques de programación del kit LEGO® Mindstorms EV3;
- mantener el sistema operativo del ladrillo, sin necesidad de instalación alguna;
- conseguir una mayor precisión en los resultados del laboratorio. Esta precisión se consigue al tener descargado el programa en el ladrillo. De esta manera, no hay tiempo de espera en la comunicación entre el sistema y el robot.

Los bloques implementados se han utilizado en las actividades de laboratorio de las distintas experiencias. Por ejemplo, en el laboratorio del plano inclinado el estudiante debe conseguir que el plano se incline los grados que se especifican en el enunciado de la actividad. Sin los bloques adicionales implementados, para conseguir esta inclinación, el alumno debería desarrollar un programa con los bloques disponibles en Blockly. Para ello, necesita ciertos conocimientos de programación. Puesto que el objetivo es centrar la clase en el contenido de Física de la unidad didáctica, proporcionando estos bloques adicionales, bastaría con descargar el programa en el ladrillo y utilizar para su funcionamiento los bloques de comunicación.

De esta forma, la actividad de programación se puede desarrollar en otra clase con la colaboración de otro docente, o se puede dejar como actividad de desarrollo

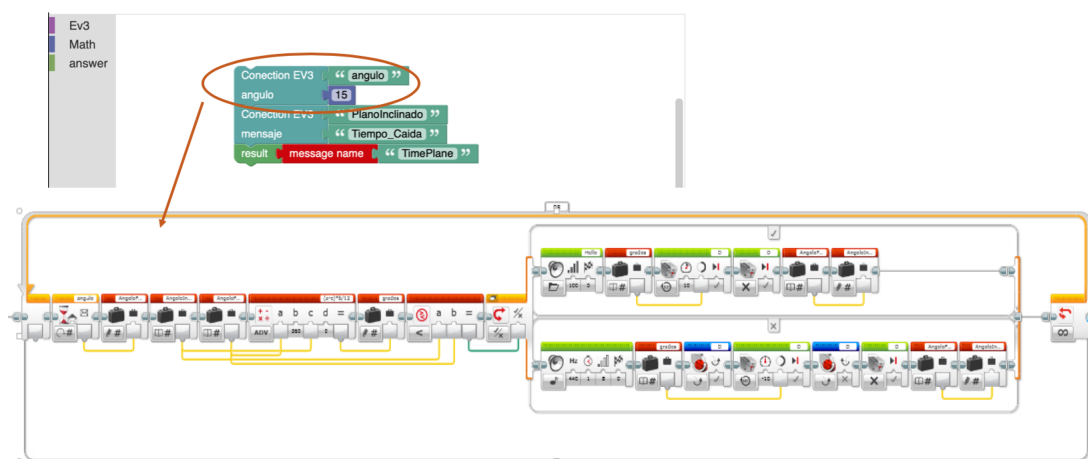


Figura 4.17: Ejemplo de uso de los bloques de comunicación.

individual para el alumno sin crear problemas de comprensión en los contenidos de física que se propone explicar.

En la Figura 4.17 se muestra un ejemplo de estos bloques para el desarrollo del laboratorio del plano inclinado y también el código que se requiere para la inclinación del plano.

4.5. Desarrollo de la interfaz del módulo del profesor

Para que el profesor pueda acceder a los análisis del aprendizaje de sus estudiantes (que se describirán en el siguiente capítulo), se ha implementado un interfaz en el lenguaje de programación de Dash. Dash es una biblioteca de Python de código abierto especialmente indicada para crear aplicaciones interactivas basadas en web. Su uso simplifica enormemente la creación de GUI (Graphical User Interface) alrededor de su código de análisis de datos.

Una aplicación Dash generalmente se compone de dos partes. La primera parte es el diseño y describe cómo se verá la aplicación y la segunda parte describe la interactividad de la aplicación.

Para la visualización de los gráficos en IDEE, se utiliza la clase `Graph` de `dash_core_components`.

4.5.1. Gráficos interactivos

El profesor necesita información sobre el proceso de aprendizaje de sus alumnos. Durante la clase, el profesor guía los alumnos en el aprendizaje y una vez que los alumnos han terminado de trabajar, el profesor puede analizar cómo han

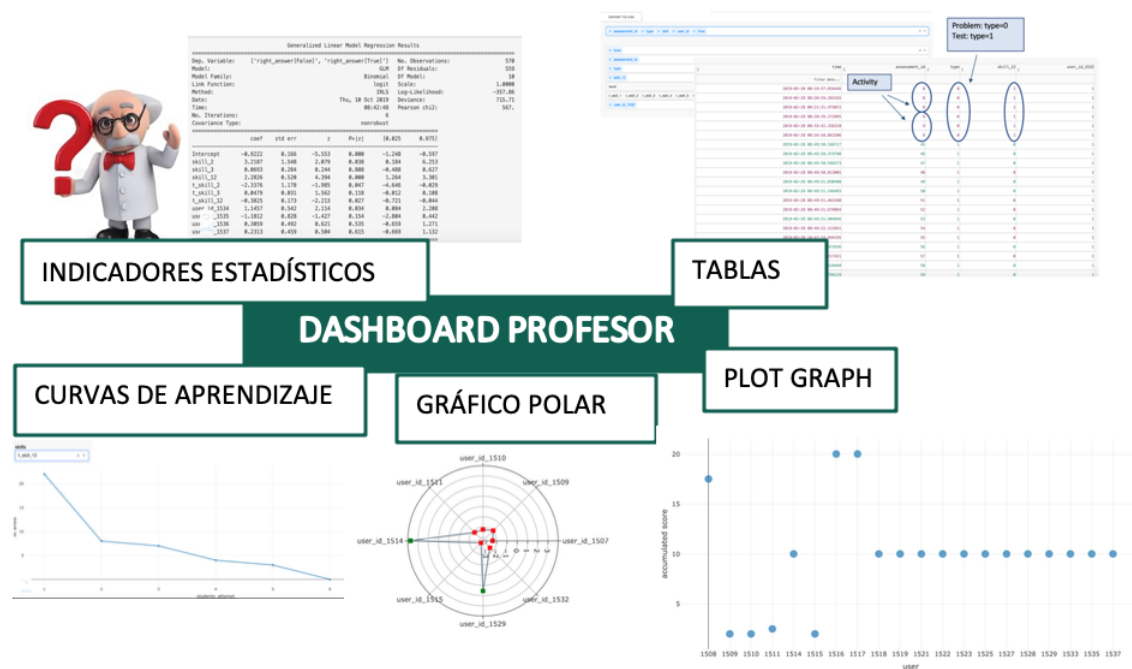


Figura 4.18: Gráficos útiles al docente para comprender el proceso de aprendizaje de los alumnos.

trabajado los alumnos.

Gracias a los gráficos interactivos, el profesor puede seleccionar el alumno o el skill que quiere analizar o visualizar gráficos de resumen del aprendizaje del grupo clase.

En IDEE se ha incluido:

- un gráfico que muestra el aprendizaje de los alumnos (*plot graph*) seleccionando el skill que interesa al profesor;
- curvas de aprendizaje que permiten al profesor visualizar si el grupo clase está aprendiendo sobre el skill seleccionando;
- tablas que resumen las actividades realizadas por el alumno y sus resultados; en este caso el profesor puede seleccionar los alumnos que le interesan, los skills y los parámetros que le interesa visualizar.

En la Figura 4.18 se muestran los gráficos creados en IDEE que ayudan al profesor en el análisis del aprendizaje de sus alumnos.

4.6. Diseño de actividades y contenidos

En el proyecto web_learning por el momento se han definido 4 experiencias: movimiento rectilíneo uniforme, movimiento rectilíneo uniformemente acelerado, plano inclinado y caída libre.



Figura 4.19: Portal de entrada a IDEE



Figura 4.20: Ejemplo de actividades en IDEE.

En la Figura 4.19 se muestra la página de entrada de IDEE desde donde se tiene acceso a las experiencias que se ofrecen al estudiante.

Una vez que el profesor haya decidido el fenómeno que se quiere analizar, el alumno puede entrar en el mundo digital que le ofrece IDEE para poder aprender con actividades de laboratorio, problemas y pruebas de conocimiento.

En la Figura 4.20 se muestra la página de IDEE una vez que el profesor haya decidido trabajar sobre el plano inclinado

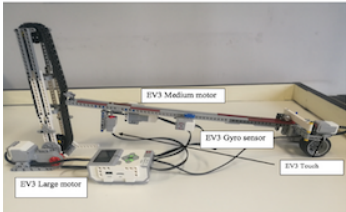
Gracias a la construcción del robot el alumno puede desarrollar conocimientos técnicos y reflexionar sobre las relaciones entre los elementos que constituyen el robot y las leyes físicas que está estudiando. En la Figura 4.21(a) se muestra la interfaz de la actividad necesaria para aprender a utilizar el robot.

Gracias a las actividades de laboratorio, el alumno puede ver una reproducción del fenómeno físico que se quiere estudiar. Con el laboratorio, el estudiante puede reproducir y visualizar el fenómeno físico descrito por el profesor. En la Figura 4.21(b) se muestra la interfaz del laboratorio de la experiencia del plano inclinado.

Los tests son de ayuda a la comprensión de las medidas físicas que intervienen en el fenómeno; ayudan a la reflexión y crean un puente entre la recogida de datos y la comprensión de la ley matemática que regula el fenómeno. En la Figura 4.21(c) se muestra la interfaz del test.

El problema final es la actividad pensada para desarrollar en los alumnos el conocimiento teórico. En este caso, la interfaz de IDEE sigue presentando un toolbox para que el alumno trabaje con la programación, pero únicamente desarrollando actividades lógico matemáticas. Los problemas que se incluyen en este tipo de actividad necesitan el uso de fórmulas matemáticas para su resolución. En este caso, el robot no puede solucionar el problema y por tanto, no está previsto que el alumno se conecte con el robot. Cuando el alumno se enfrenta con este tipo de problemas, se encuentra con la evidencia de que la experimentación es limitada a una serie de casos finitos. La experimentación tiene como objetivo ayudar el alumno a entender el problema y así, conseguir con el razonamiento, las posibles soluciones para resolverlo de forma teórica. En la Figura 4.21(d) se muestra la interfaz del problema.

APRENDEMOS A RELACIONARNOS CON NUESTRO ROBOT



Conecta el robot. Conecta en "A" el motor que gestiona la caída de la pelota; en ' al final del plano y conecta el giroscopio en la puerta "4".

Desafío1: Programa el robot, para que el plano consiga un ángulo de 10 grados.

Desafío2: Programa el robot, para que el plano baje hasta 5 grados. Visualiza el

Desafío	Tiempo
Desafío1	
Desafío2	


Invia la respuesta

▶ **Result:**

- Ev3
- Math
- answer

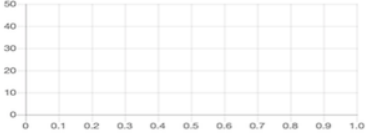
(a) Interfaz de la actividad para aprender a utilizar el robot.

GALILEO Y SUS IDEAS PRUEBA22



Respuesta incorrecta
 Respuesta correcta
 Sin respuesta
 Error en las medidas

Reflexionamos sobre la relación entre el espacio recorrido por la pelota y el tiempo d Programa el robot para que tenga una inclinación de 15 grados. Calcula el tiempo d Reflexiona sobre lo que pasaría repitiendo el experimento con una inclinación de 10



Actualiza gráfico

Incógnita	Tiempo de caída
Espacio1	
Espacio2	
Espacio3	
Espacio4	

Submit the answer

▶ **Resultado:**

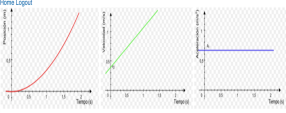
```

Ev3
  message name "TimePlane"
  Connection EV3 "angulo"
    angle
  Connection EV3 "PlanoInclinado"
    message "Tiempo_Caida"
    
```

(b) Interfaz del laboratorio del plano inclinado.

Welcome to Physics World!

Home Logout



- El movimiento acelerado de un cuerpo a lo largo de un plano inclinado está correctamente representado por las gráficas que puedes ver en esta página.
 - True False
- ¿Cuerpo la gráfica que está en el medio predice decir que la velocidad aumenta al aumentar del tiempo.
 - True False
- El movimiento de un cuerpo en caída libre está representado por las gráficas que puedes ver en esta página.
 - True False
- En el laboratorio has experimentado la relación entre el espacio recorrido y el tiempo de caída sobre un plano inclinado. Representando gráficamente los resultados sabría una gráfica como la que se encuentra en el medio.
 - True False

Back

(c) Interfaz del test.

Welcome to Physics World!

Home Logout Back

Respuesta incorrecta
 Respuesta correcta
 Sin respuesta
 Error en las medidas

LEVEL1Un cuerpo se desliza sobre un plano inclinado sin rozamiento; el plano tiene una inclinación de 60 grados respecto al plano horizon cuerpo al final del plano.(Ayúdate de los bloques para determinar la ley matemática que regula el fenómeno). Reflexiona sobre lo que cambiaría si la inclinación del plano fuera de 45 grados.

Incógnita Resultado

Velocidad

Invia la respuesta

▶ **Result:**

- Variables
- Math
- answer

(d) Interfaz del problema final.

Figura 4.21: Interfaces de las actividades de la experiencia del plano inclinado.

Capítulo 5

Análisis del aprendizaje de los estudiantes en IDEE

En este capítulo se describirá el modelo pedagógico implementado en IDEE y cómo este modelo permite ayudar al profesor en seguir el conocimiento de los alumnos.

IDEE ayuda al profesor a seguir al alumno en su aprendizaje gracias al análisis de las actividades que realiza en IDEE. En los modelos implementados a partir de este análisis se incluyen los parámetros característicos del modelo pedagógico de IDEE para mejorar el análisis del aprendizaje de los estudiantes en el contexto clase.

5.1. Obtención del nivel de conocimiento de los alumnos

La emergencia Covid-19 ha hecho más evidente la necesidad de adaptar las plataformas de aprendizaje al contexto educativo facilitando al profesor las herramientas necesarias para que puedan evaluar el proceso de aprendizaje de sus alumnos. Por ejemplo, Classroom ¹ permite a los profesores crear tests, que son evaluados por el sistema a través de guías de evaluación que el propio profesor debe introducir. Estas guías resultan limitantes cuando se quiere seguir el aprendizaje de los alumnos sobre problemas cuya resolución se centre en múltiples contenidos y habilidades. En IDEE se propone un modelo pedagógico basado en múltiples skills. La idea es que el problema que se le propone al alumno debe ser general, pero es necesario seguir al alumno en el aprendizaje de los contenidos individuales involucrados en el problema.

¹<https://support.google.com/edu/classroom>

Skill	Descripción	Score
skill_13	Movimiento rectilíneo uniformemente acelerado	10
skill_3	Relacionar magnitudes	8
skill_9	Aceleración	5

Tabla 5.1: Skills definidos en el laboratorio del plano inclinado en IDEE.

En la sección 3.3 se ha explicado que las actividades en IDEE están desarrolladas sobre skills que caracterizan los objetivos de las unidades didácticas del currículo. Las distintas actividades trabajan sobre diferentes skills; pero no todos los skills tienen la misma relevancia. Así, se ha introducido en el sistema un parámetro que determina la relevancia de cada skill en cada actividad. Se ha llamado a este parámetro *score* y se le asigna un valor entre 1 y 10.

De esta manera, si por ejemplo, se considerara la creación de la actividad de laboratorio del plano inclinado; el alumno tiene que calcular el tiempo de caída de una bola sobre un plano inclinado sobre espacios distintos con distintas inclinaciones del plano. Así, este laboratorio está relacionado con los skills mostrados en la Tabla 5.1.

Para el skill_13, el parámetro *score*=10 indica que esta actividad tiene como objetivo central el desarrollo del conocimiento del movimiento rectilíneo uniformemente acelerado.

Para el skill_3, el parámetro *score*=8 indica que esta actividad es útil para que el estudiante comprenda la relación existente entre las magnitudes espacio y tiempo de caída.

Para el skill_9, el parámetro *score*=5 indica que esta actividad trabaja también sobre el concepto de aceleración.

El nivel de conocimiento de cada alumno viene determinado calculando el *accumulated score* que representa el grado de conocimiento acumulado por el estudiante durante cada actividad y sobre cada skill. Así, cuando el estudiante da una respuesta correcta, IDEE determina el valor del *accumulated score* según las siguientes reglas:

- si la actividad es un test, el *accumulated score* es igual al *score* del skill en la actividad;
- si la actividad es un problema o un ejercicio de laboratorio, el sistema proporciona distintas respuestas al estudiante y globalmente las relaciona con uno o más skills. En este caso el *accumulated score* (AS_{ij}) para el skill i y el estudiante j se calcula de acuerdo a la ecuación 5.1:

Incógnita	Tiempo de caída
Espacio1	0.280s
Espacio2	0.45cm
Espacio3	0.4
Espacio4	0.665s

id	score	date	element_id	skill_id	user_id
1	5	2019-05-09 17:23:38.119115	2	12	8266
2	4	2019-05-09 17:23:38.126290	2	3	8266
3	2.5	2019-05-09 17:23:38.129778	2	8	8266

Figura 5.1: Ejemplo de las respuestas de un estudiante y su seguimiento en la base de datos de IDEE

$$AS_{ij} = \frac{n_j}{N_j} * W_i \quad (5.1)$$

donde n_j es el número de respuestas exactas dada por el estudiante j en la actividad, N_j es el número total de respuestas en la actividad y W_i es el parámetro *score* del skill i en la actividad.

Por ejemplo, en la respuesta del estudiante que se muestra en la Figura 5.1 hay: dos respuestas correctas (en verde), una incorrecta (en rojo) y una con medidas incorrectas (en azul); en este caso, el parámetro *score* se multiplicará por el factor $(2/4)$ ($n_j = 2$, $N_j = 4$). Ya se ha mencionado anteriormente, que el alumno tiene hasta tres intentos para realizar cada ejercicio; en este caso, el parámetro AS_{ij} se tomará con los resultados del último intento.

Para determinar el nivel de conocimiento de cada alumno se considera una media del *accumulated score* sobre todas las actividades en las que ha trabajado el estudiante. Esta media (llamada *AvgScore*) permite al sistema proporcionar al alumno una versión del problema adaptada a su nivel de conocimiento de entre las tres versiones posibles del mismo problema final: una versión sin ayuda; una versión con ayuda de organización para la resolución del sistema y una última versión con ayudas de tipo lógico y de fórmulas.

Un estudiante j con un alto nivel de conocimiento ($AvgScore_j > 0,8$) no tendrá ningún ayuda por parte del sistema. Un estudiante con un nivel medio de conocimiento ($0,5 < AvgScore_j \leq 0,8$) recibirá ayuda en los pasos lógicos para desarrollar el problema (ayuda lógica). Un estudiante con un bajo nivel de conocimiento ($AvgScore_j \leq 0,5$) recibirá ayuda también en el desarrollo de la fórmula matemática que se necesita para resolver el problema (ayuda matemática). Estos valores se han definido teniendo en cuenta los intervalos de calificaciones usados

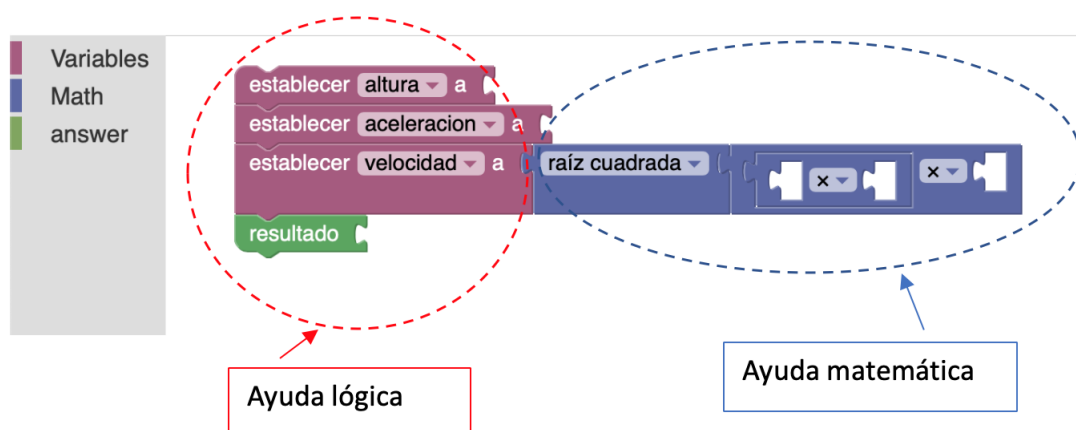


Figura 5.2: Área de trabajo de los estudiantes con un bajo nivel de conocimiento y las ayudas aportadas por el sistema

normalmente (por debajo de una calificación de 5, se considera que el estudiante necesita mejorar, entre 5 y 8 se considera que el estudiante tiene un buen nivel de conocimiento y por encima de 8, se considera que el estudiante tiene muy buen nivel de conocimiento).

En la Figura 5.2 se muestran los bloques que IDEE propone a un estudiante con un nivel de conocimiento bajo: en rojo los pasos que tiene que seguir el alumno para solucionar el problema (ayuda lógica) y en azul la estructura de la fórmula matemática que el alumno tiene que usar para poder dar una resolución teórica al problema (ayuda matemática).

5.2. Análisis de los datos con AFM

IDEA es un entorno de aprendizaje creado para ser utilizado en clase para explicar contenidos de Física usando la robótica para las actividades de laboratorio.

Todas las interacciones del alumno con el entorno se almacenan en una base de datos; para analizar estos datos se utiliza AFM (*Additive Factor Model*)[12], ampliamente utilizado para detectar el nivel de conocimiento de los estudiantes en los conceptos de un curso [15]. AFM es un algoritmo estadístico para modelar el aprendizaje y el rendimiento. Este algoritmo utiliza la regresión logística múltiple, donde la variable dependiente indica si el alumno realiza con éxito la actividad y las variables independientes representan los intentos sobre los distintos skills [56]. El AFM se basa en una Q-matriz binaria que representa las iteraciones de cada alumno con cada skill mientras realiza las actividades propuestas. A continuación se muestra un ejemplo simplificado de la Q-matriz utilizada por AFM para un estudiante:

$$\begin{pmatrix} & skill_1 & skill_2 \\ actividad_1 & 1 & 1 \\ actividad_2 & 1 & 0 \\ actividad_3 & 1 & 1 \end{pmatrix}$$

En la matriz se puede ver que el estudiante ha trabajado en el skill_1 en todas las actividades y en el skill_2 en la actividad_1 y la actividad_3.

Formalmente, dado un grupo de N estudiantes, un conjunto de K skills y un conjunto de J actividades, en el AFM, la probabilidad de que un estudiante $i \in \{1 \dots N\}$ dé una respuesta correcta en una actividad, se expresa mediante la fórmula de la ecuación 5.2.

$$p = P(Y_{ij} = 1 | \alpha_i, \beta, \gamma) = \frac{f(\alpha_i + \sum_{k=1}^K \beta_k * q_{jk} + \sum_{k=1}^K \gamma_k * q_{jk} * T_{ik})}{1 + e^{-x}} \quad (5.2)$$

donde: p es la probabilidad de éxito del estudiante en un ejercicio; Y_{ij} es la respuesta binaria del estudiante i a la actividad j ; q_{jk} es una variable binaria que indica si la actividad j usa el skill k ; α_i (*students intercept*) representa el conocimiento del alumno; β_k (*skill intercept*) representa la dificultad sobre el skill; γ_k (*skill slope*) representa la rapidez con que los estudiantes aprenden el skill k ; y T_{ik} es el número de veces que el estudiante i ha trabajado sobre el skill k .

Por tanto, en el modelo AFM, la probabilidad de éxito depende de la capacidad del estudiante i (α_i), la facilidad para conseguir el skill_ k (β_k) y la capacidad de aprendizaje sobre el skill_ k (γ_k). El parámetro *students intercept* se utiliza para cada alumno (el conocimiento inicial y final de cada alumno es diferente). El parámetro *skill intercept* se usa para cada skill (los estudiantes conocen algunos skills mejor que otros) y el parámetro *skill slope* depende del skill pero no del alumno (algunos skills son más fáciles de aprender que otros).

Para obtener el valor de los diferentes parámetros del modelo AFM, en IDEE, se utiliza el *Generalized linear model (binomial family)* (GLM) [54].

AFM se aplica generalmente dentro del marco disponible en PSLC-Datashop [30] y su uso requiere conocimientos previos sobre las técnicas involucradas. La gran mayoría de los docentes no tienen ese conocimiento previo y el reto que se plantea es permitir que los docentes puedan realizar los análisis. Para ello, en IDEE se ha incorporado este análisis en el propio panel del profesor (o *dashboard*). Además, en IDEE también se proporcionan curvas de aprendizaje [35] y

gráficos interactivos (como se verá más adelante) que pueden ser útiles para que el profesor comprenda los parámetros del AFM y los datos almacenados en la base de datos. El profesor puede visualizar el análisis estadístico del modelo y los gráficos interactivos en el dashboard del profesor. En la siguiente sección se describen los elementos de apoyo que puede encontrar el docente en el panel del profesor.

5.3. Panel del profesor

El profesor puede utilizar diferentes instrumentos para comprender el nivel de conocimiento de su grupo clase:

1. *Indicadores de aprendizaje*: los indicadores se muestran en una tabla con los coeficientes del AFM: α , β y γ mencionados en la sección 5.2. Así, el coeficiente α muestra los conocimientos de los estudiantes en los skills; gráficamente se representa utilizando un gráfico polar como el que se muestra en la Figura 5.3. El coeficiente β_k representa la facilidad del alumno para trabajar en el skill k . Así, valores altos de β_k muestran una baja dificultad inicial en el skill k ; altos valores del coeficiente γ_k indica que los alumnos son capaces aprender rápidamente el skill k . El AFM devuelve también, por cada skill, el parámetro *p_value* que indica la aceptabilidad de los resultados.

En IDEE, por cada skill k se muestran al profesor los valores estadísticamente significativos de los coeficientes (aquellos con *p_value* < 0,05). Gracias a los parámetros del AFM el profesor puede visualizar los alumnos que necesitan ayuda sobre los skills que el modelo considera estadísticamente significativos.

2. *Curvas de aprendizaje*. Las curvas de aprendizaje muestran el rendimiento del alumno teniendo en cuenta el número de interacciones [35]. Las curvas muestran el número de errores en las actividades en cada intento de todos los alumnos. En IDEE, se muestran al profesor todas las curvas de aprendizaje sobre skills estadísticamente significativos. En la Figura 5.3 se muestra un ejemplo de curva de aprendizaje de un grupo clase sobre el skill_12. El profesor, gracias a las curvas de aprendizaje, puede analizar el aprendizaje general del grupo clase sobre los skills significativos. De forma inmediata, una curva decreciente muestra que sí está habiendo un aprendizaje en el grupo clase sobre el skill seleccionado. Por el contrario, una curva creciente muestra que los alumnos tienen dificultades en el desarrollo de las actividades desarrolladas.

Learning Curves	Skill graphs	References
Regression		Polar graph

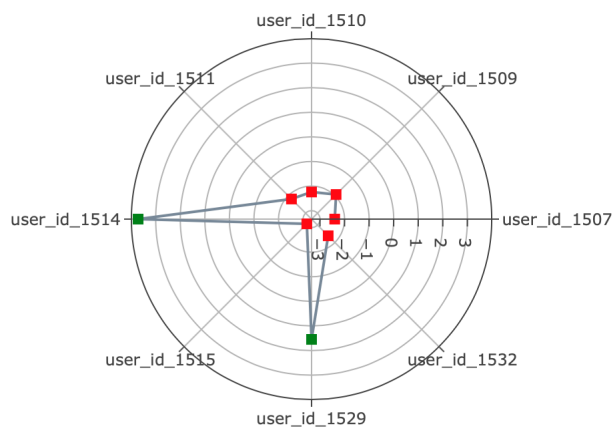


Figura 5.3: Visualización gráfica del parámetro α . Los puntos rojos representan alumnos con dificultades de aprendizaje y los puntos verde representan los alumnos significativos con buenas capacidades de aprendizaje, en los skills analizados.

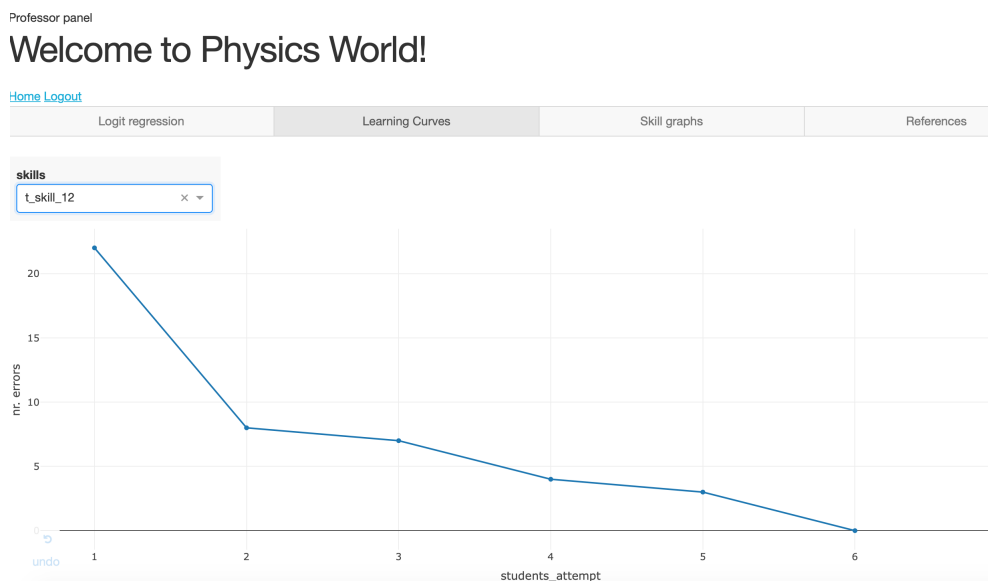


Figura 5.4: Interfaz del docente que muestra las curvas de aprendizaje al profesor. Sobre el skill mostrado el grupo clase mejora en el aprendizaje.

Professor panel

Welcome to Physics World!

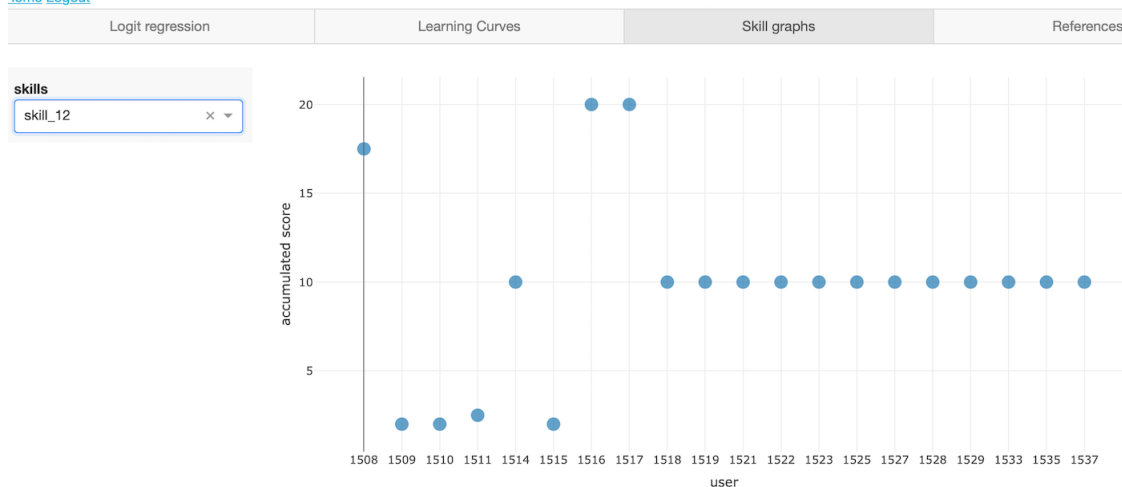
[Home](#) [Logout](#)

Figura 5.5: Interfaz del docente que muestra el conocimiento acumulado de los alumnos con la visualización del *accumulated score*

3. *Gráfico*. El profesor puede visualizar el conocimiento acumulado por cada estudiante en los distintos skills. En la Figura 5.5 se muestra el gráfico que permite al profesor ver el conocimiento acumulado de los alumnos mediante la visualización del valor del *accumulated score*. Este gráfico es útil al profesor para visualizar el aprendizaje individual de cada alumno.
4. *Tabla*. Se muestra una tabla con las actividades que el estudiante lleva a cabo, el tiempo que el estudiante trabaja en el sistema y en las distintas actividades y los errores cometidos en las distintas actividades. En la Figura 5.6 se muestra un ejemplo de tabla para un estudiante. Gracias a los datos de la tabla el profesor puede recopilar diferentes informaciones acerca de la interacción del alumno con el sistema.

En la Figura 5.7 se muestra una imagen del dashboard del módulo del profesor en IDEE, donde se visualiza toda la información a la que el profesor puede acceder para poder seguir el progreso del conocimiento de los alumnos.

Gracias a toda esta información mostrada en el dashboard el profesor puede decidir las actividades de recuperación más oportunas en cada caso. Considerando que un profesor tiene distintos grupos clases asignados cada año y que en cada grupo clase pueden necesitar ayuda individualizada distintos alumnos, puede ser una actividad onerosa para el profesor analizar toda la información proporcionada tanto en la tabla como en los gráficos, seguir el conocimiento acumulado sobre los distintos skills por cada alumno y localizar las carencias en el conocimiento de cada alumno. Siempre siguiendo la idea de ayudar al profesor en su tarea de

Professor panel

Welcome to Physics World!

Home Logout

Logit regression Learning Curves Skill graphs References

Data Regression Polar graph

EXPORT TO CSV

assessment_id type skill user_id time

time assessment_id type skill_12 user_id_1532

filter data...

Activity

Problem: type=0
Test: type=1

time	assessment_id	type	skill_12	user_id_1532
2019-02-28 08:19:37.834446	8	0	1	1
2019-02-28 08:20:59.283165	8	0	1	1
2019-02-28 08:21:31.475972	8	0	1	1
2019-02-28 08:28:29.272895	9	0	1	1
2019-02-28 08:33:42.336310	9	0	1	1
2019-02-28 08:34:58.863286	9	0	1	1
2019-02-28 08:49:58.166717	45	1	0	1
2019-02-28 08:49:58.379780	46	1	0	1
2019-02-28 08:49:58.596373	47	1	0	1
2019-02-28 08:49:58.813805	48	1	0	1
2019-02-28 08:49:51.030400	49	1	0	1
2019-02-28 08:49:51.246465	50	1	0	1
2019-02-28 08:49:51.463108	51	1	0	1
2019-02-28 08:49:51.679894	52	1	0	1
2019-02-28 08:49:51.904846	53	1	0	1
2019-02-28 08:49:52.121841	54	1	0	1
2019-02-28 10:47:24.694195	55	1	0	1
2019-02-28 10:47:24.973596	56	1	0	1
2019-02-28 10:47:25.217421	57	1	0	1
2019-02-28 10:47:25.518460	58	1	0	1
2019-02-28 10:47:25.799129	59	1	0	1

Figura 5.6: El estudiante_1532 ha trabajado sobre el skill_12 seis veces en dos actividades del tipo problema y todas las respuestas han sido incorrectas (marcadas en rojo por el sistema)

Professor panel

Welcome to Physics World!

Home Logout

1 Logit regression 2 Learning Curves 3 Skill graphs References

Data Regression Polar graph

EXPORT TO CSV

type skill t_skill user_id

Select...

Select...

type

skill_1 skill_2 skill_3 skill_7 skill_8 skill_11 skill_12

Select...

t_skill_1 t_skill_2 t_skill_3 t_skill_7 t_skill_8 t_skill_11 t_skill_12

user_id_1507 user_id_1508 user_id_1509 user_id_1510 user_id_1511 user_id_1512 user_id_1513 user_id_1514 user_id_1515 user_id_1536

tyr skill skill skill skill skill skill_j skill_j t_skill t_skill t_skill t_skill t_skill t_skill t_skill user_id_1507 user_id_1508 user_id_1509 user_id_1510

undo

0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0

4

Figura 5.7: Imagen del módulo del docente in IDEE. El profesor tiene acceso a diferentes informaciones utilizando el menú: (1) indicadores de aprendizaje; (2) curvas de aprendizaje, (3) gráficos y (4) tablas

observación del aprendizaje de los alumnos, se ha implementado en IDEE otro modelo que pueda visualizar la evolución del conocimiento del alumno. En la siguiente sección se describe dicho modelo basado en la inferencia bayesiana.

5.4. Análisis de datos con CKT

A la hora de evaluar los estudiantes, el AFM muestra los skills significativos y aquellos alumnos que tienen dificultades sobre estos skills.

Sin embargo, sería útil completar esta información proporcionando al profesor un cuadro general con las dificultades de cada alumno sobre los distintos skills. De esta manera, se ayuda al profesor a centrarse en los skills que son críticos para el alumno y poder ayudarlo así en su recuperación. De hecho, si el profesor no fuera capaz de centrar la recuperación en los conceptos esenciales para la evolución del aprendizaje del estudiante, toda la información del progreso del alumno podría resultar poco eficaz.

Con el fin de centrar la acción de recuperación únicamente sobre los skills críticos para el alumno, se ha implementado en IDEE el modelo *Conjunctive Knowledge Tracing* (CKT) basado en la inferencia bayesiana [32].

El modelo CKT, basado en el modelo de referencia *Bayesian Knowledge Tracing* (BKT)[13], permite identificar los skills dominados por el alumno y aquellos en los que necesita mejorar utilizando un modelo oculto de Markov (*Hidden Markov Model*, HMM).

El modelo utiliza cuatro parámetros para cada skill, que se ajustan a los datos de rendimiento del alumno, para poder relacionar el rendimiento con el aprendizaje.

Para cada skill, hay dos estados en el HMM, conocimiento del skill (CS) o no conocimiento del skill (\overline{CS}). Llamamos K_i la probabilidad de conocer el $skill_k$ y esta probabilidad evoluciona en el tiempo. En el modelo hay otros tres parámetros; una probabilidad de *descuido* (S) que representa la probabilidad de que el estudiante se equivoque aunque conozca el skill; una probabilidad de *adivinar* (G) que representa la probabilidad de que un estudiante responda correctamente aunque no conozca el skill, y una probabilidad de *transición* de aprendizaje (T). Estas tres probabilidades se consideran constantes en el modelo CKT [32].

En la Figura 5.8, se muestra el modelo y los parámetros mencionados anteriormente.

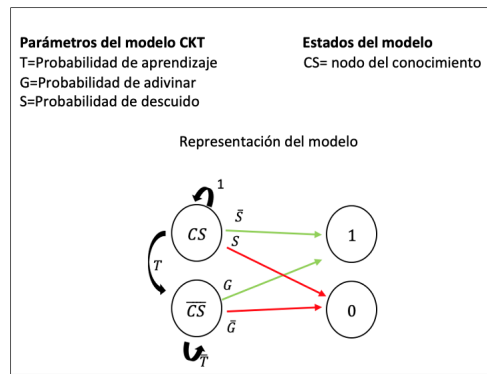


Figura 5.8: Representación del modelo tradicional CKT con descripciones de los parámetros y de los nodos.

Formalmente, dado un grupo de N estudiantes, un conjunto de K skills y un conjunto de J actividades, en el modelo CKT, la probabilidad de que un estudiante $i \in \{1 \dots N\}$ conozca el $skill_k$ habiendo cometido un error, se expresa mediante la ecuación 5.3.

$$\begin{aligned}
 &P(\text{Know} - skill_k / \text{Error}) \\
 &= P(\text{Error} / \text{Know} - skill_k) * P(\text{Know} - skill_k) / P(\text{Error}) \quad (5.3)
 \end{aligned}$$

donde la probabilidad de que el estudiante haya cometido un error, aún siendo competente en el skill, es:

$$\begin{aligned}
 &P(\text{Error} / \text{Know} - skill_k) \\
 &= S_k + (1 - S_k) * (1 - \prod_{\substack{j=1 \\ j \neq k}}^K [K_j(1 - S_j) + (1 - K_j)G_j]) \quad (5.4)
 \end{aligned}$$

y la probabilidad que el estudiante cometa un error es:

$$P(\text{Error}) = 1 - P(\text{Correct}) = 1 - \prod_{j=1}^J [K_j(1 - S_j) + (1 - K_j)G_j] \quad (5.5)$$

Un skill se considera superado por el estudiante si se consigue una probabilidad igual a 0.95 [32]. El profesor, por cada alumno, puede visualizar la evolución de aprendizaje en cada skill y entre los skills significativos y no dominados por el estudiante decidir dónde incidir para su recuperación.

5.5. Adaptación de los modelos estadísticos al modelo cognitivo de IDEE

Como se ha explicado en la sección 5.1, el modelo pedagógico incluido en IDEE estructura los objetivos de las unidades didácticas con múltiples skills y considera la relevancia de cada skill en las actividades propuestas, información que viene incluida en el parámetro *score*. Por su parte, el AFM y el CKT tienen en cuenta las interacciones del alumno sobre los diferentes skills. El siguiente objetivo es añadir como información en los dos modelos el parámetro *score*. De esta forma los resultados tendrán en cuenta la relevancia del skill en la actividad.

Additive Factor Model mejorado (eAFM)

El AFM considera las interacciones de los alumnos en diferentes skills para comprender sus procesos de aprendizaje.

Sin embargo, en el proceso de aprendizaje del alumno en el contexto educativo podría ser útil obtener información acerca de:

- la influencia en el aprendizaje de los alumnos los diferentes recursos didácticos; en concreto IDEE proporciona tests y problemas;
- la influencia en la evaluación del aprendizaje de los alumnos y la relevancia que los skills presentan en las diferentes actividades.

Para incluir esta nueva información en el análisis, se ha incluido en la ecuación matemática del AFM dos nuevos parámetros: A es el tipo de actividad (parámetro *type*) y W_j es el parámetro *score* (representando el peso que cada skill tiene en cada actividad).

Dado un grupo de N estudiantes, un conjunto de K skills y un conjunto de J actividades, en el método AFM mejorado, la probabilidad de que el estudiante $i \in \{1 \dots N\}$ dé una respuesta correcta en una actividad, se expresa con la fórmula en la ecuación (5.6):

$$\begin{aligned}
 p = P(Y_{ij} = 1 | \alpha_i, \beta, \gamma) = & \\
 & f(\alpha_i + \gamma * \mathbb{1}_A(j) + \sum_{k=1}^K \beta_k * q_{jk} * W_k + \\
 & \sum_{k=1}^K \gamma_k * q_{jk} * T_{ik}) & (5.6) \\
 f(x) = & \frac{1}{(1 + e^{-x})}
 \end{aligned}$$

$$\begin{pmatrix} & \text{skill}_1 & \text{skill}_2 & \text{respuesta} & \text{estudiante}_1 & \text{estudiante}_2 \\ \text{actividad 1} & 1 & 1 & 1 & 1 & 0 \\ \text{actividad 2} & 1 & 0 & 0 & 1 & 0 \\ \text{actividad 3} & 1 & 1 & 0 & 1 & 0 \\ \text{actividad 1} & 1 & 1 & 0 & 0 & 1 \\ \text{actividad 2} & 1 & 0 & 0 & 0 & 1 \\ \text{actividad 3} & 1 & 1 & 1 & 0 & 1 \end{pmatrix}$$

Figura 5.9: Ejemplo de la Q-matriz utilizada en el AFM.

actividad	skill_1	skill_2	estudiante_1	estudiante_2
1	score=10	score=6	true	false
2	score=4	-	false	false
3	score=3	score=7	false	true

Tabla 5.2: Ejemplo de actividades de dos estudiantes en tres actividades. Las columnas skill_1 y skill_2 muestran el valor del parámetro score de cada skill en las actividades. Las columnas student_1 y student_2 muestran si cada estudiante ha trabajado o no en cada actividad.

donde $A \subset J$ es el conjunto de tests definidos en IDEE y W_k es el parámetro score del skill k . $\mathbb{1}_A(j)$ es 1 si $j \in A$ (si j es un test) y 0 si es un problema. Con este nuevo modelo, el parámetro γ denota la dificultad del tipo de ejercicio (problema o test).

La tabla 5.2 muestra un ejemplo de cómo dos diferentes estudiantes han trabajado en tres actividades diferentes.

Con el modelo AFM se obtendría la misma Q-matriz para el *estudiante_1* y el *estudiante_2*, esto quiere decir que el modelo determinará el mismo aprendizaje para los dos estudiantes (ver Figura 5.9). Con el modelo AFM mejorado tendremos la Q-matriz mostrada en la Figura 5.10. En este caso, el *estudiante_1* da una respuesta correcta en la *actividad_1*. El valor del parámetro *score* de *skill_1* en la *actividad_1* es 1. En esta misma actividad el *estudiante_2* no responde correctamente, pero responde bien en la *actividad_3*. El valor del parámetro score de *skill_1* en la *actividad_3* es 0.3. Esto quiere decir que, sobre el *skill_1* en el modelo AFM mejorado detecta que los dos estudiantes tienen un aprendizaje diferente.

$$\begin{pmatrix} & \text{skill}_1 & \text{skill}_2 & \text{respuesta} & \text{estudiante}_1 & \text{estudiante}_2 & \text{type} \\ \text{actividad}_1 & 1 & 0,6 & 1 & 1 & 0 & 1 \\ \text{actividad}_2 & 0,4 & 0 & 0 & 1 & 0 & 0 \\ \text{actividad}_3 & 0,3 & 0,7 & 0 & 1 & 0 & 1 \\ \text{actividad}_1 & 1 & 0,6 & 0 & 0 & 1 & 1 \\ \text{actividad}_2 & 0,4 & 0 & 0 & 0 & 1 & 0 \\ \text{actividad}_3 & 0,3 & 0,7 & 1 & 0 & 1 & 1 \end{pmatrix}$$

Figura 5.10: Ejemplo de Q-matriz utilizada en el AFM mejorado (la columna *Type* indica si la actividad es un test o un problema).

Conjunctive Knowledge Tracing mejorado (eCKT)

El modelo CKT considera los parámetros S , G y T constantes; en el modelo mejorado se cambia S o G en función del parámetro $score$ atribuido a cada $skill$. Así, para cada actividad que se propone a un estudiante:

- S es la probabilidad de equivocarse conociendo el $skill$. Cuando una actividad está relacionada con un cierto $skill$ con $score$ alto, quiere decir que el alumno necesita un buen conocimiento sobre este $skill$ para que acierte en la contestación y tiene una probabilidad más alta de equivocarse. Esto quiere decir que el valor de S tiene que ser tanto mayor cuanto más relevante es el $skill$ en la actividad que está desarrollando el alumno.

$$S = \begin{cases} (score/10) * 0,05 & score \neq 10 \\ 0,05/10 & score = 10 \end{cases} \quad (5.7)$$

- G es la probabilidad de acertar sin conocer el $skill$. G representa para el alumno la probabilidad de adivinar la respuesta sin tener conocimiento; esta probabilidad tiene que ser tanto más baja cuanto más relevante es el $skill$ en la actividad que está desarrollando el alumno.

$$G = \begin{cases} (1 - score/10) * 0,2 & score \neq 0 \\ 0,2/10 & score = 0 \end{cases} \quad (5.8)$$

La Tabla 5.3 muestra, en la parte:

- izquierda; un ejemplo de datos de como un estudiante ha trabajado en cuatro actividades diferentes;
- derecha; los resultados del modelo CKT y del modelo mejorado eCKT.

Actividad	datos				Resultados			
	skill_1	skill_2	skill_3	respuesta	CKT	eCKT		
					skill_1	skill_2	skill_1	skill_2
					0.500	0.500	0.500	0.500
1	4	1	0	false	0.843	0.869	0.902	0.984
2	0	0	1	true	0.843	0.869	0.902	0.984
3	2	8	0	true	0.964	0.975	0.869	0.999
4	2	8	0	true	0.892	0.975	0.856	0.999

Tabla 5.3: Ejemplo de desarrollo del aprendizaje en un alumno que trabaja en cuatro actividades con los modelos CKT y eCKT.

En la parte izquierda de la tabla que se refiere a los datos, se puede observar que el alumno acierta en las actividad_3 y actividad_4. Para el skill_1, que no es

fundamental en estas actividades (el parámetro *score* es 2), en la parte derecha de la tabla con los resultados del Modelo CKT y del Modelo mejorado eCKT se observa que cuando el alumno responde de forma correcta en la actividad_3, la introducción del parámetro *score* en el modelo determina una disminución en el resultado (0.869) respecto al valor 0.964 del modelo CKT.

Este comportamiento es el esperado, ya que el valor del parámetro *score* del skill_1 es bajo. Por tanto, cuando el alumno acierta, el modelo considera de menor relevancia el conocimiento adquirido por parte del alumno.

Para el skill_2; que es fundamental en estas actividades (el parámetro *score* es 8); en la parte derecha de la tabla con los resultados del Modelo CKT y del Modelo mejorado eCKT se observa que cuando el alumno responde de forma correcta en la actividad_3, la introducción del parámetro *score* en el modelo determina un aumento en el resultado (0.999) respecto al valor 0.975 del modelo CKT. Este comportamiento es el esperado, ya que el valor del parámetro *score* del skill_2 es alto; el acertar por parte del alumno sirve como confirmación del conocimiento del alumno sobre este skill.

Capítulo 6

Evaluación

En este capítulo se describirá el proceso de evaluación que se ha llevado a cabo con IDEE. El primer objetivo de la evaluación era validar que el sistema podía ser utilizado en clase y que las herramientas robóticas incluidas en el sistema no proporcionaban dificultades para los alumnos, y podrían ser consideradas herramientas de interés para el estudio experimental de la física.

Comprobada la eficacia del sistema en clase el siguiente objetivo de la evaluación ha sido comprobar que el sistema fuera capaz de seguir el desarrollo del conocimiento en los alumnos en un ambiente de clase no estructurado.

6.1. Objetivos

A partir de mi experiencia docente como profesora de Matemáticas y Física en institutos de educación secundaria, se ha diseñado e implementado una propuesta metodológica que ve al alumno como centro de su propio aprendizaje.

Los diferentes recursos que engloban la propuesta presentada, como se ha descrito en los capítulos 3 y 4, tienen una finalidad educativa y se vinculan a la programación didáctica y al currículo vigente.

La propuesta metodológica ha sido evaluada en contextos educativos reales, recogiendo datos para ver como se pudieran beneficiar tanto alumnos como profesores de los recursos que se proporcionan en IDEE.

Los objetivos que se han planteado en el proceso de evaluación de IDEE son:

- comprobar la validez de este sistema en la práctica docente diaria, en un contexto educativo real, y orientado a un alumnado de educación secundaria.
- Validar si la herramienta robótica incluida en el sistema supone una mejora del entorno de aprendizaje de la física.

- Validar que la modalidad de inclusión de la herramienta de programación en IDEE no es un obstáculo al aprendizaje de la física sino una herramienta útil y necesaria para el desarrollo del laboratorio.
- Integrar el proceso de evaluación del sistema dentro del propio proceso de aplicación.
- Analizar si este tipo de sistemas ayuda a mejorar el análisis del proceso de aprendizaje de los alumnos por parte del profesor.

Para conseguir estos objetivos se han llevado a cabo tres experiencias en contextos educativos reales cuyo análisis de datos y objetivos se detallarán en la siguiente sección.

6.2. Experimentación en clase con IDEE

Con el sistema creado se han realizado tres experimentaciones en tres institutos diferentes en Madrid. La primera experimentación se realizó para verificar la funcionalidad del sistema y se hizo en Abril del 2018 en la escuela italiana de Madrid E. Fermi. En esta primera experimentación en IDEE solo se había implementado la experiencia del plano inclinado que proporcionaba a los alumnos una actividad de laboratorio, dos tests y un problema final. La experiencia en clase fue satisfactoria y nos dio ideas válidas para hacer que el plano robótico fuera más eficiente y, sobre todo, nos hizo reflexionar sobre la necesidad de incluir nuevas experiencias en el sistema.

La segunda experimentación en clase se realizó en Mayo 2018 en el instituto Cabrini de Madrid. Se implementaron otras tres experiencias en IDEE que permitieron el uso de diferentes tipos de sensores del robot y analizar el tema del estudio del movimiento rectilíneo de una manera más completa. La experiencia fue positiva y gracias a esta experimentación fue posible averiguar que algunos resultados evaluados como no correctos por el sistema no fueron causados por la falta de comprensión de los alumnos, sino por retrasos en la comunicación entre la programación en IDEE y la ejecución del robot. Este retraso no permitía que el sistema respetara el error del 3% de aceptabilidad de la solución en las actividades de laboratorio. Esta experimentación nos ha sido útil para poder adecuar las herramientas robóticas y sus funcionalidades al contexto educativo. Para solucionar el problema, se crearon los bloques de comunicación entre el sistema y el robot analizados en la sección 4.4.3 que permitieron que el programa se ejecutara en el ladrillo.

La última experimentación se realizó en Febrero 2019 en el instituto GSD-Las Suertes de Madrid. Los alumnos mostraron interés y participación, lo que

también nos permitió analizar la precisión del sensor de distancia con la participación activa de los estudiantes. El funcionamiento de IDEE nos dejó satisfechos y decidimos dedicar nuestros esfuerzos al análisis de datos. Los datos recopilados durante esta última experiencia se han utilizado como parte experimental del artículo [43] que analiza la validez del AFM como un apoyo para el profesor en el análisis de los contenidos aprendidos por los alumnos durante sus explicaciones.

La experiencia en el colegio Italiano de Madrid se planteó para evaluar la conveniencia de un sistema con laboratorio robótico y las mejoras que pudiera aportar en el contexto educativo. Los datos recopilados en esta experiencia han sido útiles para comprobar la validez de los modelos eAFM y eCKT para el seguimiento del conocimiento de los alumnos.

Por otra parte, la experiencia en el instituto Cabrini de Madrid ha sido útil para averiguar las dificultades de los estudiantes en las actividades planteadas.

Todas las experiencias en contextos educativos reales han sido útiles para comprobar la validez de IDEE en la práctica docente diaria.

El análisis estadístico de los datos recopilados en dichas experiencias se mostrará en la siguiente sección.

6.2.1. Evaluación de las experiencias en clase

El entorno educativo propuesto IDEE ha sido probado con éxito en la escuela Italiana de Madrid. IDEE se ha utilizado con un grupo de 15 estudiantes de 13/14 años; sin ninguna selección de preferencia de estudio, de conocimiento previo o sexo. La actividad propuesta duró 2 horas con cinco kits EV3 trabajando solo con la experiencia del plano inclinado implementado en IDEE.

La experiencia hecha en el Liceo Italiano de Madrid fue pensada para verificar la validez del sistema en el aprendizaje de una unidad didáctica de física y para comprobar si el sistema creado pudiera ser útil para motivar a los estudiantes.

Antes de la experiencia, se presentó una prueba previa para conocer el nivel inicial de preparación de los estudiantes.

Los resultados de la prueba preliminar muestran que los estudiantes presentan dificultades en los temas del movimiento rectilíneo uniformemente acelerado y cuando el ejercicio propuesto requiere relacionar los conceptos de velocidad y espacio con el movimiento rectilíneo uniformemente acelerado (MRUA).

En la Figura 6.1 se muestra el porcentaje de errores de los estudiantes y en la Figura 6.2 se muestra la distribución porcentual de errores en los temas de movimiento rectilíneo uniforme, caída libre y cómo se relacionan estos conceptos. Los estudiantes 1, 2, 3 y 8 realizan el 100 % de los errores en el MRUA y al menos el 50 % en los ejercicios de relación conceptual.



Figura 6.1: El gráfico muestra el porcentaje de error de los alumnos en su conocimiento inicial.

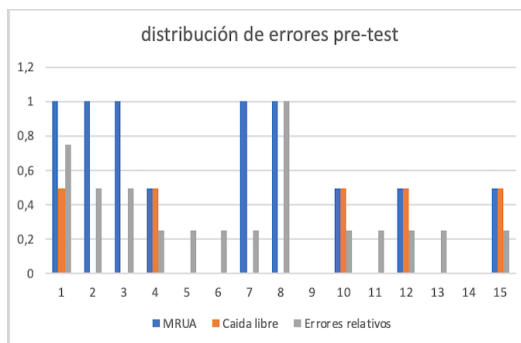


Figura 6.2: El gráfico muestra el porcentaje de error de los estudiantes en los diferentes temas evaluados en el pre-test.

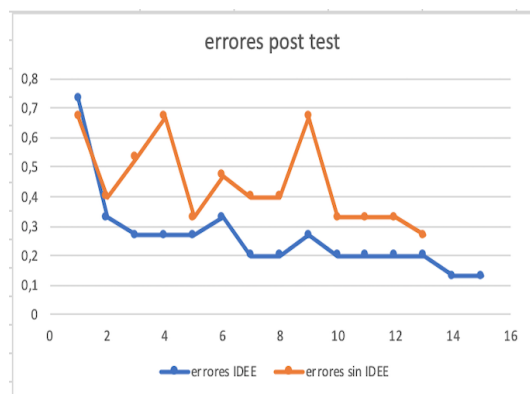


Figura 6.3: El gráfico muestra el porcentaje de error en el post-test para los dos grupos.

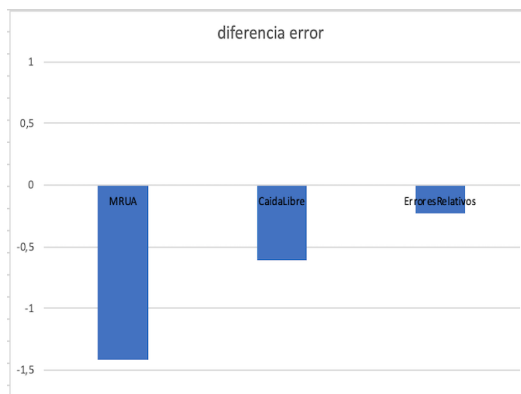


Figura 6.4: El gráfico muestra la diferencia entre las medias de los dos grupos sobre el número de errores.

Después de la experiencia, se presentó un post-test a los estudiantes para evaluar el nivel de aprendizaje adquirido sobre el movimiento rectilíneo uniformemente acelerado.

El mismo post-test se presentó a un grupo de 13 estudiantes que siguieron una clase tradicional sobre los temas tratados.

En la Figura 6.3 se muestra el porcentaje de error en el post-test para los dos grupos. Los estudiantes que han trabajado con IDEE muestran que han entendido mejor los temas estudiados; la significación estadística de esta afirmación se demuestra con una prueba t-student ($p \leq 0.05$).

En la Figura 6.4 el gráfico expresa la diferencia entre las medias de los dos grupos sobre el número de errores. Es posible observar cómo el uso de IDEE (representado en la parte positiva del gráfico) permite a los estudiantes estar más preparados cuando tienen que relacionar el MRUA con la caída libre o el plano inclinado.

En la Figura 6.5 se muestra el interés en el estudio de la física del grupo clase.

Es posible encontrar como dato relevante que la clase considera difícil el estudio de la física.

Para determinar el grado de cumplimiento de los objetivos propuestos en esta experiencia, los resultados han sido analizados mediante encuestas a los estudiantes que nos han permitido obtener información para medir sus conocimientos previos en robótica e informática, las dificultades encontradas en el desempeño de esta experiencia en relación con los objetivos planificados, su nivel de interés y su grado de satisfacción.

Los resultados de la investigación, presentados en la Figura 6.6 son alentadores, el grupo de clase no tiene grandes conocimientos iniciales de robótica y programación y considera que la experiencia no ha sido compleja pero sí interesante. En general, los estudiantes están satisfechos con la experiencia.

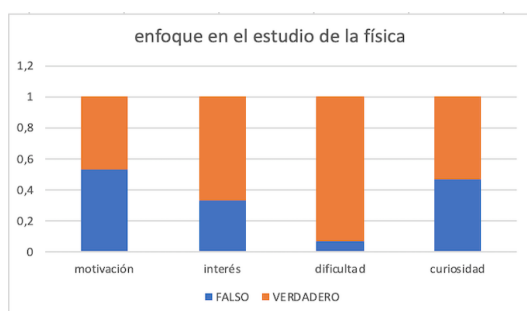


Figura 6.5: El gráfico muestra el interés del grupo clase en la asignatura de física.



Figura 6.6: Gráfico para determinar el grado de cumplimiento de los objetivos propuestos en esta experiencia. El alumno puede expresar su opinión respondiendo con un número entre 1 y 5.

La experimentación en el instituto GSD-Las Suertes se hizo después de haber incluido en IDEE cuatro experiencias y los bloques de comunicación entre el sistema y el ladrillo. Estas modificaciones se hicieron después de la experiencia en el colegio Cabrini, cuyo análisis de datos permitió averiguar problemas de retraso de comunicación entre el sistema y el robot que pudieran causar errores en las contestaciones de las actividades de laboratorio. La última experiencia en clase nos dejó satisfechos, ya que fue posible dar una clase completa de física utilizando IDEE con actividades de laboratorio robótico.

6.2.2. Análisis del aprendizaje en IDEE

Las experiencias llevadas a cabo nos ha permitido analizar si los resultados de los modelos de análisis del aprendizaje obtenidos utilizando las técnicas explicadas en el capítulo anterior son de ayuda al profesor para evaluar el progreso de aprendizaje de los alumnos.

Los datos cuantitativos de la base de datos son necesarios para la determinación de la Q-matriz que utiliza el método AFM y el método CKT.

A continuación, se detallan los resultados de los modelos implementados y cómo estos proporcionan información útil al profesor para poder detectar el conocimiento adquirido por parte de los alumnos durante las actividades.

Experimentación en el Liceo Italiano E.Fermi de Madrid

En el caso de los estudiantes del colegio Italiano la tabla *data* se compone de 257 entradas y la tabla *perfil* de 250.

La Q-matriz de la experiencia en el colegio Italiano está compuesta por 242 entradas. Cada entrada está compuesta por el *estudiante_id*, el *skill* en el que él/ella está trabajando y el número de intentos en cada *skill* que él/ella ha hecho previamente.

Los resultados significativos que se obtienen aplicando el eAFM se resumen en la Tabla 6.1. Cuanto mayor es el parámetro *skill_intercept*(β), menor es la dificultad inicial que tiene el alumno sobre este *skill*. Cuanto mayor sea el parámetro *slop*(γ) de cada *skill*, más rápido aprenderán los estudiantes sobre este *skill*. Con los resultados obtenidos se comprueba que el *skill_3* (Relacionar magnitudes), *skill_1* (Interpretación gráfica) y *skill_10* (Velocidad) son *skills* sobre los cuales los estudiantes tienen poco conocimiento. El *skill_7* (Movimiento uniformemente acelerado) es la habilidad más fácil con $\beta=5.66$. El *skill_10* con $\gamma=8.1$ permite observar que la práctica con IDEE mejora el conocimiento de este *skill*. La probabilidad de éxito en este caso es $p=0.99$; esto quiere decir que IDEE ayuda en el conocimiento de este *skill*, ya que inicialmente los alumnos tienen poco conocimiento y al final tienen una alta probabilidad de acertar. También para el *skill_7* la probabilidad de éxito es alta ($p=0.88$), pero para el conocimiento inicial del estudiante y no para la práctica con el sistema.

El profesor puede analizar el grupo clase a través de los gráficos significativos presentes en la dashboard del profesor en IDEE.

En la Figura 6.11 se visualizan algunas curvas de aprendizaje en habilidades significativas; el profesor observa que hay una disminución en los errores en *skill_3* (los errores disminuyen durante la realización de actividades); incertidumbres en *skill_1* y aumento de errores en la *skill_7* y *skill_10*.

En la Figura 6.14, el profesor observa a los alumnos con dificultades en rojo y a los alumnos con buen aprendizaje en verde.

El profesor puede analizar el recorrido del conocimiento individual de los alumnos. Un posible análisis que el profesor puede hacer al observar el aprendizaje del estudiante con ID 779 en IDEE es:

skill_id	skill	(β)	$\text{Pf>(> z)$	(γ)	$\text{Pr>(> z)$	P
skill_1	Inter. gráfica	0.61	0.05	-9.61	0.022	0.00
skill_3	Rel. magnitudes	0.43	0.033	-0.94	0.004	0.12
skill_7	Mov. acelerado	5.66	0.002	-2.56	0.000	0.83
skill_10	Velocidad	0.75	0.006	8.1	0.001	0.99

(a) El parámetro β (skill intercept) es la dificultad inicial para cada skill, el parámetro γ (slope) es la tasa de aprendizaje de cada habilidad y P es la probabilidad de aprendizaje.

student	student intercept (α)	$\text{Pf>(> z)$
user_id_779	-1.79	0.009
user_id_780	1.98	0.021
user_id_788	-1.16	0.045

(b) El parámetro de intercepción de habilidades para estudiantes significativos (en negrita se muestran aquellos estudiantes con bajo aprendizaje).

Tabla 6.1: Indicadores del proceso de aprendizaje. La intersección de habilidades es el nivel de dificultad inicial para cada habilidad, la pendiente es la tasa de aprendizaje y la probabilidad es la probabilidad de éxito de usar una habilidad una vez que la experiencia ha terminado.

- Al analizar el *accumulated score* en los skills significativos, el estudiante tiene valores bajos, quiere decir que el alumno ha acumulado poco conocimiento. En la Figura 6.13 se muestra el score acumulado del alumno en el skill_3.
- El estudiante tiene dificultades en los temas tratados, aunque el sistema lo ha ayudado con un problema de nivel 3 (con ayudas matemáticas y lógicas), el estudiante no tiene el conocimiento adecuado para resolver las actividades. En la Figura 6.12 se muestra una tabla con un resumen de las actividades realizadas por el alumno en IDEE.
- El profesor debe actuar con acciones de recuperación para este estudiante.

Gracias a los resultados del método eCKT el profesor puede visualizar el conocimiento adquirido por los estudiantes en los diferentes skills significativos del eAFM. Los resultados del modelo para el estudiante con ID 779 se pueden observar en la Tabla 6.2.

Teniendo en cuenta que en el modelo eCKT una habilidad se considera dominada cuando se obtiene el valor de 0.95, el profesor se centrará en los skills donde el estudiante no ha conseguido este valor; en el caso del estudiante con ID 779 el profesor centrará la recuperación del alumno en el skill_1 y el skill_10. El profesor proporcionará al alumno nuevas actividades sobre la interpretación gráfica (skill_1) y sobre el concepto de velocidad (skill_10).

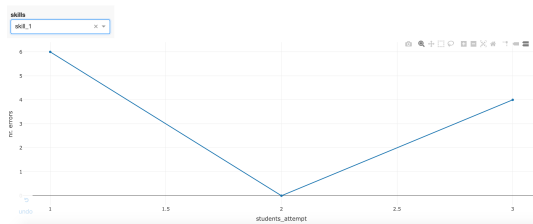


Figura 6.7: Curva de aprendizaje skill_1.

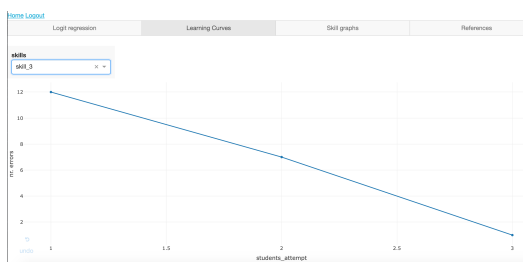


Figura 6.8: Curva de aprendizaje skill_3.

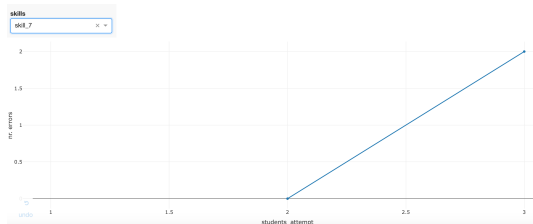


Figura 6.9: Curva de aprendizaje skill_7.

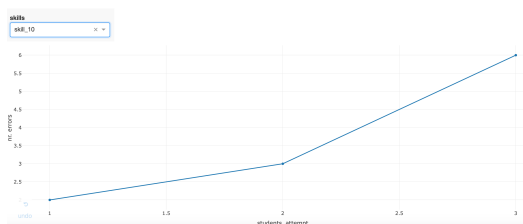


Figura 6.10: Curva de aprendizaje skill_10.

Figura 6.11: Curvas de aprendizaje

time	assessment_id	type	skill_3	level	t_skill_3	user_id_779
2	0	0	0	1	0	1
3	0	1	1	1	1	1
3	0	1	1	1	2	1
6	1	1	1	1	3	1
9	1	0	1	1	0	1
10	1	0	1	1	0	1
11	1	1	1	1	4	1
12	1	1	1	1	5	1
13	1	1	1	1	6	1
14	1	0	1	1	0	1
15	1	1	1	1	7	1
16	1	1	1	1	8	1
17	1	1	1	1	9	1
7	0	0	0	3	0	1
7	0	0	0	3	0	1
7	0	0	0	3	0	1

Figura 6.12: Resumen de las actividades realizadas en IDEE por el alumno con ID 779.

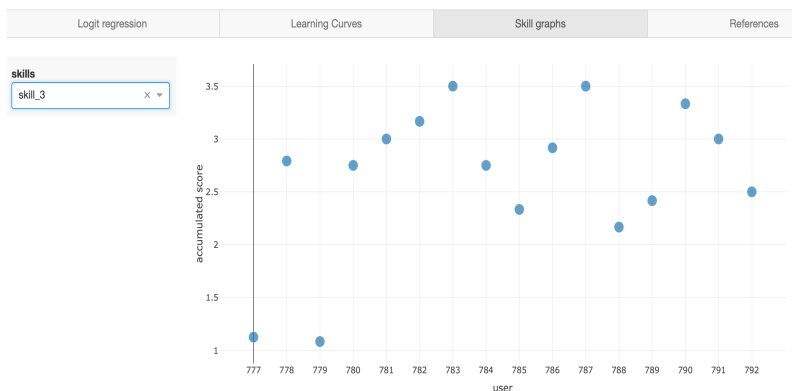


Figura 6.13: Análisis gráfico de la puntuación del alumno en skill_3.

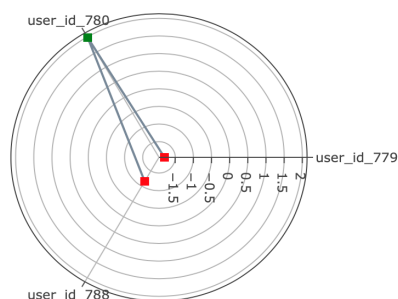


Figura 6.14: Análisis gráfico del parámetro de intercepción α del estudiante con un gráfico Polar .

actividad	skill_1	skill_3	skill_7	skill_9	skill_10
1	0.5	0.5	0.5	0.5	0.5
2	0.5	0.27	0.5	0.5	0.5
3	0.5	0.23	0.5	0.5	0.5
4	0.5	0.90	0.95	0.97	0.5
5	0.5	0.90	0.99	0.97	0.5
6	0.5	0.90	0.98	0.97	0.5
7	0.5	0.87	0.97	0.97	0.32
8	0.5	0.99	0.99	0.99	0.90
9	0.5	0.98	0.99	0.99	0.67
10	0.19	0.98	0.99	0.99	0.67
11	0.93	0.99	0.99	0.99	0.67
12	0.65	0.99	0.99	0.99	0.67
13	0.39	0.98	0.99	0.99	0.67
14	0.39	0.98	0.99	0.99	0.63
15	0.39	0.98	0.99	0.99	0.62

Tabla 6.2: Resultados del modelo eCKT para el estudiante con ID 779

skill_id	skill	(β)	Pf($> z $)	(γ)	Pr($> z $)
skill_2	Sistema de referencia	-4.23	0.026738	3.2	0.022850
skill_12	Conocimiento en robótica	-4.16	3.26e-09	0.84	0.000111

(a) El parámetro β (skill intercept) es la dificultad inicial para cada skill, el parámetro γ (slope) es la tasa de aprendizaje de cada habilidad.

Estudiante	student intercept (α)	Pf($> z $)
user_id_1507	-2.39929	0.004915
user_id_1509	-1.92893	0.024076
user_id_1510	-2.24017	0.005639
user_id_1511	-2.18629	0.006486
user_id_1514	3.71865	0.001396
user_id_1515	-3.07097	0.000200
user_id_1529	1.54727	0.037535
user_id_1532	-2.38468	0.003317

(b) Resultados del parámetro student intercept para estudiantes significativos (en negrita se muestran aquellos estudiantes que aprenden poco).

Tabla 6.3: Indicadores del proceso de aprendizaje.

Experimentación en el instituto GSD-Las Suertes de Madrid

IDEE se ha utilizado en una clase de física en la escuela GSD-Las Suertes con 30 estudiantes de entre 16 y 17 años y supervisado por un profesor de matemáticas y física de educación secundaria. El objetivo de estas experiencias era desarrollar en los estudiantes la capacidad de describir el movimiento rectilíneo de un cuerpo utilizando las ecuaciones que conectan el espacio, la velocidad y el tiempo. La actividad duró dos horas; cada alumno trabajó con su ordenador portátil y realizó las actividades de laboratorio en grupo. IDEE procesa los datos almacenados de esta experiencia y completa una tabla derivada, hecha de 570 entradas. Cada entrada está compuesta por el estudiante_id, el skill en el que él/ella está trabajando y el número de intentos en cada skill que él/ella ha hecho previamente. Esta es la tabla utilizada para aplicar los modelos estadísticos AFM y CKT.

IDEE analiza los datos con el AFM cuyos resultados significativos se resumen en la Tabla 6.3a y en la Tabla 6.3b.

De los resultados en la Tabla 6.3a, gracias a los resultados estadísticos en la columna skill intercept (β), se puede observar que:

- skill_2 (Sistema de referencia) con $\beta = -4,23$; los estudiantes tienen dificultades cuando los conceptos estudiados están relacionados con el concepto de sistema de referencia.
- skill_12 (Conocimiento en robótica) con $\beta = -4,16$; los estudiantes pre-

sentan dificultades iniciales cuando interactúan con el sistema IDEE y el robot.

En la misma Tabla 6.3b, la columna de la pendiente o slope (γ) permite analizar la evolución en el conocimiento que tiene el alumno al interactuar con el sistema:

- skill_2 con $\gamma = 3,2$, los estudiantes aprenden rápidamente.
- sobre skill_12 with $\gamma = 0,84$, los estudiantes aprenden lentamente. Con el uso continuo de IDEE en el aula, se espera una mejora de esta habilidad.

Además, de acuerdo con los resultados mostrados en la Tabla 6.3b, el análisis del parámetro α muestra un grupo de 6 estudiantes con dificultades de aprendizaje sobre las habilidades analizadas en la tabla, y dos estudiantes que no presentan dificultades.

Como se ha descrito en la sección 5.2, en IDEE los resultados estadísticos del análisis se muestran gráficamente al profesor mediante curvas de aprendizaje y gráficos significativos. El objetivo principal de las curvas de aprendizaje en IDEE es visualizar el progreso o las dificultades del grupo clase en las habilidades analizadas. Los gráficos y las tablas permiten entender más en detalle lo que ha pasado individualmente a los estudiantes. Como ejemplo, en la Figura 6.18, se muestra la curva de aprendizaje que visualiza los primeros 6 intentos para todos los estudiantes en skill_12. Los estudiantes muestran una dificultad inicial alta (gran cantidad de errores). Durante las actividades, el número de errores disminuye, por lo que la curva es suave y decreciente. Así, en esta habilidad, los estudiantes mejoran en su aprendizaje. Considerando el gráfico en la Figura 6.15, y de acuerdo con nuestros resultados, un grupo de 6 estudiantes presentó dificultades de aprendizaje (puntos rojos) sobre las habilidades analizadas, y dos estudiantes no presentaron dificultades (puntos verdes).

Además, mirando el gráfico en la Figura 6.16, el profesor puede inspeccionar el valor *accumulated score* para el skill_12 de aquellos estudiantes con dificultades. Los estudiantes con IDs 1509, 1510, 1511 y 1515 tienen una puntuación baja, mientras que los estudiantes con IDs 1507 y 1532 no tienen valor para *accumulated score*.

Al seleccionar estos estudiantes en la tabla de interacción del estudiante, en la Figura 6.17, el profesor puede ver que el estudiante con ID 1507 no ha trabajado en las actividades pero el estudiante con ID 1532 ha trabajado pero nunca ha dado las respuestas correctas. Con esta información, el profesor puede decidir ofrecer ayuda adicional o supervisar a ciertos estudiantes más estrechamente.

Learning Curves	Skill graphs	References
Regression		Polar graph

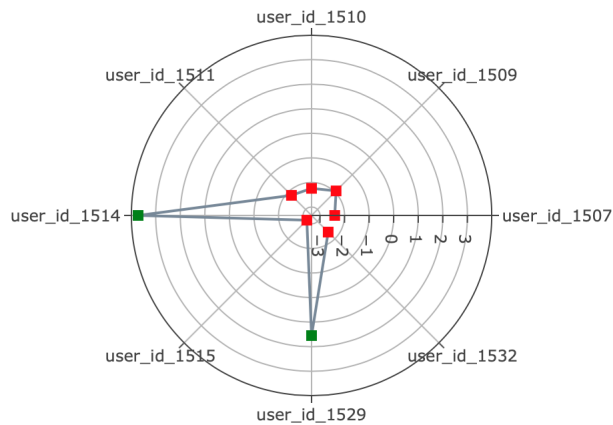


Figura 6.15: Visualización gráfica del parámetro α . Los puntos rojos representan alumnos con dificultades de aprendizaje y los puntos verdes representan los alumnos significativos con buenas capacidades de aprendizaje en los skills analizados.

Professor panel

Welcome to Physics World!

[Home](#) [Logout](#)

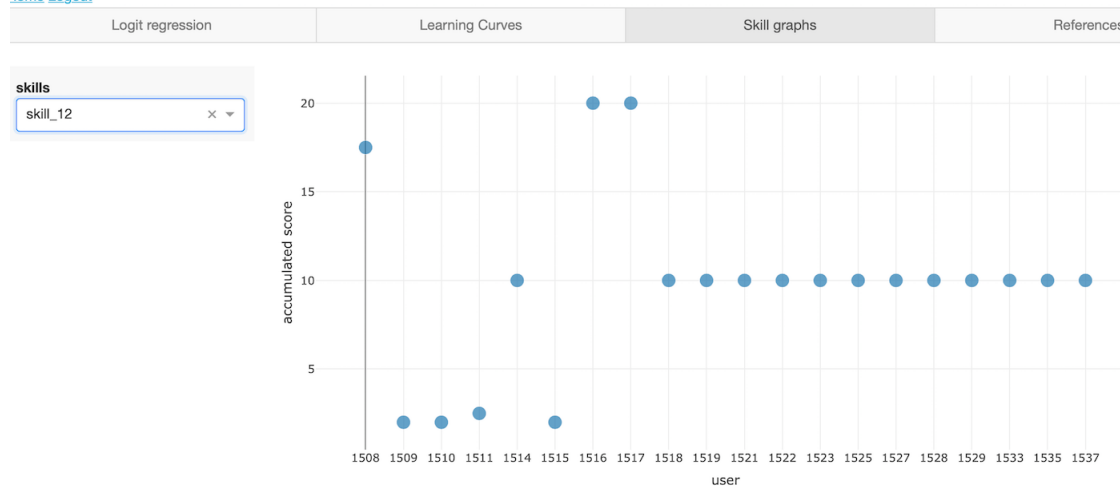


Figura 6.16: Interfaz del docente que muestra el conocimiento acumulado de los alumnos con la visualización del *accumulated score*

time	assessment_id	type	skill_12	user_id_1532
2010-02-20 00:10:37.020440	8	1	1	1
2010-02-20 00:10:38.912010	8	1	1	1
2010-02-20 00:11:31.475972	8	1	1	1
2010-02-20 00:12:29.272865	8	1	1	1
2010-02-20 00:13:42.138318	8	1	1	1
2010-02-20 00:14:18.483206	8	1	1	1
2010-02-20 00:15:18.156717	21	1	1	1
2010-02-20 00:15:18.370768	46	1	1	1
2010-02-20 00:15:18.595373	47	1	1	1
2010-02-20 00:15:18.513065	48	1	1	1
2010-02-20 00:15:18.630488	49	1	1	1
2010-02-20 00:15:18.246465	58	1	1	1
2010-02-20 00:15:18.463388	51	1	1	1
2010-02-20 00:15:18.670804	52	1	1	1
2010-02-20 00:15:18.984846	53	1	1	1
2010-02-20 00:15:18.121841	54	1	1	1
2010-02-20 10:47:18.494395	55	1	1	1
2010-02-20 10:47:18.919998	56	1	1	1
2010-02-20 10:47:19.217421	57	1	1	1
2010-02-20 10:47:21.118468	58	1	1	1
2010-02-20 10:47:21.799129	59	1	1	1

Figura 6.17: El alumno ha trabajado en skill_12 seis veces en dos actividades tipo problema y todas las respuestas han sido incorrectas (todas están marcadas en rojo).

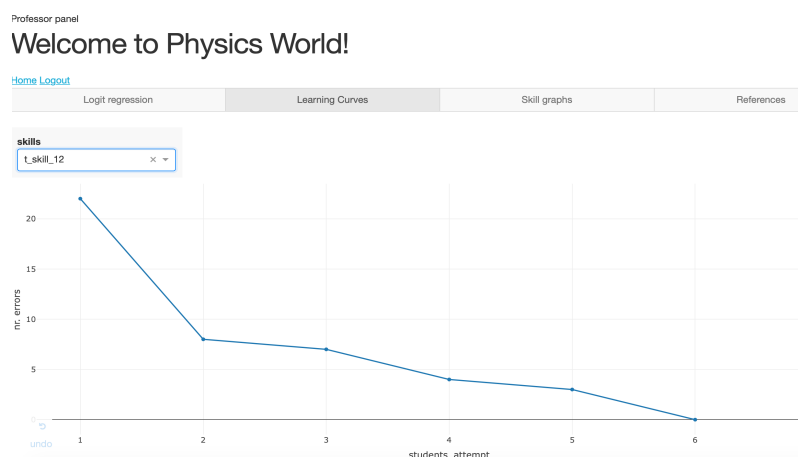


Figura 6.18: Pantalla del módulo del profesor que muestra la curva de aprendizaje para el skill_12. En esta habilidad, el alumno mejora el aprendizaje.

6.3. Valoración

Desde las experimentaciones llevadas a cabo con IDEE en entornos educativos reales, se ha comprobado su utilidad y adecuación a las metas educativas propuestas.

Gracias al proceso de evaluación llevado a cabo se ha puesto de manifiesto:

- la importancia de ayudar los profesores en favorecer la introducir en el contexto educativo de las nuevas tecnologías, para ayudarle en su uso diario.
- La motivación por parte del alumnado, que influye en la disposición hacia el estudio y la consecución de sus objetivos de aprendizaje.
- Una propuesta real y efectiva para la introducción de nuevos entornos en el contexto educativo.

- Un ejemplo concreto de cómo se pueden utilizar técnicas de inteligencias artificial en entornos educativos reales como apoyo a la mejora en el proceso de enseñanza aprendizaje.
- El dashboard del profesor es una herramienta fundamental para el profesor en la interpretación de los resultados de los modelos obtenidos al analizar las interacciones de los alumnos con el sistema.
- La introducción de actividad de laboratorio permite trabajar el aprendizaje colaborativo. Por su parte las actividades individuales permiten el desarrollo del conocimiento de acuerdo al ritmo de estudio particular de cada estudiante.
- La autocorrección de las actividades y el nivel de dificultad de los problemas según el conocimiento adquirido son herramientas útiles para el desarrollo individual del conocimiento en los alumnos.

Capítulo 7

Conclusiones y trabajos futuros

En este capítulo se comentarán los principales resultados que se han conseguido en la realización de este trabajo de tesis. Se presentan también las líneas de trabajo futuro considerando las potencialidades del sistema.

7.1. Conclusiones y principales aportaciones

La presente tesis se engloba dentro del campo de investigación que trata de utilizar las nuevas herramientas tecnológicas en el contexto educativo para favorecer el aprendizaje de los alumnos y al mismo tiempo ayudar al profesor a seguir sus aprendizajes.

El punto de partida de esta investigación ha sido la necesidad de incorporar nuevos recursos y nuevas metodologías en el contexto educativo. IDEE es una propuesta de un sistema interactivo de enseñanza y aprendizaje basado en web e integrado con una metodología constructivista del aprendizaje gracias a la implementación de actividades de laboratorio con instrumentos robóticos. Las evaluaciones en clase han sido útiles para poder verificar la funcionalidad del sistema bajo el aspecto metodológico y de implementación.

En la Figura 7.1 se muestran los pasos que se han seguido para la consecución de los objetivos de este proyecto de tesis.

Así, en el aspecto informático se ha trabajado en la implementación del entorno de aprendizaje:

- Se ha desarrollado el prototipo IDEE para educación secundaria que, mediante reglas, proporciona ayuda personalizada a los alumnos. Además, proporciona apoyo a los profesores para fomentar la introducción de las nuevas tecnologías en la didáctica.
- Se ha introducido en el sistema el componente robótico LEGO EV3 para que el alumno pueda desarrollar actividades de laboratorio. El alumno necesita

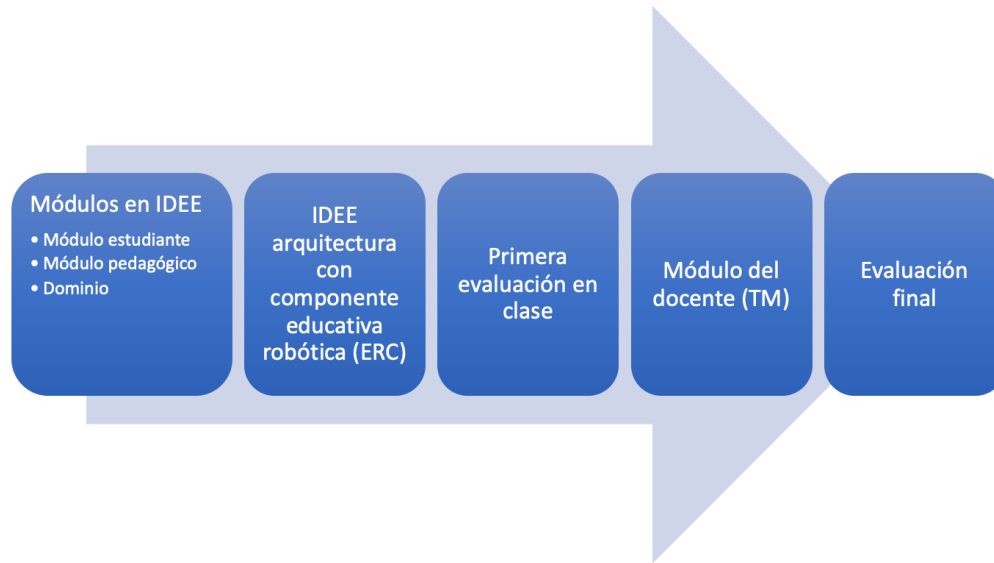


Figura 7.1: Pasos para la realización del proyecto de tesis

la programación para ejecutar los experimentos y como instrumento de cálculo.

- Se han creado bloques de comunicación entre el sistema y el robot que permiten activar desde el sistema, programas descargados en el robot. Esto permite adaptar el nivel de programación que se requiere a los alumnos y al mismo tiempo permite desarrollar los programas en un momento distinto de la explicación en clase, o en otras asignaturas.

Adicionalmente al desarrollo informático, se ha dado relevancia al modelo pedagógico y didáctico, considerando que el sistema creado se debe utilizar en un contexto real de clase en educación secundaria:

- Los modelos pedagógico y didáctico han sido desarrollados por la doctoranda teniendo en cuenta su experiencia como profesora de educación secundaria y considerando las actuales metodologías adoptadas en el contexto educativo.
- El modelo del estudiante permite que el sistema se adapte a los conocimientos del estudiante y a sus tiempos de aprendizaje.
- Se ha creado un modelo del profesor que le permite tener memoria de las actividades desarrolladas y del conocimiento adquirido por el estudiante. El modelo del profesor es abierto, esto le permite introducir o modificar actividades para adaptar los contenidos del sistema a sus contenidos didácticos.

Además, se ha considerado fundamental razonar sobre los modelos de minería de datos disponibles y se han incluido en el sistema los modelos que se han

considerado especialmente útiles para que el profesor pueda obtener información sobre el proceso de aprendizaje de los alumnos:

- El AFM ha sido utilizado para seleccionar los skills significativos en el aprendizaje de los alumnos y para identificar los alumnos que presentan dificultades en el aprendizaje.
- El CKT ha sido implementado para que el profesor pueda individualizar las actividades de recuperación únicamente sobre los skills necesarios.

Por último, se ha considerado fundamental ayudar al profesor en la interpretación de los resultados con la creación de una interfaz gráfica para el docente que permita el seguimiento del conocimiento de los alumnos.

Las experiencias llevadas a cabo en los institutos de educación secundaria nos permiten afirmar que el prototipo creado tiene las potencialidad para poder ser utilizado en contextos educativos reales.

7.2. Trabajos futuros

Gracias a la experiencia almacenada durante la puesta en marcha de IDEE, así como los resultados obtenidos en su evaluación, se han abierto nuevas expectativas a desarrollar que constituyen la base de las líneas futuras de trabajo en la que se seguirá investigando.

Los objetivos que concretizan estas líneas de trabajo son:

- Ampliar el dominio de aplicación de IDEE a otros temas dentro del nivel y la asignatura en la que se desarrolla, convirtiéndose así en una propuesta global para desarrollar los diferentes bloques temáticos de la programación curricular.
- Validar la propuesta implementada en IDEE en otros dominios, trabajando en la adaptación de la experimentación robótica.
- Estudiar la posible ampliación de la herramienta de análisis de datos para mejorar la información acerca del recorrido de aprendizaje del alumno.
- Ampliar el número de robots que puedan ser utilizados en el sistema.
- Trabajar en la presentación de los resultados de los modelos de minería de datos, para facilitar al profesor la lectura y la comprensión de los resultados.
- Facilitar la inclusión de las actividades por parte del profesor. Se considera fundamental que el profesor pueda encontrar en IDEE un instrumento donde pueda incluir los contenidos didácticos de la programación curricular.

7.3. Publicaciones

Por último, respecto a la difusión y publicación de los resultados presentados en esta memoria, se mencionan las publicaciones y las actividades derivadas.

Publicaciones en Revistas Internacionales

- Orlando, S., Gaudioso, E., De La Paz, F. (March, 2020). Supporting teachers to monitor student's learning progress in an educational environment with robotics activities. *IEEE Access*, 8, 48620-48631.

Publicaciones en actas de congresos

- Orlando, S., Gaudioso, E., De la Paz, F. (2018, April). IDEE: A visual programming environment to teach physics through robotics in secondary schools. In *International Conference on Robotics and Education RiE 2017* (pp. 241-246). Springer, Cham.
- Orlando, S., De la Paz López, F., Gaudioso, E. (2019, June). Design and implementation of a robotics learning environment to teach physics in secondary schools. In *International Work-Conference on the Interplay Between Natural and Artificial Computation* (pp. 69-76). Springer, Cham.

Actividades de divulgación científica

- Orlando, S., Gaudioso, E., De la Paz, F. (2018, Octubre). Design and implementation of a learning environment for robotics integration in a physics laboratory at secondary schools (Poster). In *International Conference on Intelligent Robots and Systems (IROS)*.
- XVIII Semana de la Ciencia (2018 Noviembre). Actividad: Galileo y el plano inclinado, una práctica de Física con LEGO mindstorms EV3.
- Aula Semana de la educación 2019. Actividades de robótica con el uso del sistema IDEE.
- Orlando, S., Gaudioso, E., De la Paz, F. (2019, Diciembre). Design and implementation of a learning environment for robotics integration in a physics laboratory at secondary schools (Poster). In *RoboCity2030 Robótica e inteligencia artificial: retos y nuevas oportunidades*.

Bibliografía

- [1] E. Afari y M. S. Khine. Robotics as an educational tool: Impact of lego mindstorms. *International Journal of Information and Education Technology*, 7(6):437–442, 2017.
- [2] Ishrat Ahmed, Nichola Lubold, y Erin Walker. Robin: using a programmable robot to provide feedback and encouragement on programming tasks. En *International Conference on Artificial Intelligence in Education*, págs. 9–13. Springer, 2018.
- [3] R. Al-Shabandar, A. J. Hussain, P. Liatsis, y R. Keight. Analyzing learners behavior in moocs: An examination of performance and motivation using a data-driven approach. *IEEE Access*, 6:73669–73685, 2018.
- [4] D. Alimisis. Educational robotics: open questions and new challenges. *Themes in Science & Technology Education*, 6(1):63–71, 2013.
- [5] Ethem Alpaydin. Introduction to machine learning. isbn: 978-0-262-028189. 2014.
- [6] M. Armoni, S. O. Meerbaum, y A. M. Ben. From scratch to 'real'programming. *ACM Trans. Comput. Educ.*, 14(4), 2015.
- [7] A. Badeleh. The effects of robotics training on students'creativity and learning in physics. *Education and Information Technologies*, 2019. doi:10.1007/s10639-019-09972-6. URL <https://doi.org/10.1007/s10639-019-09972-6>.
- [8] F. Barreto y V. Benitti. Exploring the educational potential of robotics in schools: A systematic review. *Computers & Education*, 58(3):978–988, 2012.
- [9] N. A. Bascou y K. Menekse, M. Robotics in k-12 formal and informal learning environments: A review of literature. En *ASEE Annual Conference*, pág. 26119. Springer Berlin Heidelberg, Louisiana, 2016.

- [10] Anis Bey, Mar Pérez-Sanagustín, y Julien Broisin. Unsupervised automatic detection of learners' programming behavior. En *European Conference on Technology Enhanced Learning*, págs. 69–82. Springer, 2019.
- [11] A. Bogarin, R. Cerezo, y C. Romero. A survey on educational process mining. *WIREs Data Mining Knowledge Discovery*, 8:e1230, 2018.
- [12] H. Cen, K. Koedinger, y B. Junker. Learning factors analysis. a general method for cognitive model evaluation and improvement. En M. Ikeda, Kevin D. Ashley, y T.W. Chan, eds., *Intelligent Tutoring Systems*, págs. 164–175. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [13] Albert T Corbett y John R Anderson. Knowledge tracing: Modeling the acquisition of procedural knowledge. *User modeling and user-adapted interaction*, 4(4):253–278, 1994.
- [14] N. Diana, M. Eagle, J. Stamper, S. Grover, M. Bienkowski, y S. Basu. An instructor dashboard for real-time analytics in interactive programming assignments. En *Proceedings of the Seventh International Learning Analytics & Knowledge Conference, LAK '17*, págs. 272–279. ACM, New York, NY, USA, 2017. ISBN 978-1-4503-4870-6.
- [15] G. Durand, Goutte C., N. Belacel, Y. Bouslimani, y S. Léger. Review, computation and application of the additive factor model (AFM). Inf. Téc. 23002483, National Research Council Canada, 2017.
- [16] A. Dutt, M. A. Ismail, y T. Herawan. A systematic review on educational data mining. *IEEE Access*, 5:15991–16005, 2017.
- [17] E. Emerson, A. Smith, C. Smith, F.J. Rodríguez, W. Min, E.N. Wiebe, B.W. Mott, K.E. Boyer, y J.C. Lester. Predicting early and often: Predictive student modeling for block-based programming environments. En *12th International Conference on Educational Data Mining*. Montreal, Canada, 2019.
- [18] Paola Ferrarelli, Wilson Villa, Margherita Attolini, Donatella Cesareni, Federica Micale, Nadia Sansone, Luis Claudio Pantaleone, y Luca Iocchi. Improving students' concepts about newtonian mechanics using mobile robots. En Wilfried Lopuschitz, Munir Merdan, Gottfried Koppensteiner, Richard Balogh, y David Obdržálek, eds., *Robotics in Education*, págs. 113–124. Springer International Publishing, Cham, 2019. ISBN 978-3-319-97085-1.

- [19] Daniel Amo Filvà, Marc Alier Forment, Francisco José García-Peñalvo, David Fonseca Escudero, y María José Casañ. Clickstream for learning analytics to assess students' behavior with scratch. *Future Generation Computer Systems*, 93:673–686, 2019.
- [20] Harvey B. Garcia D. y Barnes T. The beauty and joy of computing. En *ACM inroads New York, NY, USA*, págs. 71–79. 2015.
- [21] Howard Gardner. *Frames of mind: The theory of multiple intelligences*. Hachette Uk, 2011.
- [22] Ilaria Gaudiello y Elisabetta Zibetti. *Learning robotics, with robotics, by robotics: Educational robotics*. John Wiley & Sons, 2016.
- [23] E. Gaudioso, M. Montero, L. Talavera, y F. Hernandez-del Olmo. Supporting teachers in collaborative student modeling: A framework and an implementation. *Expert Systems with Applications*, 36(2, Part 1):2260–2265, 2009.
- [24] Luis-Eduardo Imbernón Cuadrado, Ángeles Manjarrés Riesco, y Félix De La Paz López. Artie: An integrated environment for the development of affective robot tutors. *Frontiers in computational neuroscience*, 10:77, 2016.
- [25] Andri Ioannou, Chrysanthos Socratous, y Elena Nikolaedou. Expanding the curricular space with educational robotics: A creative course on road safety. En Viktoria Pammer-Schindler, Mar Pérez-Sanagustín, Hendrik Drachslar, Raymond Elferink, y Maren Scheffel, eds., *Lifelong Technology-Enhanced Learning*, págs. 537–547. Springer International Publishing, Cham, 2018. ISBN 978-3-319-98572-5.
- [26] I. Jormanainen y E. Sutinen. Role blending in a learning environment supports facilitation in a robotics class. *Educational Technology & Society*, 17(1):294–306, 2014.
- [27] Yasmin B Kafai y Quinn Burke. Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, 50(4):313–334, 2015.
- [28] Markus Ketterl, Beate Jost, Thorsten Leimbach, y Reinhard Budde. Tema 2: Open roberta-a web based approach to visually program real educational robots. *Tidsskriftet Læring og Medier (LOM)*, 8(14), 2015.
- [29] A. Khanlari. Effects of educational robots on learning stem and on students' attitude toward stem. En *2013 IEEE 5th Conference on Engineering Education (ICEED)*, págs. 62–66. 2013.

- [30] K. Koedinger, R. Baker, K. Cunningham, A. Skogsholm, B. Leber, y J. Stamper. A data repository for the edm community: The pslc datashop. En C. Romero, S. Ventura, M. Pechenizkiy, y R. Baker, eds., *Handbook of Educational Data Mining*, págs. 43–53. CRC Press, 2010.
- [31] K. Koedinger, S. Mello, E. McLaughlin, Z. Pardos, y C. Rose. Data mining and education. *Wiley Interdisciplinary Reviews:Cognitive Science*, 6(4):333–353, 2015.
- [32] K. Koedinger, P Pavlik, J. Stamper, T. Nixon, y S. Ritter. Avoiding problem selection thrashing with conjunctive knowledge tracing. En *Educational data mining 2011*. 2010.
- [33] Chia-Yu Liu, Chao-Jung Wu, Wing-Kwong Wong, Yunn-Wen Lien, y Tsung-Kai Chao. Scientific modeling with mobile devices in high school physics labs. *Computers & Education*, 105:44–56, 2017.
- [34] Y. Mao, R. Zhi, F. Khoshnevisan, T. Price, T. Barnes, y M. Chi. One minute is enough: Early prediction of student success and event-level difficulty during novice programming tasks. En *12th International Conference on Educational Data Mining*. Montreal, Canada, 2019.
- [35] B. Martin, A. Mitrovic, K.R. Koedinger, y S. Mathan. Evaluating and improving adaptive educational systems with learning curves. *User Modeling and User-adapted interaction*, 21(3):249–283, 2011.
- [36] Riccardo Mazza. *Introduction to information visualization*. Springer Science & Business Media, 2009.
- [37] RD Merlo-Espino, M Villareal-Rodríguez, A Morita-Aleander, J Rodríguez-Reséndiz, GI Pérez-Soto, y KA Camarillo-Gómez. Educational robotics and its impact in the development of critical thinking in higher education students. En *2018 XX Congreso Mexicano de Robótica (COMRob)*, págs. 1–4. IEEE, 2018.
- [38] J. E. Michaelis y B. Mutlu. Supporting interest in science learning with a social robot. En *Proceedings of the 18th ACM International Conference on Interaction Design and Children, IDC '19*, págs. 71–82. ACM, New York, NY, USA, 2019. ISBN 978-1-4503-6690-8. doi:10.1145/3311927.3323154.
- [39] Ariana Milašinčić, Bruna Anđelić, Liljana Pushkar, y Ana Sović Kržić. Using robots as an educational tool in native language lesson. En Munir Merdan, Wilfried Lopuschitz, Gottfried Koppensteiner, Richard Balogh, y David

- Obdržálek, eds., *Robotics in Education*, págs. 296–301. Springer International Publishing, Cham, 2020. ISBN 978-3-030-26945-6.
- [40] Omar Mubin, Catherine J Stevens, Suleman Shahid, Abdullah Al Mahmud, y Jian-Jie Dong. A review of the applicability of robots in education. *Journal of Technology in Education and Learning*, 1(209-0015):13, 2013.
- [41] S. Orlando, F. de la Paz, y E. Gaudio. Design and implementation of a robotics learning environment to teach physics in secondary schools. En *From Bioinspired Systems and Biomedical Applications to Machine Learning - 8th International Work-Conference on the Interplay Between Natural and Artificial Computation*, págs. 69–76. 2019.
- [42] S. Orlando, E. Gaudio, y F. de la Paz. IDEE: A visual programming environment to teach physics through robotics in secondary schools. En *Robotics in Education*, págs. 241–246. Springer International Publishing, 2018. ISBN 978-3-319-97085-1.
- [43] S. Orlando, E. Gaudio, y F. De La Paz. Supporting teachers to monitor student’s learning progress in an educational environment with robotics activities. *IEEE Access*, 8:48620–48631, 2020.
- [44] Sofia Papavlasopoulou, Michail N Giannakos, y Letizia Jaccheri. Empirical studies on the maker movement, a promising approach to learning: A literature review. *Entertainment Computing*, 18:57–78, 2017.
- [45] S. Papert. *Mindstorms. Children, Computers and Powerful Ideas*. New York: Basic books, 1980.
- [46] Seymour Papert. *The connected family: Bridging the digital generation gap*, tomo 1. Taylor Trade Publishing, 1996.
- [47] Martha C Polson y J Jeffrey Richardson. *Foundations of intelligent tutoring systems*. Psychology Press, 2013.
- [48] M Ponticorvo, F Rubinacci, D Marocco, F Truglio, y O Miglino. Educational robotics to foster and assess social relations in students’ groups. *Front. Robot. AI* 7: 78. doi: 10.3389/frobt, 2020.
- [49] M. Prensky. Digital natives, digital immigrants. *On the horizon*, 9(5), 2001.
- [50] Dong Y., Price T. W. y Lipovac D. iSnap: Towards intelligent tutoring in novice programming environments. En *SIGCSE ’17 Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*, págs. 483–488. Springer International Publishing, 2017. ISBN 978-1-4503-4698-6.

- [51] José-Manuel Sáez-López, Marcos Román-González, y Esteban Vázquez-Cano. Visual programming languages integrated across the curriculum in elementary school: A two year case study using “scratch” in five schools. *Computers & Education*, 97:129–141, 2016.
- [52] A. Savard y V. Freiman. Investigating complexity to assess student learning from a robotics-based task. *Digital Experiences in Mathematics Education*, 2(2):93–114, 2016.
- [53] David Scaradozzi, Lorenzo Cesaretti, Laura Screpanti, y Eleni Mangina. Identification of the students learning process during education robotics activities. *Frontiers in Robotics and AI*, 7:21, 2020.
- [54] S. Seabold y J. Perktold. Statsmodels: Econometric and statistical modeling with python. En S. van der Walt y J. Millman, eds., *Proceedings of the 9th Python in Science Conference*, págs. 57–61. Austin, Texas, 2010.
- [55] Del Siegle. Developing student programming and problem-solving skills with visual basic. *Gifted Child Today*, 32(4):24–29, 2009.
- [56] J. Stamper y K. R. Koedinger. Human-machine student model discovery and improvement using data. En *Proceedings of the 15th Artificial Intelligence in Education*, págs. 353–360. Berlin:Springer-Verlag, 2011.
- [57] Toivonen T., I. Jormanainen, y Tukiainen M. An open robotics environment motivates students to learn the key concepts of artificial neural networks and reinforcement learning. En *Advances in Intelligent Systems and Computing*. International Conference on Robotics and Education RiE 2017, 2017.
- [58] Khushboo Thaker, Paulo Carvalho, y Kenneth Koedinger. Comprehension factor analysis: Modeling student’s reading behaviour: Accounting for reading practice in predicting students’ learning in moocs. En *Proceedings of the 9th International Conference on Learning Analytics & Knowledge*, págs. 111–115. 2019.
- [59] Julio Vega y José M Cañas. Pybokids: an innovative python-based educational framework using real and simulated arduino robots. *Electronics*, 8(8):899, 2019.
- [60] L. Xia y B. Zhong. A systematic review on teaching and learning robotics content knowledge in k-12. *Computers & Education*, 127:267–282, 2018.

-
- [61] Claus Zinn y Oliver Scheuer. Getting to know your student in distance learning contexts. En *European Conference on Technology Enhanced Learning*, págs. 437–451. Springer, 2006.
- [62] O. Zughoul, F. Momani, O. H. Almasri, A. A. Zaidan, B. B. Zaidan, M. A. Alsalem, O. S. Albahri, A. S. Albahri, y M. Hashim. Comprehensive insights into the criteria of student performance in various educational domains. *IEEE Access*, 6:73245–73264, 2018.

Apéndice A

Códigos destacables de la aplicación

Actualmente el código del sistema se encuentra en un repositorio GitHub:
(poner el link GitHub)

A.1. Uso de Django en la creación de la base de datos

En la sección 4.3.1 se ha descrito la organización de la base de datos de IDEE.

Se detallan los códigos Python desarrollados en Django en nuestro proyecto *web_learning* en la carpeta *modelos*, que caracterizan la organización de la base de datos para la determinación del modelo del estudiante.

Para que el sistema tenga memoria de las interacciones del alumno con el sistema se ha creado la clase *Task* en el modelo *task.py*:

```
1 class Task(models.Model):
2     start_time = models.DateTimeField(auto_now_add=True)
3     end_time = models.DateTimeField(null=True, blank=True,
4     default = None)
5     element = models.ForeignKey(Element, on_delete=models.
6     CASCADE)
7     user = models.ForeignKey(User, on_delete=models.CASCADE)
```

Listing A.1: Clase Task en el modelo task.py

En la clase *Data* del modelo *data.py* el sistema guarda la información sobre las interacciones del alumno con el sistema; memorizando la actividad desarrollada, la respuesta del alumno, la hora y la respuesta correcta.

```
1 class Data(models.Model):
```

```

2     answer = models.CharField(max_length=500, blank=False, null
3     =True)
4     time = models.DateTimeField(auto_now_add=True)
5     right_answer = models.BooleanField(blank=True)
6     task = models.ForeignKey(Task, on_delete=models.CASCADE)
7     user = models.ForeignKey(User, on_delete=models.CASCADE)
8     assessment = models.ForeignKey(Assessment, on_delete=models
    .CASCADE)

```

Listing A.2: Clase Data en el modelo data.py

En la clase *Profile* del modelo *profile.py*, el sistema memoriza el conocimiento acumulado *accumulate_score* por el alumno según las actividades ejecutadas.

```

1     class Profile(models.Model):
2         user = models.ForeignKey(User, on_delete=models.CASCADE)
3         skill = models.ForeignKey(Skill, on_delete=models.CASCADE)
4         )
5         element = models.ForeignKey(Element, on_delete=models.
6         CASCADE)
7         accumulate_score = models.FloatField(default=0)
8         date = models.DateTimeField(auto_now_add=True)

```

Listing A.3: Clase Profile en el modelo profile.py

A.2. Creación de actividades

En la sección 4.6, se ha descrito el diseño de las actividades implementadas en IDEE. A continuación, se muestra el código Python desarrollado en Django correspondiente a la creación de una actividad de laboratorio del plano inclinado.

En nuestro proyecto `web_learning`, las actividades están desarrolladas en la carpeta `->management ->command`

```

1
2     # MovimientoPlanoInclinado-lev1
3     toolbox = xml.read('tb-MovimientoPlanoInclinado-lev1.xml')
4     startblocks = xml.read('ws-MovimientoPlanoInclinado-lev1.xml')
5     htmltext = html.read('MovimientoPlanoInclinado.html')
6     obj, created = VersionProblem.objects.get_or_create(
7         problem=Problem.objects.get(name='Movimiento Plano Inclinado')
8         ),
9         level = 1)
10    obj.text = htmltext
11    obj.toolbox = toolbox
12    obj.startblocks = startblocks

```

```

12 obj.requireRobots = True
13 obj.format_answer = '{"type": "table",' \
14     'col_names": ["Inc'ognita", "Tiempo de ca'ida"],' \
15     'rows": [["Espacio1", ""], ["Espacio2", ""], ["Espacio3", ""], ["Espacio4", ""]]}' \}'
16 obj.save()
17 if created:
18     # create relation_MovimientoPlanoInclinado-lev1
19     new_relation = Relation_VersionProblem(skill=skill13,
20                                           versionproblem=obj,
21                                           score=10)
22     new_relation.save()
23     new_relation = Relation_VersionProblem(skill=skill3,
24                                           versionproblem=obj,
25                                           score=8)
26     new_relation.save()
27     new_relation = Relation_VersionProblem(skill=skill19,
28                                           versionproblem=obj,
29                                           score=5)
30     new_relation.save()

```

Listing A.4: Código python para la creación del laboratorio

- El problema está relacionado con tres skills, cuya relevancia en el problema se indica con el correspondiente score.
- El toolbox incluye los bloques necesarios para el desarrollo de la actividad en código xml:

```

1 <xml xmlns="http://www.w3.org/1999/xhtml" id="toolbox"
  style="display: none;">
2   <category id="%{BKY_CATEV3}" name="Ev3" colour="%{
  BKY_EV3_HUE}">
3     <block type="ev3_read_message">
4       <value name="name">
5         <block type="text">
6           <field name="TEXT">TimePlane</field>
7         </block>
8       </value>
9     </block>
10    <block type="ev3_send_message_angle">
11      <value name="name">
12        <block type="text">
13          <field name="TEXT">angulo</field>
14        </block>
15      </value>
16    </block>
17    <block type="ev3_send_message_inclinedPlane">

```

```

18         <value name="name">
19             <block type="text">
20                 <field name="TEXT">PlanoInclinado</
field>
21             </block>
22         </value>
23         <value name="payload">
24             <block type="text">
25                 <field name="TEXT">Tiempo_Caida</field>
26             </block>
27         </value>
28     </block>
29 </category>
30 <category name="Math" colour="#5C68A6">
31     <block type="math_number">
32         <field name="NUM">0</field>
33     </block>
34 </category>
35     <category id="%{BKY_CATDELAYS}" name="answer" colour="
%{BKY_DELAYS_HUE}">
36         <block type="answer"></block>
37     </category>
38 </xml>

```

Listing A.5: Código xml para la creación del toolbox de la actividad de laboratorio del plano inclinado.

- El *startblocks* incluye las ayudas proporcionadas al estudiante por el sistema según su conocimiento en los skills sobre que trabaja la actividad. Un nivel1 no presupone ninguna ayuda por parte del sistema.
- El *htmltext* incluye el texto y la gráfica del laboratorio.

```

1     <b>GALILEO Y SUS IDEAS</b>
2     <br>
3     <br>Reflexionamos sobre la relación entre el espacio
recorrido por la pelota y el tiempo de caída
reproduciendo el mismo experimento que hizo Galileo
Galilei.
4     <br>Programa el robot para que tenga una inclinación de
15 grados. Calcula el tiempo de caída de la pelota con
los espacios indicados.
5     <br>Reflexiona sobre lo que pasaría repitiendo el
experimento con una inclinación de 10 grados.
6     <link rel="stylesheet" href="/static/db/css/chart.css">
7     <div id="myChartContainer"><canvas id="myChart"><
/canvas></div>

```



```

8   <button id="updateChartBtn">Actualiza gráfico</button>
9   <script src="/static/js/lib/Chart.bundle.min.js"></script>
10  <script src="/static/db/js/chart.js"></script>

```

Listing A.6: Código xml para la creación del htmltext

- Se muestra el código en JavaScript para la creación del gráfico chart.js de la actividad de laboratorio del plano inclinado.

```

1   var ctx = document.getElementById("myChart").getContext(
2   '2d');
3   var myChart_defaultData = [{x:0, y: 10.1},{x:0, y:
4   21.4},{x:0, y: 32.7},{x:0, y: 44}];
5   var myChart_trueData = [{x:0.337, y: 10.1},{x:0.487, y:
6   21.4},{x:0.599, y: 32.7},{x:0.701, y: 44}];
7   var myChart_datasets;
8
9   myChart_datasets = [{ data: myChart_defaultData }];
10
11
12  if (already_answered) {
13      myChart_datasets.push({ data: myChart_trueData,
14      borderColor: 'green', fill: false })
15  }
16
17  var myChart = new Chart(ctx, {
18      type: 'line',
19      data: {
20          datasets: myChart_datasets
21      },
22      options: {
23          legend: {
24              display: false
25          },
26          scales: {
27              yAxes: [{
28                  ticks: {
29                      beginAtZero: true
30                  }
31              }],
32              xAxes: [{
33                  type: 'linear',
34                  ticks: {
35                      min: 0
36                  }
37              }
38          ]
39      }
40  }

```

```

35     });
36
37     function graphMe(){
38         for (var i = 0; i < myChart.data.datasets[0].data.
length; i++) {
39             var cell = "'R"+i+"C1'";
40             var data = parseFloat($("#input[name="+cell+"]").
val())
41             if (data != undefined){
42                 myChart.data.datasets[0].data[i]['x'] = data
;
43             }
44         }
45         myChart.update();
46     }
47
48     $("#updateChartBtn").click(graphMe);
49     $.ready(graphMe)
50

```

Listing A.7: Código JavaScript para la creación del gráfico chart.js.

A.3. Creación de los bloques de comunicación entre el robot y el sistema

En la sección 4.4 se ha descrito el uso de Blockly en IDEE, que ha permitido la introducción de los bloques para la programación del robot. Se ha elegido mostrar, como ejemplo de uso de Blockly in IDEE, los códigos de comunicación entre el sistema y el robot, considerando la implementación de estos bloques fundamentales a la hora de utilizar una herramienta como IDEE en clase. Los códigos son en JavaScript.

Código para la creación de la forma y del generador del bloque de comunicación desde el robot EV3 al sistema.

El robot en las actividades de laboratorio desarrolla el programa en el ladrillo y posteriormente envía un mensaje con un resultado al sistema para que el alumno pueda visualizar el resultado en la pantalla del robot y también en la interfaz del sistema IDEE.

```

1 Blockly.Blocks['ev3_read_message'] = {
2   init: function () {
3     this.appendValueInput("name")
4       .setCheck([TYPES.STRING, TYPES.VARIABLE])

```

```

5     .appendField("message name");
6     this.setColour(constants.colors.input);
7     this.setOutput(true, 'Number');
8     this.onchange = onchange;
9 }
10 };

```

Listing A.8: Código JavaScript para la creación de la forma del bloque de comunicación desde el robot EV3 al sistema.

```

1   Blockly.JavaScript['ev3_read_message'] = function (block) {
2   var variables = {
3     name: generators.valueToCode(block, 'name', generators.
ORDER_ATOMIC),
4     ev3: quoted(Blockly.Constants.ev3.Names.EXT),
5     cmd: quoted('readMailboxMessage')
6   };
7
8   var code = Mustache.render("flog(runCommandAsync({{ev3}}),{{
cmd}}),[{{name}}]),'message: ')", variables);
9   return [code, Blockly.JavaScript.ORDER_FUNCTION_CALL];
10 };

```

Listing A.9: Código JavaScript para la creación de la forma del bloque de comunicación desde el robot EV3 al sistema.

Código para la creación de la forma y del generador del bloque de comunicación desde el sistema al robot EV3. El código reportado se refiere a un mensaje de tipo numérico.

El sistema puede enviar al robot mensajes numéricos, palabras y parámetros. El código que se muestra es para la creación de un mensaje numérico.

```

1   Blockly.Blocks['ev3_send_message_num'] = {
2   init: function () {
3     this.appendValueInput("name")
4       .setCheck([TYPES.STRING, TYPES.VARIABLE])
5       .appendField("NUMERIC message name");
6     this.appendValueInput("payload")
7       .setCheck([TYPES.NUMBER, TYPES.VARIABLE])
8       .appendField("payload");
9     this.setColour(constants.colors.output);
10    this.setPreviousStatement(true);
11    this.setNextStatement(true);
12    this.onchange = onchange;
13  }

```

```
14 };
```

Listing A.10: Código JavaScript para la creación de la forma del bloque de comunicación desde el sistema al robot EV3.

```
1   Blockly.JavaScript['ev3_send_message_num'] = function (block)
2   {
3       var variables = {
4           name: generators.valueToCode(block, 'name',
5           generators.ORDER_ATOMIC),
6           payload: generators.valueToCode(block, 'payload',
7           generators.ORDER_ATOMIC),
8           ev3: quoted(Blockly.Constants.ev3.Names.EXT),
9           cmd: quoted('sendMailboxMessage_num')
10      };
11
12      return Mustache.render("runCommand({{ev3}},{{cmd}},[{{
13      name}}, {{payload}}]);\n", variables);
14  };
15
```

Listing A.11: Código JavaScript para la creación del generador del bloque de comunicación desde el sistema al robot EV3.

A.4. Determinación del conocimiento del alumno

Se muestra el código Python que permite al sistema determinar el conocimiento del alumno *accumulate score*.

```
1  def update_skill_table_dict(table_skills, versionproblem_id,
2  correct):
3      skills = Relation_VersionProblem.objects.filter(
4      versionproblem=versionproblem_id)
5      for skill in skills:
6          key = str(skill.skill_id)
7          if key in table_skills:
8              if correct:
9                  table_skills[key]['right'] += skill.score
10             else:
11                 table_skills[key]['wrong'] += skill.score
12             elif correct:
13                 table_skills[key] = {'right': skill.score, 'wrong':
14                 0, 'score': skill.score}
15             else:
16                 table_skills[key] = {'right': 0, 'wrong': skill.
17                 score, 'score': skill.score}
18
19  def update_user_skills(table_skills, data):
```

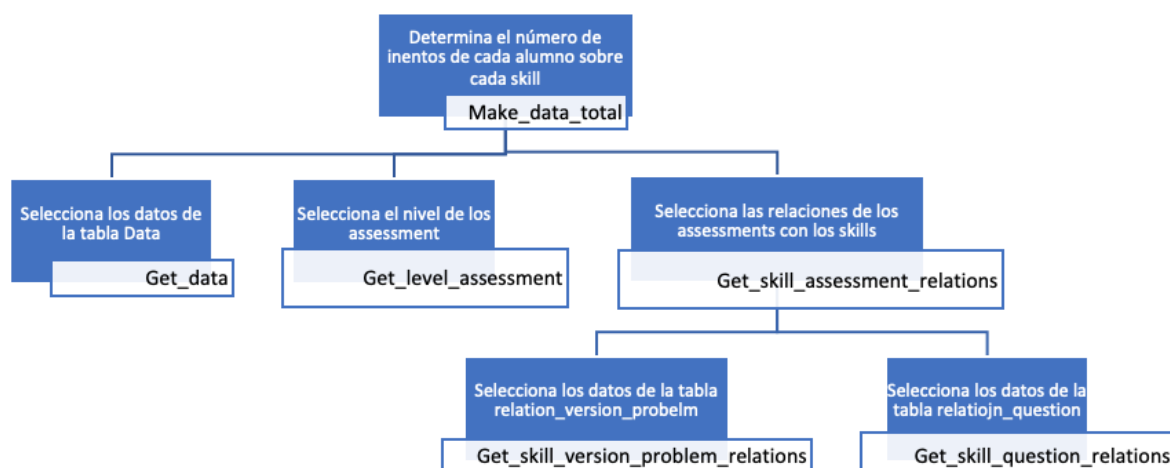


Figura A.1: Diagrama de las funciones utilizadas para convertir los datos cuantitativos de la base de datos útiles para la aplicación de los métodos de minería de datos.

```

16     for key in table_skills:
17         skill = Skill.objects.get(pk=key)
18         score = table_skills[key]['right']/(table_skills[key]['right']+table_skills[key]['wrong']) * table_skills[key]['score']
19
19         if score >0:
20             new_profile = Profile(user=data['user'], element=data['element'], date=data['date'],
21                                   skill=skill, score=score)
22             new_profile.save()
23
  
```

Listing A.12: Código para determinar el *accumulate score* de cada estudiante en cada actividad.

A.5. Implementación de los modelos de minería de datos

Determinación de la Q-matriz para la aplicación de los modelos

Los datos cuantitativos de la base de datos del sistema se deben convertir en datos útiles para los modelos de minería de datos utilizados.

En la Figura A.1 se resumen los pasos necesarios para la determinación de la Q-matriz, necesaria para la implementación de los modelos AFM y CKT.

Códigos para obtener la Q-matriz

Se muestran los códigos Python necesarios para que los datos de la base de datos se conviertan en datos para la determinación de la Q-matriz.

```

1
2     def load_data(local=True):
3
4         #level
5         ldf = get_level_assessment(local)
6
7         # Assessment
8         adf = get_skill_assessment_relations(local)
9
10        # multiply skills by score
11
12        if False:
13            skill_names = [c for c in adf.columns if c.startswith('
skill')]
14            for col in skill_names:
15                adf[col] = adf[col] * adf['score']
16
17        adf = adf.drop('score', axis=1)
18        adf = adf.groupby(by=['assessment_id', 'type']).sum().
reset_index()
19
20        # Data
21        ddf = get_data(local)
22        ddf['num_attempt'] = ddf.groupby(['user_id', 'assessment_id'
]).cumcount() + 1
23        ddf = ddf.merge(adf, on='assessment_id', how="left")
24            #.drop('task_id', axis=1)
25        ddf = ddf.merge(ldf, on='assessment_id', how="left")
26        return ddf

```

Listing A.13: Código para determinar la Q-matriz.

```

1
2     def get_data(local=True, filename='d_It.csv'):
3         edir = 'ephysic/controls/'
4
5         if local:
6             # read data from file
7             ddf = pd.read_csv(edir + filename)
8         else:
9             # read Relation_VersionProblem
10            ddf = pd.DataFrame(Data.objects.all().values())
11

```

```

12 ddf = ddf.drop(['id', 'answer'], axis=1)
13 return ddf

```

Listing A.14: Código para seleccionar los datos de la tabla Data.

```

1
2 def get_level_assessment(local=True, filename='l.csv'):
3     edir = 'ephysic/controls/'
4
5     if local:
6         # read data from file
7         ldf = pd.read_csv(edir + filename)
8     else:
9         # read Relation_VersionProblem
10        ldf = pd.DataFrame(Assessment.objects.all().values())
11
12    return ldf[['id', 'level']].rename(columns={'id': '
assessment_id'})

```

Listing A.15: Código para seleccionar los datos de la tabla Assessment de la base de datos para tener información del nivel de la actividad.

```

1
2 def get_skill_assessment_relations(local=True):
3
4     # VersionProblem
5     vdf = get_skill_versionproblem_relations(local)
6
7     # Question
8     qdf = get_skill_question_relations(local)
9
10    adf = vdf.append(qdf)
11    adf = pd.get_dummies(adf, columns=['skill'])
12
13    return adf
14
15
16 def get_skill_versionproblem_relations(local=True, filename='
v.csv'):
17    edir = 'ephysic/controls/'
18
19    if local:
20        # read data from file
21        vdf = pd.read_csv(edir + filename)
22    else:
23        # read Relation_VersionProblem
24        vdf = pd.DataFrame(Relation_VersionProblem.objects.all().
values())

```

```
25
26     vdf = vdf.drop(['id', ], axis=1)
27     vdf.columns = ['score', 'skill', 'assessment_id']
28     vdf.insert(0, 'type', 0)
29     return vdf
30
31
32     def get_skill_question_relations(local=True, filename='q.csv'
33 ):
34     edir = 'ephysic/controls/'
35
36     if local:
37         # read data from file
38         qdf = pd.read_csv(edir + filename)
39     else:
40         # read Relation_VersionProblem
41         qdf = pd.DataFrame(Relation_Question.objects.all().values
42 ())
43
44     qdf = qdf.drop(['id', ], axis=1)
45     qdf.columns = ['score', 'skill', 'assessment_id']
46     qdf.insert(0, 'type', 1)
47     return qdf
```

Listing A.16: Código para seleccionar los datos de las relaciones de los assessment con los skills.

Código para la determinación del número de intentos de cada alumno

```
1     def make_data_total(local=True, dummy_userid =True):
2
3     data_df = load_data(local)
4
5     for c_skill in data_df.columns:
6
7         if c_skill.startswith("skill"):
8             if_skill = [1*(x>0) for x in data_df[c_skill]]
9             t_skill = if_skill * (data_df.groupby(['user_id',
10 c_skill]).cumcount() + 1)
11
12             if max(t_skill) > 1:
13                 data_df['t_' + c_skill] = t_skill
14
15     data_df = data_df.drop(['num_attempt'], axis=1)
16
17     if dummy_userid:
```



```

18     data_df = pd.get_dummies(data_df, columns=['user_id'])
19
20     return data_df

```

Listing A.17: Código para la determinación del número de intentos de cada alumno.

Implementación del método AFM

Implementación de la función para la aplicación de la regresión logística del método AFM.

```

1 def compute_logit_regression(data_df, statistic_model='Logit'):
2
3     y = 'right_answer'
4     x = np.array([i for i in data_df.columns if i not in y])
5     model = y + " ~ " + " + ".join(data_df[x].columns)
6
7
8     if statistic_model=='glm':
9         logit_reg = smf.glm(formula=model, data=data_df, family=
sm.families.Binomial())
10         #logit_reg = sm.GLM(data_df['right_answer'], data_df[x],
family=sm.families.Binomial())
11
12     elif statistic_model == 'OLS':
13         # logit_reg = smf.ols(formula=model, data=data_df) # not
working...
14         logit_reg = smf.OLS(data_df['right_answer'], data_df[x])
15
16     else:
17         logit_reg = (sm.Logit(data_df['right_answer'], data_df[x
]))
18
19     res = logit_reg.fit()
20     print('AIC', res.aic)
21     print('BIC', res.bic)
22     return res.summary()

```

Listing A.18: Función que permite la implementación del modelo AFM

Implementación del método CKT

Códigos para la implementación de las funciones para la aplicación del método CKT en el modelo del estudiante.

```

1
2 def conjunctiveKT_score(obs, df, arrayK):
3     df = df / 10

```

```

4     n_states=len(arrayK)
5     n_obs=len(obs)
6     delta=np.copy(arrayK)
7     for t in range(n_obs):
8         o = obs[t]
9         inc=(df.iloc[t].values !=0)
10        arrayS, arrayG, arrayT = parameters_score(n_states, df,t)
11        arrayK = np.array([prob_cKT(s, o, inc, arrayK, arrayS,
arrayG) for s in range(n_states)])
12        arrayTr=np.multiply(1-arrayT,arrayK) + arrayT
13        arrayK = np.multiply(arrayK, 1 - df.iloc[t].values) + np.
multiply(arrayTr, df.iloc[t].values)
14        delta = np.vstack((delta,arrayK))
15    return pd.DataFrame(delta).to_html()
16
17 def prob_cKT(s, obs, inc, arrayK, arrayS, arrayG):
18     if inc[s] == 0:
19         return arrayK[s]
20     if obs == 0:
21         return probability_conjunctiveKT_given_error(s, inc,
arrayK, arrayS, arrayG)
22     return probability_conjunctiveKT_given_correct(s, inc, arrayK,
arrayS, arrayG)
23
24
25 def probability_conjunctiveKT_error_konwKC(s, inc, arrayK, arrayS,
arrayG):
26     arrayR = np.multiply(1-arrayS,arrayK)+np.multiply(arrayG,1-
arrayK)
27     arrayR[s] = 1;
28     prob=arrayS[s]+(1-arrayS[s])*(1-np.prod(arrayR[inc]))
29     return prob
30
31 def probability_conjunctiveKT_correct(inc, arrayK, arrayS, arrayG)
:
32     arrayR=(np.multiply(1-arrayS,arrayK)+np.multiply(arrayG,1-
arrayK))
33     prob=np.prod(arrayR[inc])
34     return prob
35
36 def probability_conjunctiveKT_given_error(s, inc, arrayK, arrayS,
arrayG):
37     prob_egK = probability_conjunctiveKT_error_konwKC(s, inc,
arrayK, arrayS, arrayG)
38     prob_e = 1 - probability_conjunctiveKT_correct(inc, arrayK,
arrayS, arrayG)
39     prob_k = arrayK[s]

```

```
40     prob = prob_egK * prob_k / prob_e
41     return prob
42
43 def probability_conjunctiveKT_given_correct(s, inc, arrayK, arrayS
44 , arrayG):
45     prob_cgK = 1 - probability_conjunctiveKT_error_konwKC(s, inc,
46 arrayK, arrayS, arrayG)
47     prob_c = probability_conjunctiveKT_correct(inc, arrayK, arrayS
48 , arrayG)
49     prob_k = arrayK[s]
50     prob = prob_cgK * prob_k / prob_c
51     return prob
52
53 def parameters(slipping, guessing, learning, n_states):
54     arrayS = np.full(n_states, slipping)
55     arrayG = np.full(n_states, guessing)
56     arrayT = np.full(n_states, learning)
57     return (arrayS, arrayG, arrayT)
58
59 def parameters_score(n_states, df, t):
60     arrayG = (1-df.iloc[t].values)*0.2
61     arrayG[arrayG == 0] = 0.02
62     #arrayS = np.full(n_states, 0.05)
63     arrayS = df.iloc[t].values*0.05
64     arrayS[arrayS == 0] = 0.05/10
65     arrayT = np.full(n_states, 0.25)
66     print('arrayG', arrayG )
67     print('arrayS', arrayS)
68     return (arrayS, arrayG, arrayT)
```

Listing A.19: Funciones para la implementación del modelo CKT mejorado

Apéndice B

Documentos para la evaluación

B.1. Pre-test presentado a los estudiantes

Código identificativo alumno: _____

Contestar si las afirmaciones son verdaderas o falsas.
--

1. En un movimiento uniformemente acelerado la velocidad aumenta linealmente al aumentar del tiempo.

V F

2. En un movimiento uniformemente acelerado la aceleración aumenta linealmente al aumentar del tiempo.

V F

3. La aceleración de gravedad en la tierra mide 9.8 m/s^2 .

V F

4. Un cuerpo en caída libre no tiene aceleración.

V F

5. El tiempo que emplea en caer una pelota de masa 1 Kg, es igual al tiempo que emplea una pelota del mismo diámetro con una masa de 500gr.

V F

6. La aceleración de un cuerpo en caída libre depende de la posición inicial del cuerpo.

V F

7. Un cuerpo en caída libre (sin considerar la resistencia del aire) emplea un segundo para que su velocidad llegue a 9.8 m/s.

V F

8. Un cuerpo en caída libre desde una altura h llega con velocidad v al suelo.

V F

9. Considero interesante las clases de física.

V F

10. Las clases de física son difíciles.

V F

11. La física que aprendo en la escuela ha aumentado mi curiosidad en relación a los contenidos que voy descubriendo.

V F

12. Las clases de física me han hecho más crítico.

V F

13. Me gustaría hacer más clases de física.

V F

B.2. Post-test presentado a los estudiantes

Código identificativo alumno: _____

Examen para evaluar los contenidos adquiridos sobre el movimiento
rectilíneo uniformemente acelerado.

1. Identifica el gráfico que representa correctamente la relación espacio/tiempo en un objeto que se mueve sobre un plano inclinado.

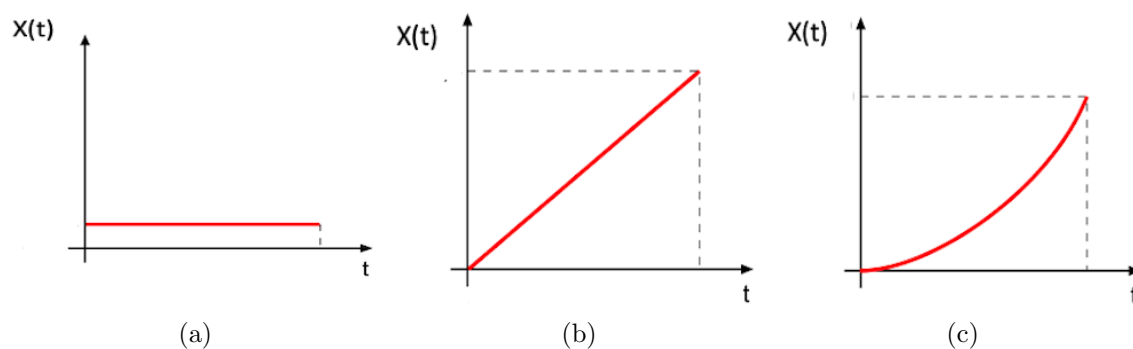


Figura B.1: Gráfico espacio-tiempo

2. Identifica el gráfico que representa correctamente la relación velocidad/- tiempo en un objeto que se mueve sobre un plano inclinado.

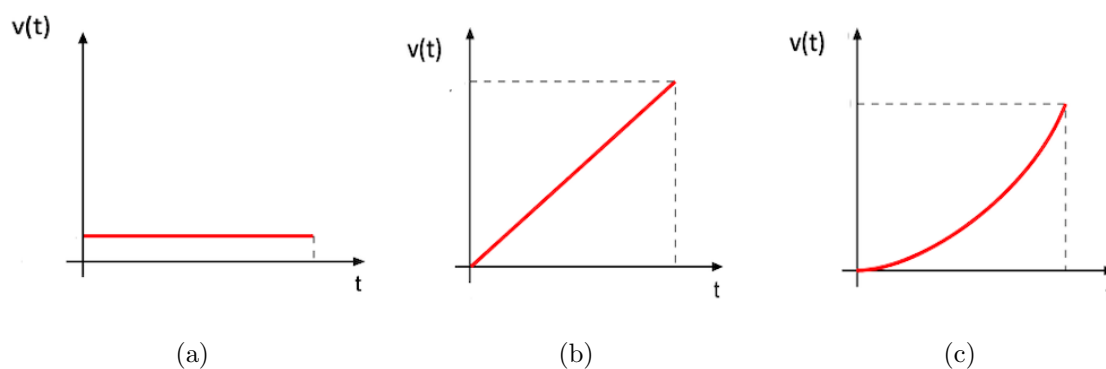


Figura B.2: Gráfico velocidad-tiempo

3. En un movimiento uniformemente acelerado la distancia es:
 - a) inversamente proporcional al tiempo al cuadrado;
 - b) directamente proporcional a la aceleración;
 - c) directamente proporcional a la velocidad.

4. Completa en modo oportuno la siguiente frase:

Considerado un gráfico velocidad-tiempo podemos deducir el movimiento del punto material.

Si el gráfico es una recta al eje t del tiempo, significa que el movimiento es

Si la recta no es paralela al eje t del tiempo, el movimiento es

si la recta pasa por el centro de los ejes, significa que en el sistema de referencia utilizado la velocidad inicial es

5. Dos objetos, con la misma forma, empiezan a caer juntos desde la misma altura. En la realidad:

- a) llegará al suelo primero el objeto que pesa más.
- b) llegaran al suelo en el mismo instante;
- c) llegaran al suelo en instantes de tiempo diferentes.

6. Dos objetos, con la misma forma, empiezan a caer juntos desde la misma altura. Sin resistencia del aire:

- a) llegará al suelo primero el objeto que pesa más.
- b) llegarán al suelo en el mismo instante;
- c) llegarán al suelo en instantes de tiempo diferentes.

7. Un automóvil que está parado, se mueve con aceleración constante a en un tiempo t , recorriendo una distancia s . al tener una aceleración doble, en el mismo tiempo recorrería una distancia:

- a) doble;
- b) reducida a la mitad;
- c) cuádruple.

8. Un cuerpo que cae sin resistencia del aire se mueve de movimiento:

- a) uniformemente acelerado;
- b) rectilíneo uniforme;
- c) acelerado.

9. Una pelota que rueda sobre un plano inclinado sin fricción, se mueve de movimiento:

- a) uniformemente acelerado;

- b) rectilíneo uniforme;
- c) acelerado.

10. Problema. Se considera tres planos inclinados de altura 2m e inclinaciones distintas.

- El plano A tiene una inclinación de 30 grados.
- El plano B tiene una inclinación de 45 grados.
- El plano C tiene una inclinación de 60 grados.

Tres cuerpos de masa respectivamente 5Kg, 3kg y 2Kg rodean sobre los tres planos A, B y C. Calcula:

- a) la aceleración con la que se mueven los tres cuerpos.
- b) La velocidad final de cada cuerpo. ¿Qué puedes observar?;
- c) ¿El tiempo de caída de cada cuerpo será diferente? Justifica la respuesta y observa si hay datos no utilizados en contestar a esta última pregunta.

Apéndice C

Construcción del plano robótico con LEGO EV3

Se muestra el plano inclinado construido con el Kit robótico EV3 para que el estudiante pueda medir el tiempo de caída de una pelota.

Se ha utilizado un motor grande para la subida del plano, dos sensores de contacto; uno para determinar el instante de inicio del tiempo y uno el momento en que termina la caída. El motor pequeño activa el movimiento de la pelota.

En figura C.1 se puede visualizar el plano robótico y en Figura C.2 los mecanismos para su correcto funcionamiento.



Figura C.1: Plano robótico con LEGO EV3.

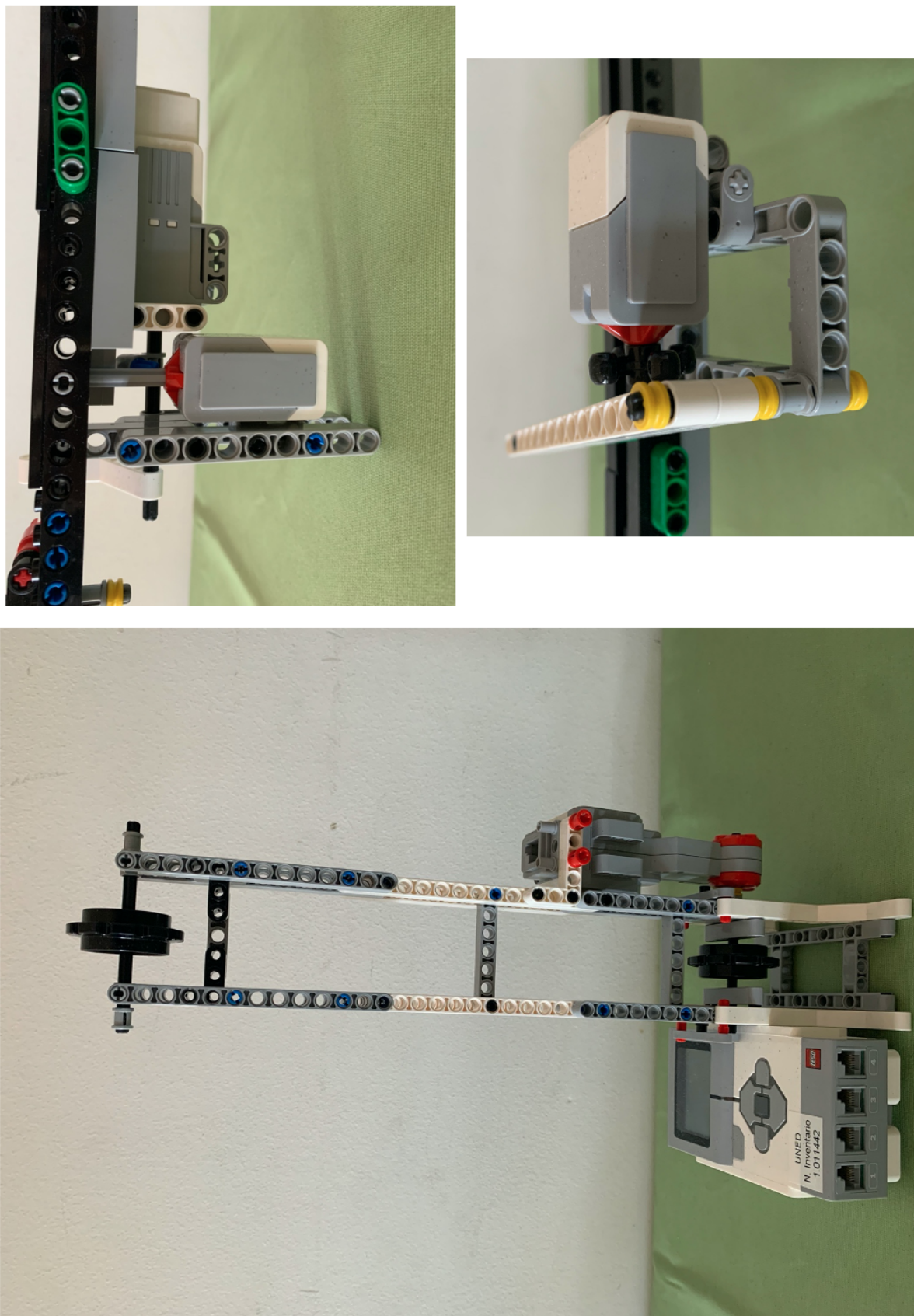


Figura C.2: Detalles plano robótico con LEGO EV3.