

SISTEMA DE DETECCIÓN DEL TRASTORNO DEL ESPECTRO DEL AUTISMO (TEA) CON EL ROBOT NAO BASADO EN EL Q-CHAT-10

Rubén Romero-García



Máster en Inteligencia Artificial Avanzada

Universidad Nacional de Educación a Distancia (UNED)

20 de agosto de 2019

Director/es:

Félix de la Paz López
Rafael Martínez de Tomás

Índice general

Agradecimientos	vii
Resumen	viii
Abstract	ix
1 Introducción	1
1.1 Objetivos y motivación	2
1.2 Estructura de la memoria	3
2 Revisión de la bibliografía	5
2.1 Robots humanoides. El robot NAO	6
2.2 El robot NAO como herramienta para terapia, evaluación y diagnóstico	9
2.3 El trastorno del espectro autista o TEA	12
2.4 Métodos automáticos para la detección y evaluación del TEA .	14
2.5 Propuesta de trabajo	17
3 Diseño de la solución	19
3.1 Infraestructura	19
3.2 Actividades	21
3.3 Clasificador	35
3.3.1 Conjunto de datos y preprocesamiento	36
3.3.2 Modelo de aprendizaje automático	39
3.3.3 Ingeniería de datos: lanzamiento y mantenimiento del sistema	40
4 Desarrollo	41
4.1 Infraestructura	41
4.1.1 Sistema operativo, framework web e IDE	44

4.1.2	Buenas prácticas en el desarrollo de software	46
4.1.3	Registro de usuarios	47
4.1.4	Interfaz de usuario. Comunicación	51
4.1.5	Comunicación con el robot	56
4.2	Actividades	59
4.2.1	Estructura de la base de datos	61
4.3	Modelo clasificador	61
4.3.1	Modelado y entrenamiento	61
4.3.2	Preprocesado de los datos	69
4.3.3	Resultados del usuario	70
4.3.4	Integración de los datos extraídos: actualización del modelo	73
5	Evaluación. Análisis de resultados	75
6	Conclusiones. Trabajos futuros	83
6.1	Conclusiones	83
6.2	Trabajos futuros	84
	Bibliografía	86
A	Anexo 1: Código	97

Índice de figuras

2.1	Robot NAO.	7
2.2	Componentes del software NAOqi.	8
2.3	Interfaz gráfica de programación de NAO, Choregraphe.	9
3.1	Distribuciones de ambos conjuntos de datos, por pregunta.	38
4.1	Diagrama del sistema.	42
4.2	Pantalla de registro de nuevo usuario (detalle).	48
4.3	Jerarquía de los grupos de usuarios.	49
4.4	Datos generales de la pantalla de perfil de usuario.	50
4.5	Datos de las cuatro preguntas del Q-CHAT-10 no implementadas en el perfil del usuario.	50
4.6	Menú de actividades.	51
4.7	<i>Pop-up</i> con la descripción de la actividad.	52
4.8	Flujo de ejecución del sistema.	53
4.9	Interfaz de actividad con los componentes estándar.	54
4.10	Interfaz de actividad con el usuario seleccionado.	55
4.11	Aviso de error: la actividad no puede detenerse sin un motivo.	55
4.12	Lista de usuarios.	71
4.13	Resultados finales del usuario.	71
5.1	Valor relativo de los parámetros.	79
5.2	Correlación entre los diferentes parámetros.	80

Índice de tablas

5.1	Datos obtenidos en función del modelo.	76
-----	--	----

Índice de fragmentos de código

4.1	Ejecución del script de conversación	58
4.2	Script de conversación	59
4.3	Almacenamiento del modelo	69
4.4	Inferencia a partir de los datos del usuario.	71
5.1	Peso relativo de las características	77

Familia, amigos y Andrea, claro, como siempre.

Y al gatín, que no entiende.

Поехали!

Agradecimientos

Agradecer la ayuda y consejo prestado a todo el equipo de psicología que ha formado parte de este TFM, sin el cuál nada de esto habría sido posible: Esther de la Paz, Marta Gutiérrez, Juan Luesia y Encarnación Sarriá.

Gracias también a los dos tutores, Félix de la Paz y Rafael Martínez, por su supervisión continuada y su apoyo con la redacción de la memoria.

Gracias a José Manuel Cuadra por toda su ayuda a la hora de aprender Machine Learning partiendo desde el desarrollo de software.

Y gracias al Programa Operativo de Empleo Juvenil y la Iniciativa de Empleo Juvenil (YEI), que ha financiado este proyecto.

Resumen

El uso de robots humanoides como asistentes en procesos de terapia no es nuevo: existen estudios desde hace años que muestran resultados prometedores cuando se combinan las técnicas habituales con la interacción del paciente con un robot. En este trabajo de fin de máster se pretende dar un paso en esa dirección, diseñando e implementando un sistema de detección temprana con el apoyo del robot Nao con el que un experto pueda analizar las dificultades específicas asociadas al trastorno del espectro del autismo.

Palabras clave

Robots humanoides, interacción humano-máquina, sistema clasificador, trastorno del espectro del autismo.

Abstract

The use of humanoid robots as assistants in therapy processes is not new at all: there are papers from several years ago that show promising results when combining human-robot interaction with the standard techniques. The goal of this MSc Thesis is pushing forward on that direction, designing and implementing a screening system supported by a Nao robot which could be used by an expert to early detect specific difficulties associated to autism spectrum disorder.

Keywords

Humanoid robots, human-computer interaction, classifier system, autism spectrum disorder.

Capítulo 1

Introducción

A lo largo de esta memoria se presenta un trabajo de fin de máster (TFM) cuya finalidad es diseñar e implementar un sistema de detección temprana con el apoyo del robot Nao basado en técnicas de aprendizaje automático con el que un experto pueda analizar las dificultades específicas asociadas al trastorno del espectro del autismo. La implementación incluye todo lo necesario para llevar a cabo un estudio real: interfaz de usuario, actividades de valoración, sistema de extracción y preprocesamiento de datos y finalmente sistema de análisis, con un modelo clasificador, que procesa los datos y presenta una serie de conclusiones.

1.1 Objetivos y motivación

Al hablar de los objetivos es preciso diferenciar entre los del proyecto en el que se enmarca este trabajo, más relacionados con la psicología y la terapia, y los del trabajo en sí, más enfocados a la inteligencia artificial.

El objetivo del proyecto es implementar una serie de pruebas complementarias a los diagnósticos estandarizados, ADOS y ADI-R ([American Psychiatric Association, 2013](#); [Le Couteur et al., 2008](#); [Mazefsky and Oswald, 2006](#)), basadas en el Q-CHAT-10 ([Allison et al., 2012](#)) que realicen un primer análisis de detección del trastorno del espectro autista. Las pruebas serán integradas en un sistema que clasifique automáticamente al sujeto según la presencia o no de este trastorno, detectándolo de forma temprana, lo que serviría para orientar la planificación de intervención psicoeducativa y para el seguimiento y evaluación de mejora por efecto de la intervención.

Por otro lado, en el contexto de la investigación en inteligencia artificial el objetivo principal de este trabajo es implementar un sistema clasificador que pueda efectuar una valoración general de un sujeto, es decir: analizar con un grado aceptable de probabilidad si dados los resultados de las pruebas, el paciente presenta el trastorno del espectro autista. Además, hablando únicamente a nivel técnico, otro de los objetivos es que la herramienta de evaluación, basada en el conocimiento existente proporcionado por expertos, pueda ser usada por un terapeuta sin conocimientos en informática ni tecnología.

La motivación a nivel personal es integrar los conocimientos previos en diseño y desarrollo de software con los adquiridos durante el máster en inteligencia artificial para conseguir un impacto real; lograr, aunque sea mínima, una mejora en la calidad de vida de sujetos con TEA, ya sea porque el sistema agiliza la evaluación, minimizando el tiempo de espera, ya sea porque se convierte en la primera piedra de un sistema más amplio de apoyo a la terapia.

Por último, es conveniente indicar que al ser este un trabajo de naturaleza sintética no se pretende tanto poner a prueba una hipótesis como construir un sistema informático capaz de obtener y procesar datos con significado y extraer conclusiones; la adaptación de las actividades del Q-CHAT-10 se ha llevado a cabo con la supervisión de varios psicólogos expertos en TEA, mientras que los datos empleados para la evaluación del sistema han sido extraídos de un experimento previo, ya que la extracción estaba enormemente limitada por el tamaño de la población necesaria. El enfoque ha estado dirigido a la evaluación del comportamiento del algoritmo y la selección del más apropiado de entre las alternativas.

1.2 Estructura de la memoria

Se presenta una breve introducción con la revisión de la bibliografía, que incluye información sobre la terapia con robots humanoides y la evaluación del trastorno del espectro autista con métodos automáticos. A continuación,

se describirá el proyecto a nivel técnico, con el análisis de alternativas y diseño de la solución implementada. El siguiente apartado profundiza en estos aspectos, detallando la implementación realizada dividida en tres grandes bloques: infraestructura e interfaz de usuario, actividades con el robot y análisis y procesado de datos. Se incluye también un estudio de prueba, con un conjunto de datos preexistente, para mostrar la integración de todo el sistema. Por último se puede encontrar una breve conclusión que incluye el peso que puede tener este proyecto en líneas de trabajo futuras.

Capítulo 2

Revisión de la bibliografía

Se presenta en este capítulo una revisión bibliográfica sobre el tema tratado. Con un enfoque *bottom up*, se describirán primero las bases: robots humanoides y la influencia de la tecnología en la terapia, así como el trastorno del espectro autista. A continuación, se hará un resumen de los estudios actuales sobre gamificación y sus resultados en la terapia de niños autistas, centrada en el robot Nao; finalmente, se estudiará el uso de métodos automáticos para detección y evaluación del trastorno del espectro autista, presentando por último la propuesta de trabajo.

2.1 Robots humanoides. El robot NAO

Los robots humanoides son aquellos que imitan la forma del cuerpo humano. La definición no es estricta: puede ser una imitación total, incluyendo rasgos faciales como boca u ojos, o esquemática, con simplemente piernas, torso y cabeza. Los objetivos son múltiples; desde la investigación del movimiento bípedo ([Yamaguchi et al., 1999](#)) hasta la interacción con humanos en entornos controlados, como en el presente trabajo. A su vez, existen multitud de modelos; su análisis detallado podría servir de base para otro TFM: el robot humanoide de Honda ([Hirai et al., 1998](#)), el HRP-3 ([Kaneko et al., 2008](#)), el WABIAN-2 ([Ogura et al., 2006](#)) o el KHR-3 ([Park et al., 2005](#)), por poner solo algunos ejemplos.

En este trabajo se ha utilizado un robot NAO, visible en la [Figura 2.1](#). Desarrollado por Softbank Robotics y presentado como su primer robot, es una herramienta programable especialmente útil en ámbitos educativos y de investigación ([SoftBank Robotics, 2019a](#)). Sus dimensiones son de $574 \times 311 \times 275$, con un procesador Atom E3845 Quad Core a 1.91 GHz y 4GB de RAM. Consta de dos altavoces en los laterales de la cabeza, cuatro micrófonos omnidireccionales y dos videocámaras de resolución 2560×1920 a 1 frame por segundo o 640×480 a 30 frames por segundo, lo cuál las hace poco apropiadas para aplicaciones en las que la visión sea un factor crítico. Consta también de sónars, giroscopios y sensores resistivos de fuerza en las manos, que son capaces de medir la presión aplicada, además de una serie de LEDs. Tiene también incorporada una suite de idiomas que permite al robot



Figura 2.1: Robot NAO.

hablar imitando diferentes acentos.

El software del robot, llamado NAOqi, incluye un framework para facilitar la programación: proporciona apoyo y soluciones para paralelismo, uso y gestión de recursos, sincronización, eventos, etc. El fabricante proporciona APIs en varios lenguajes de programación, siendo los más frecuentes y usados C++ y Python; con ellos se puede acceder a los diferentes módulos del robot y ejecutar una serie de acciones, como se ve en la Figura 2.2. La programación puede efectuarse de varias maneras: enviando directamente las órdenes por medio de un script, diseñando programas y cargándolos en el robot, o incluso mediante una interfaz gráfica, Choregraphe (Figura 2.3). Entre los casos de uso y aplicaciones más frecuentes destacan la enseñanza, mediante actividades personalizadas para pequeños grupos o estudiantes individuales, que permiten extraer y monitorizar resultados; la venta al por menor, para

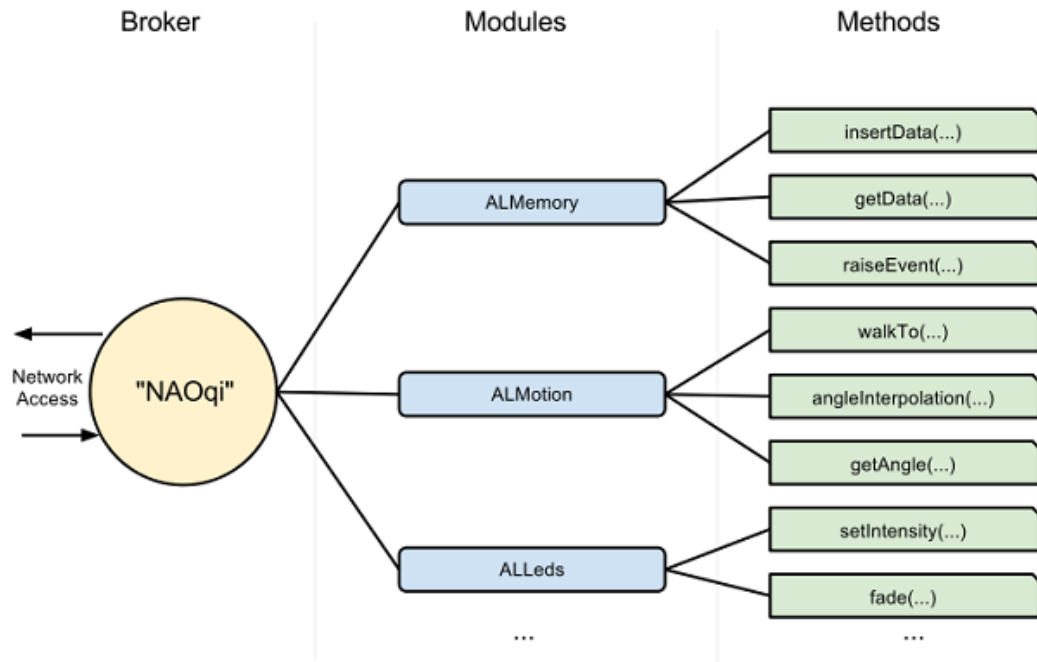


Figura 2.2: Componentes del software NAOqi.

obtener una interacción directa y diferente con el cliente; el turismo, como método de bienvenida a hoteles y embajador de marcas; y la salud, asistiendo a pacientes para llevar a cabo un autodiagnóstico, como herramienta de apoyo para telemedicina o para monitorizar la evolución de resultados. Es en este último área donde se enmarca este TFM.

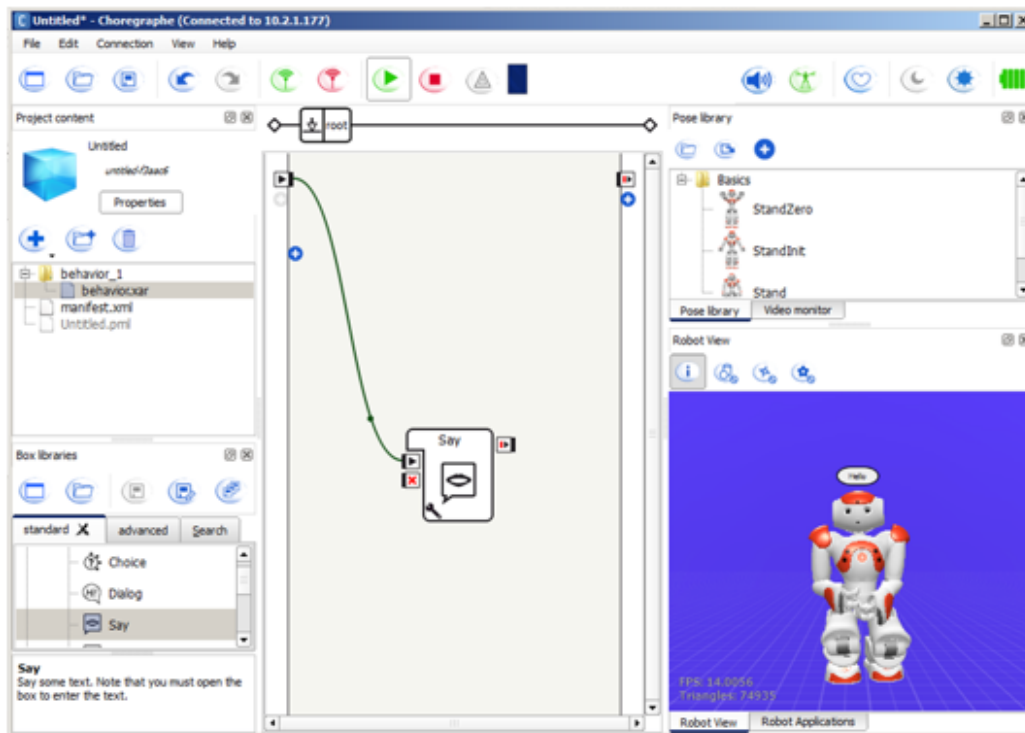


Figura 2.3: Interfaz gráfica de programación de NAO, Choregraphe.

2.2 El robot NAO como herramienta para terapia, evaluación y diagnóstico

En 2012, la ENSTA de París y la Universidad de Babes-Bolyai de Rumanía presentaron un estudio basado en la interacción de niños autistas con un robot NAO, comparando el grado de la interacción social en un entorno humano-robot con respecto a un entorno humano-humano (Tapus et al., 2012). El estudio consistía en cuatro actividades donde el robot imitaba el movimiento del tren superior de los niños, siendo los criterios de medición el desvío de la

mirada, la presencia de sonrisas o el inicio libre de movimiento buscando la imitación del robot, entre otros. Los resultados presentaban una variabilidad muy alta: varios niños no mostraban efectos visibles, otros mantenían la mirada fija o sonreían más, y solo uno de los sujetos de estudio mostró mejoría en todos los criterios. Aunque no especialmente positivos, estos resultados abrieron camino; el robot NAO parecía funcionar *en algunos casos*.

En el mismo año y de forma independiente, la Universidad Tecnológica MARA de Malasia (UiTM) investigaba la influencia de la interacción humano-robot (y en concreto, la del robot NAO) en el marco de un programa de intervención (Ismail et al., 2012b). Las respuestas iniciales en cuanto al contacto visual indicaban una ligera mejora general cuando la interacción se llevaba a cabo con un robot con respecto a cuando se llevaba a cabo con un humano (Ismail et al., 2012a). Continuando con este enfoque, la UiTM desarrolló un framework de actividades con el robot NAO para la rehabilitación y el apoyo continuo en la terapia de niños con trastorno del espectro autista (Shamsuddin et al., 2012a). En un estudio piloto, la UiTM plantea la hipótesis de que la interacción humano robot entre un niño con TEA y un NAO podría aumentar el interés del niño en comunicarse, disminuyendo de forma significativa las características comunicacionales asociadas al autismo (Shamsuddin et al., 2012b). Entre los motivos para tal hipótesis destacan la apariencia humana del robot, su capacidad de guiñar los ojos, hablar y reproducir música. Los resultados muestran que 4 de los 5 niños tratados muestran un descenso del comportamiento autista en el ámbito comunicativo durante los experimentos. En 2013, la misma universidad presentó un estudio en los

que el robot NAO lograba llamar la atención de un niño y enseñarle conceptos acerca de las emociones, de nuevo mediante un enfoque basado en juegos y canciones; en concreto, la postura corporal del robot se asociaba con una emoción que el niño debe adivinar ([Shamsuddin et al., 2013](#)). Las conclusiones, esta vez sí, mostraban resultados esperanzadores sobre la influencia del robot en la rehabilitación de niños con TEA.

Desde entonces se han presentado diversos estudios en la misma línea: la UiTM, que ha dedicado muchos esfuerzos a este tema, tiene artículos focalizados en la influencia de NAO en la interacción social y su efecto en la rehabilitación de niños con TEA, basados en los seis elementos de la *Gilliam Autism Rating Scale-second edition* (GARS-2), mostrando efectos positivos principalmente en la comunicación bidireccional y en la evolución del aprendizaje y la interacción social ([Miskam et al., 2013](#)). También hay publicados estudios sobre el uso de NAO como una herramienta de enseñanza para el reconocimiento de emociones en un entorno móvil ([Miskam et al., 2014b](#)) o para mejorar las habilidades sociales ([Miskam et al., 2014a](#)). Al otro lado del mundo, la Universidad Internacional de Florida indica que los robots humanoides pueden influir positivamente en las habilidades sociales y de comunicación entre niños con Trastorno del Espectro Autista; concretamente, muestra mejoría en la mirada y la atención conjunta ([Feng et al., 2013](#)). Para medirlo, los autores diseñaron dos juegos en los que se pedía responder a unas preguntas y tocar la cabeza del robot, activando uno de los sensores, sosteniéndole la mirada. NAO dispone de un módulo diseñado específicamente para mantener la mirada de la persona con la que interactúa, pero los investigadores utili-

zaron cámaras externas para medir los resultados, probablemente debido a la baja calidad de los sensores del robot.

En general, los estudios coinciden en que un entorno basado en el robot NAO influye positivamente en evaluación y terapia, aunque el grado de mejoría es altamente variable.

2.3 El trastorno del espectro autista o TEA

El Trastorno del Espectro del Autismo (TEA) es un trastorno del neurodesarrollo. Según el Manual Diagnóstico y Estadístico de los Trastornos Mentales ([American Psychiatric Association, 2013](#)) los criterios que definen el trastorno se dividen en dos grandes bloques: déficits persistentes en la comunicación y en la interacción social y patrones repetitivos y restringidos de conducta, actividades e intereses. Existe además una serie de especificaciones previas que deben ser conocidas ([American Psychiatric Association, 2013](#)). El primer bloque consta de tres elementos, que deben aparecer, mientras que el segundo bloque consta de cinco aspectos de los que al menos deben aparecer dos.

Déficits persistentes en la comunicación y en la interacción social

- Dificultades en reciprocidad socio-emocional: dificultades relacionadas con la interacción social, con tener acercamientos sociales inusuales, dificultades para mantener conversaciones fluidas y guardar turnos de

respuestas, entre otras.

- Déficits en conductas comunicativas no verbales utilizadas en la interacción social: dificultades para la comunicación verbal y no verbal, déficit en la comprensión y uso de gestos, falta de expresividad emocional, entre otras.
- Dificultades para desarrollar, mantener y comprender las relaciones: dificultades, por ejemplo, para ajustar el comportamiento a los diferentes contextos y para compartir juegos de ficción.

Patrones repetitivos y restringidos de conductas, actividades e intereses

- Movimientos motores, utilización de objetos o habla estereotipados o repetitivos. Por ejemplo, estereotipias motoras simples, alineación de los juguetes o cambio de lugar de los objetos, ecolalias o frases idiosincrásicas.
- Adherencia excesiva a rutinas, patrones de comportamiento verbal y no verbal ritualizado o resistencia excesiva a los cambios. Por ejemplo, gran angustia frente a cambios pequeños, dificultades con las transiciones, patrones de pensamiento rígidos, rituales de saludo, necesidad de tomar el mismo camino o de comer los mismos alimentos cada día.
- Intereses muy restringidos y fijos que son anormales en cuanto a su intensidad o foco de interés. Por ejemplo, fuerte apego o preocupación

por objetos inusuales, intereses excesivamente circunscritos o perseverantes.

- Hiper o hiposensibilidad a los estímulos sensoriales o interés inusual en aspectos sensoriales del entorno. Por ejemplo, indiferencia aparente al dolor/temperatura, respuesta adversa a sonidos o texturas específicas, oler o tocar objetos en exceso, fascinación visual por las luces u objetos que giran.

Las personas con TEA muestran una gran variabilidad en la presentación de estos síntomas, pudiendo haber personas con una grave afectación y otras personas donde los síntomas se manifiestan de forma leve, lo que lleva a la existencia de un amplio espectro de severidad dentro del trastorno.

2.4 Métodos automáticos para la detección y evaluación del TEA

Existen numerosos estudios que buscan la confluencia entre la detección, diagnóstico y evaluación del autismo y la inteligencia artificial. Uno de los más populares, de la Harvard Medical School, data de 2012: aplica métodos de aprendizaje automático sobre un conjunto de datos procedente del ADOS (*Autism Diagnostic Observation Schedule-Generic*), el instrumento de diagnóstico de trastornos del espectro autista más usado, detectando que con un pequeño porcentaje de características sobre el total del test es suficiente

para clasificar el autismo ([Wall et al., 2012](#)). Sucesivos análisis rebajan ligeramente las expectativas, pero el estudio concluye que es posible desarrollar herramientas de evaluación preliminar para acelerar la detección precoz. En 2015, David Bone, de la Universidad del Sudeste de California, presenta un extenso artículo valorando las posibilidades de la aplicación de inteligencia artificial al diagnóstico del autismo. Estrechamente relacionado con el estudio anterior de Wall, Bone intenta replicar los resultados obtenidos, sin éxito. Indica que "a pesar de que el aprendizaje automático tiene un inmenso potencial para mejorar el diagnóstico y la intervención, su uso en ausencia de un conocimiento del dominio clínico puede llevar a conclusiones erróneas" ([Bone et al., 2015](#)). Propone, por tanto, una serie de buenas prácticas a la hora de investigar el autismo con métodos automáticos.

En 2015 Alessandro Crippa, del Instituto de Imagen Molecular y Fisiología de Milán, presenta un artículo sobre el uso de aprendizaje automático para la identificación de anomalías motoras en niños con TEA, logrando una clasificación máxima del 96.7% utilizando siete características relacionadas con el movimiento del tren superior ([Crippa et al., 2015](#)). Los autores avisan de que el conjunto de individuos analizado es pequeño y muy definido: niños de 2 a 4 años con autismo diagnosticado y bajo nivel de movimiento funcional; no se estudia si la generalización del método es buena para otro tipo de pacientes, uno de los puntos que señalaba Bone en su artículo. En 2016, la Universidad de Stanford publica un estudio en el que se definen cinco características (de un grupo inicial de 65) que bastan para diferenciar entre sujetos con trastorno del espectro autista y sujetos con trastorno de

déficit de atención e hiperactividad (Duda et al., 2016). Más recientemente, Fadi Thabtah del Instituto Tecnológico Nelson Marlborough de Auckland ha publicado trabajos analizando el estado de la cuestión de diferentes métodos de aprendizaje automático aplicados al diagnóstico del TEA (Thabtah, 2017, 2018b; Thabtah et al., 2018) abordando el problema de que muchos de ellos están desactualizados en cuanto al conocimiento del dominio, ya que emplean el DSM-IV en vez del más reciente DSM-V (de nuevo, como decía Bone, problemas por falta de conocimiento clínico). Propone métodos a seguir con respecto a la conceptualización, la implementación y los datos, y en sus últimos artículos, de 2018, empieza a investigar nuevos enfoques dentro del aprendizaje máquina para potenciar los buenos resultados obtenidos.

De especial importancia para este trabajo, aunque no sea un método de detección automática, es el artículo de 2012 de Allison et al en que se define un método que funciona como prediagnóstico para niños y niñas a partir de 18 meses (Allison et al., 2012). La idea de los autores es definir una *bandera roja*, un signo que pueda usarse como herramienta de decisión temprana para o bien descartar completamente la presencia de TEA, o bien marcar al sujeto para una revisión exhaustiva posterior. La "lista cuantitativa de elementos que denotan TEA en niños" o Q-CHAT (*Quantitative Checklist for Autism in Toddlers*) es un conjunto de preguntas dirigidas a los padres o cuidadores que sirven a ese propósito. Partiendo de medio centenar, los autores refinan el método estadísticamente hasta encontrar las diez preguntas que más peso tienen, y que sirven de base al trabajo desarrollado aquí. Se ampliará la información sobre este artículo en el capítulo 3.

A día de hoy no existe un consenso en el mundo académico sobre cómo afrontar la confluencia entre el aprendizaje automático y la detección, diagnóstico y evaluación del trastorno del espectro autista. Lo que sí se puede afirmar es que la variabilidad de resultados es muy grande, ya que la población con TEA es muy heterogénea, y hay, en general, un ambiente de cautela a la hora de anunciar resultados generalizables.

2.5 Propuesta de trabajo

El trabajo a realizar, desde un enfoque muy de alto nivel, se puede resumir en: implementar en software alguna prueba de diagnóstico, lo más rápida y sencilla posible, empleando un robot NAO como actor principal para la interacción con el sujeto, y almacenar los datos del usuario para clasificarlo con métodos automáticos como TEA o neurotípico tan pronto como la termine. En los siguientes dos capítulos se describe en detalle el proceso seguido para alcanzar este objetivo, pormenorizando la selección entre las diversas alternativas y explicando los diferentes problemas encontrados y su solución, hasta llegar al producto final.

Capítulo 3

Diseño de la solución

Este capítulo describe el problema a resolver y la solución escogida: las actividades, que son el núcleo del sistema, y las especificaciones técnicas: servicio web y modelo clasificador.

3.1 Infraestructura

Cuando se crea un proyecto desde cero, sin limitaciones procedentes del código heredado, el análisis se convierte en el paso más importante ([Bass et al., 2003](#); [Feathers, 2004](#); [Shaw et al., 1996](#)). Las especificaciones técnicas vienen después; la toma de requisitos es la primera tarea ([Ghezzi et al., 2002](#); [Royce, 1987](#)). De este modo, se definieron una serie de exigencias: por muy bueno que sea el sistema clasificador, no sirve de nada si la extracción de datos

no está, al menos, semi-automatizada, o si son necesarios conocimientos en ingeniería para realizar el procesado. El módulo de inteligencia artificial debe ser una parte de un mecanismo mayor, que permita que el sistema se utilice en un entorno real. Con esto en mente, se trazaron las líneas rojas que el sistema debía respetar:

- Versatilidad: entendida como multiplataforma; es conveniente reducir al mínimo las especificaciones técnicas necesarias, ya que no puede asumirse que se va a disponer de la última tecnología en los centros de extracción.
- Usabilidad: el sistema será usado en igual medida por personas con perfil técnico y por terapeutas; no es admisible que se requieran conocimientos avanzados de ingeniería o inteligencia artificial para poder obtener resultados.
- Fácil acceso: idealmente, el sistema debe poder ser accesible desde cualquier parte.
- Etiquetado automatizado: en el momento en el que se registre un dato, el sistema tiene que saber a qué persona, actividad y momento temporal pertenece, sin necesidad de indicarlo explícitamente en cada ocasión.

Es decir: se necesita un entorno multiplataforma, con un grado de complejidad asociado a la utilización mínimo; de ser posible, evitar instalaciones largas, uso de línea de comandos, atención constante por parte del usuario,

etc. Hace falta que el acceso al sistema sea rápido y fácil, para evitar depender del dispositivo en el que se ha instalado; y es necesario que exista una persistencia de los datos de usuario y que sea internamente única, de forma que el etiquetado pueda ser automático. Con todas estas características, la solución es obvia: debe ser un entorno web.

Una vez definido el tipo de infraestructura, los requisitos se convierten en:

- Versatilidad: soporte para múltiples navegadores.
- Usabilidad: interfaz de usuario amigable y sencilla.
- Fácil acceso: ya conseguido, por definición; basta con acceso a internet.
- Etiquetado automatizado: registro de usuarios centralizado.

Se analizarán estos requisitos en profundidad en el capítulo 4, implementación y desarrollo.

3.2 Actividades

El diagnóstico temprano del autismo es especialmente complejo debido a la alta variabilidad en los casos: los síntomas pueden manifestarse de forma muy sutil y ser pasados por alto ([Allison et al., 2012](#)). En su artículo, los autores indican la necesidad de diseñar una herramienta de diagnóstico temprano, que

pueda ser usada como *bandera roja*: un sistema de alerta que indique la posibilidad real de un problema. Plantean que una herramienta que proporcione un ratio de verdaderos positivos (VP) detectados del 80% sería suficiente como comienzo para mejorar la calidad de vida de los pacientes, facilitar el trabajo a las médicos y enfermeras de atención primaria y ahorrar costes en dinero público. Con esto en mente crearon el CHAT (*Checklist for Autism in Toddlers*), lista de elementos que identifican el TEA en niños centrada en los déficits persistentes en la comunicación y en la interacción social, y más concretamente en la falta de atención y ausencia de juego de roles. A pesar de unos buenos resultados en la detección de verdaderos negativos (VN) con > 90% de casos correctamente etiquetados, el ratio de VP es de apenas un 40%. Por otro lado, el AQ (*Autism Spectrum Quotient*), test preexistente de 50 preguntas desarrollado para evaluar el grado de rasgos autistas en adultos con inteligencia media, presenta un ratio de VP del 80% consistente a lo largo del tiempo y para diferentes culturas. Ambas pruebas se combinan en una revisión del CHAT basada en el AQ, el Q-CHAT (*Quantitative Checklist for Autism in Toddlers*), test de 25 preguntas enfocado a niños de 18 a 30 meses de edad que discrimina entre aquellos que parecen desarrollar TEA y los que siguen un proceso de desarrollo neurotípico. Los autores efectúan un análisis estadístico para determinar cuáles de las 25 preguntas del Q-CHAT tienen un mayor peso en la decisión, extrayendo las diez más importantes para formar el Q-CHAT-10, que presenta un ratio de VP del 91% y de VN del 89%. Sin embargo, todos estos métodos presentan el problema de que las respuestas a las preguntas vienen dadas por el padre/madre o los cuidadores responsables

del niño, y por tanto están sujetas a un cierto grado de subjetividad.

La idea detrás de este proyecto es adaptar esas diez preguntas a un entorno gamificado, logrando obtener datos directos del niño mediante la evaluación *in situ* de un terapeuta que opere con el sistema. Las preguntas originales del Q-CHAT-10 son:

1. ¿Te mira tu hijo/a cuando le llamas por su nombre? En el original: *Does your child look at you when you call his/her name?*
2. ¿Cómo de sencillo es lograr un contacto visual con tu hijo/a? En el original: *How easy is it for you to get eye contact with your child?*
3. ¿Suele tu hijo/a señalar para indicar que quiere algo? (p.ej, un juguete fuera de su alcance) En el original: *Does your child point to indicate that s/he wants something? (e.g. a toy that is out of reach)*
4. ¿Suele tu hijo/a señalar para llamar tu atención sobre algo? (p.ej, un paisaje interesante) En el original: *Does your child point to share interest with you? (e.g. pointing at an interesting sight)*
5. ¿Juega tu hijo/a a juegos de roles? (p. ej, cuidar de muñecas como si fueran niños, hablar por un teléfono móvil de mentira...) En el original: *Does your child pretend? (e.g. care for dolls, talk on a toy phone)*
6. ¿Sigue tu hijo/a la dirección de tu mirada? En el original: *Does your child follow where you're looking?*

7. Si tú o alguien de la familia está claramente enfadado/molesto, ¿muestra tu hijo/a signos de querer confortarle? (p. ej, abrazarle, acariciarle el pelo) En el original: *If you or someone else in the family is visibly upset, does your child show signs of wanting to comfort them? (e.g. stroking hair, hugging them)*
8. Describirías sus primeras palabras como: (comunes, infrecuentes, raras...) En el original: *Would you describe your child's first words as (common, uncommon, rare...)*
9. ¿Usa tu hijo/a gestos simples? (p. ej, decir adiós con la mano) En el original: *Does your child use simple gestures? (e.g. wave goodbye)*
10. ¿Suele tu hijo mirar a la nada sin propósito aparente? En el original: *Does your child stare at nothing with no apparent purpose?*

Los parámetros de medición varían según la pregunta, pero siempre son cinco opciones. Para las preguntas 1 y 7, son:

- Siempre (*Always*)
- Generalmente (*Usually*)
- A veces (*Sometimes*)
- Raramente (*Rarely*)
- Nunca (*Never*)

Para la pregunta 2, son:

- Muy fácil (*Very easy*)
- Más o menos fácil (*Quite easy*)
- Más o menos difícil (*Quite difficult*)
- Muy difícil (*Very difficult*)
- Imposible (*Impossible*)

Para la pregunta 8, son:

- Muy comunes (*Very typical*)
- Más o menos comunes (*Quite typical*)
- Poco comunes (*Slightly unusual*)
- Muy infrecuentes (*Very unusual*)
- Mi hijo/a no habla (*My child doesn't speak*)

Para el resto de preguntas, son:

- Muchas veces al día (*Many times a day*)
- Unas pocas veces al día (*A few times a day*)

- Unas pocas veces a la semana (*A few times a week*)
- Menos de una vez al mes (*Less than once a month*)
- Nunca (*Never*)

Los autores del artículo original binarizan estas respuestas: las dos primeras opciones puntúan cero, mientras que las tres últimas puntúan uno (salvo en la décima pregunta, que invierte los valores). Sin embargo, en el sistema planteado aquí estas respuestas no tienen sentido: ¿Cómo puede un terapeuta, sin exposición previa al niño o niña, saber si el resultado de una de las pruebas ocurre *siempre* o solo *a veces*? La respuesta es fácil: no puede. Y la solución escogida, acordada con los expertos, ha sido binarizar las pruebas en origen; es decir: el terapeuta tan solo marcará sí o no, en función de si el niño responde según lo esperado o no, e idealmente, cada niño o niña efectuará varias sesiones en distintos momentos temporales, para obtener una muestra estadísticamente significativa.

No todas las preguntas son gamificables; en cuatro de los diez casos los expertos han propuesto que la información siga siendo proporcionada por la madre, padre o cuidadores responsables. En un futuro podría intentar realizar una gamificación completa; esta posibilidad será explorada más adelante, en el capítulo 6. De momento, las preguntas 5, 6, 8 y 10 quedan fuera del sistema y deben ser recogidas al dar de alta al paciente.

Actividad 1. ¿Te mira tu hijo/a cuando le llamas por su nombre?

Objetivo Comprobar si el niño reacciona al ser llamado por su nombre.

Procedimiento En primer lugar se hará pasar al sujeto a una sala acompañado por el terapeuta y se hará un contacto previo con el robot NAO. El objetivo de este punto es establecer un clima de confianza en el contexto terapéutico: es conveniente que el terapeuta introduzca al NAO al sujeto. el terapeuta debe situarse lo suficientemente cerca como para observar al sujeto en vivo y controlar aspectos del robot o de la actividad si es necesario.

Son necesarias una silla para el sujeto y una mesa donde se encontrará situado el robot NAO. Es conveniente que el robot se encuentre detrás del niño, para poder discernir mejor si se cumple el objetivo o no. Una vez el sujeto esté sentado, el robot se dirigirá a él, llamándolo por su nombre: "*¡Hola, Sara! Soy NAO, ¿qué tal estás?*". Basta una frase simple para que el terapeuta pueda valorar si existe una reacción al nombre.

Una vez marcada la reacción (o su ausencia), la actividad terminará. Al ser tan breve, es preferible que no exista un refuerzo ni una despedida. Es conveniente que esta sea la primera actividad de la sesión, ya que facilita que el flujo sea lógico y minimiza la necesidad de atención por parte del sujeto.

Actividad 2. ¿Cómo de sencillo es lograr un contacto visual con tu hijo/a?

Objetivo Comprobar si el niño mantiene el contacto visual.

Procedimiento En primer lugar se hará pasar al sujeto a una sala acompañado por el terapeuta y se hará un contacto previo con el robot NAO. El objetivo de este punto es establecer un clima de confianza en el contexto terapéutico: es conveniente que el terapeuta introduzca al NAO al sujeto. el terapeuta debe situarse lo suficientemente cerca como para observar al sujeto en vivo y controlar aspectos del robot o de la actividad si es necesario.

Son necesarias una silla para el sujeto y una mesa donde se encontrará situado el robot NAO. Es necesario que el robot se encuentre delante del niño, para poder establecer contacto visual. Una vez el sujeto esté sentado, el robot se dirigirá a él, mirándolo y llamándolo por su nombre: *"¡Hola, Sara! Soy NAO. ¿Cuál es tu animal favorito? El mío es el elefante. Me encantan, con esa trompa, ¡y los colmillos! (...)"*. Es conveniente que la frase dure lo suficiente como para que el terapeuta pueda valorar si le mantiene la mirada a Nao.

Actividad 3. ¿Suele tu hijo/a señalar para indicar que quiere algo?

Objetivo Comprobar si el niño realiza gestos socialmente comunes, como señalar.

Procedimiento En primer lugar se hará pasar al sujeto a una sala acompañado por el terapeuta y se hará un contacto previo con el robot NAO. El objetivo de este punto es establecer un clima de confianza en el contexto terapéutico: es conveniente que el terapeuta introduzca al NAO al sujeto. el terapeuta debe situarse lo suficientemente cerca como para observar al sujeto en vivo y controlar aspectos del robot o de la actividad si es necesario.

Son necesarias una silla para el sujeto y una mesa donde se encontrará situado el robot NAO. Es necesario que el robot se encuentre delante del niño, para poder establecer contacto visual. En un caso ideal se dispondría de algo que al niño le guste, como un juguete concreto; previsiblemente esto no siempre será posible, por lo que se ha adaptado ligeramente la actividad: sobre la mesa, entre NAO y el sujeto, debe haber dos láminas, una con un coche y otra en blanco. Una vez el sujeto esté sentado, el robot se dirigirá a él, mirándolo y llamándolo por su nombre: "*¡Hola, Sara! Soy NAO. ¿Te gustan los coches? ¡A mí me encantan! (...)*". El robot introducirá el tema, los coches, durante un breve discurso. A continuación, preguntará "*Oye, ¿dónde está el coche?*". El terapeuta apuntará si el niño señala o no (la precisión no es demasiado importante; si el niño señala hacia las láminas, es suficiente), y la actividad finalizará. Al acabar, Nao le agradecerá su ayuda al niño: "*Muchas gracias, Sara!*".

Actividad 4. ¿Suele tu hijo/a señalar para llamar tu atención sobre algo?

Objetivo Muy en línea con la actividad anterior, se trata de comprobar si el niño emplea gestos para llamar la atención, como señalar.

Procedimiento En primer lugar se hará pasar al sujeto a una sala acompañado por el terapeuta y se hará un contacto previo con el robot NAO. El objetivo de este punto es establecer un clima de confianza en el contexto terapéutico: es conveniente que el terapeuta introduzca al NAO al sujeto. el terapeuta debe situarse lo suficientemente cerca como para observar al sujeto en vivo y controlar aspectos del robot o de la actividad si es necesario.

Son necesarias una silla para el sujeto y una mesa donde se encontrará situado el robot NAO, además de un altavoz pequeño situado cerca (pero no al lado) del robot. Es necesario que el robot se encuentre delante del niño, para poder establecer contacto visual. Una vez el sujeto esté sentado, el robot se dirigirá a él, mirándolo y llamándolo por su nombre: *"¡Hola, Sara! Soy NAO. ¡Vamos a escuchar una canción!"*. A continuación, desde el altavoz empezará a sonar una canción. En el clímax, la canción debe detenerse abruptamente; es importante que el corte sea claro, inconfundible con un final natural. El terapeuta debe marcar si el niño señala a Nao o al altavoz, indicando que algo ha ocurrido. Al finalizar, Nao debe despedirse y decir algo como *Vaya, ¡no sé qué ha pasado! Lo siento"*.

Actividad 5. ¿Juega tu hijo/a a juegos de roles?

Objetivo El objetivo de esta actividad es determinar si el niño es capaz de ponerse en el lugar de otra persona/ser, mediante juegos.

Procedimiento En el artículo, Allison et al. la mencionan como una de las más críticas; esto, sumado a su complejidad, ha hecho que sea una de las que no se han gamificado. Al dar de alta al niño, se le preguntará a sus responsables si esto ocurre, con las cinco respuestas originales como posibilidad: muchas veces al día, unas pocas veces al día, unas pocas veces a la semana, menos de una vez al mes o nunca. La respuesta dada se binarizará según los criterios del artículo: muchas/unas pocas veces al día será 0; cualquier otra opción será 1.

Actividad 6. ¿Sigue tu hijo/a la dirección de tu mirada?

Objetivo En línea con una de las actividades previas, el objetivo es comprobar si, además de mantener la mirada, el niño es capaz de interpretar que debe seguirla para entender el contexto.

Procedimiento Al igual que la actividad anterior, es necesario que los responsables del niño proporcionen esta información, con las cinco respuestas originales como posibilidad: muchas veces al día, unas pocas veces al día, unas pocas veces a la semana, menos de una vez al mes o nunca. La respuesta dada

se binarizará según los criterios del artículo: muchas/unas pocas veces al día será 0; cualquier otra opción será 1.

Actividad 7. Si tú o alguien de la familia está claramente enfadado/molesto, ¿muestra tu hijo/a signos de querer confortarle?

Objetivo Analizar si el niño presenta rasgos empáticos y su alcance.

Procedimiento En primer lugar se hará pasar al sujeto a una sala acompañado por el terapeuta y se hará un contacto previo con el robot NAO. El objetivo de este punto es establecer un clima de confianza en el contexto terapéutico: es conveniente que el terapeuta introduzca al NAO al sujeto. el terapeuta debe situarse lo suficientemente cerca como para observar al sujeto en vivo y controlar aspectos del robot o de la actividad si es necesario.

Son necesarias una silla para el sujeto y una mesa donde se encontrará situado el robot NAO, así como un objeto (por ejemplo, un coche de juguete) que debe sostener Nao. Es necesario que el robot se encuentre delante del niño, para poder establecer contacto visual. Una vez el sujeto esté sentado, el robot se dirigirá a él, mirándolo y llamándolo por su nombre: "*¡Hola, Sara! Soy NAO. ¡Mira qué juguete tengo! Es nuevo, me lo ha regalado un amigo*". A continuación, Nao soltará el juguete, que caerá sobre la mesa. Nao debe mostrar aflicción inmediatamente: "*¡Oh, no! ¡Se ha roto!*". El terapeuta debe marcar si, según su criterio, el niño hace algún gesto de consuelo.

Actividad 8. Describirías sus primeras palabras como...

Objetivo La información sobre las primeras palabras del niño puede arrojar información sobre los intereses del niño y su adecuación al entorno; pretende medirse eso.

Procedimiento Por motivos obvios, es necesario que los responsables proporcionen esta información, con las cinco respuestas originales como posibilidad: muy comunes, más o menos comunes, poco comunes, muy infrecuentes o mi hijo/a no habla. La respuesta dada se binarizará según los criterios del artículo: más o menos/poco comunes será 0; cualquier otra opción será 1.

Actividad 9. ¿Usa tu hijo/a gestos simples?

Objetivo Analizar si el niño es capaz de comunicarse mediante gestos, si dispone de unas herramientas/nociones básicas de comunicación no verbal.

Procedimiento En primer lugar se hará pasar al sujeto a una sala acompañado por el terapeuta y se hará un contacto previo con el robot NAO. El objetivo de este punto es establecer un clima de confianza en el contexto terapéutico: es conveniente que el terapeuta introduzca al NAO al sujeto. el terapeuta debe situarse lo suficientemente cerca como para observar al sujeto en vivo y controlar aspectos del robot o de la actividad si es necesario.

Son necesarias una silla para el sujeto y una mesa donde se encontrará situado el robot NAO. Es necesario que el robot se encuentre delante del niño, para poder establecer contacto visual. Una vez el sujeto esté sentado, el robot se dirigirá a él, mirándolo y llamándolo por su nombre: *"¡Hola, Sara! Soy NAO. ¿Te gustan los dinosaurios? A mí me gustan mucho, mi favorito es el brontosaurio, porque ¡es muy grande! (...)"*. Al acabar, Nao se despedirá del niño con la mano, haciendo el gesto de despedida. El terapeuta debe observar si el niño se despide de él, vigilando que no sea simple imitación. De nuevo, es necesario que opere según su criterio profesional.

Actividad 10. ¿Suele tu hijo mirar a la nada sin propósito aparente?

Objetivo Analizar la presencia de uno de los rasgos más típicos en las personas con TEA: la mirada perdida sin razón.

Procedimiento De nuevo, debido a la complejidad de creación de un contexto en el que este rasgo pueda medirse rápidamente, es necesario que los responsables proporcionen la información, con las cinco respuestas originales como posibilidad: muchas veces al día, unas pocas veces al día, unas pocas veces a la semana, menos de una vez al mes o nunca. La respuesta dada se binarizará según los criterios del artículo: muchas/unas pocas veces al día/unas pocas veces a la semana será 1; cualquier otra opción será 0.

Conclusiones

En resumen, se han diseñado seis actividades que, junto con la información proporcionada para las otras cuatro, completan el Q-CHAT-10. Se propone por tanto una herramienta de evaluación directa que, en un tiempo muy breve, proporciona una información muy valiosa para el terapeuta a la hora de realizar una primera valoración de la presencia de TEA en el sujeto.

3.3 Clasificador

A la hora de desarrollar desde cero un proyecto de inteligencia artificial hay una serie de pasos que son comunes, o al menos lo suficientemente frecuentes como para formar una lista de tareas ([Géron, 2017](#)):

1. Observar el problema a resolver en perspectiva, sin centrarse en los detalles.
2. Obtener los datos.
3. Explorar los datos para encontrar sus características principales.
4. Preparar y limpiar los datos.
5. Explorar diferentes modelos de aprendizaje automático, escogiendo los mejores para el problema tratado.

6. Realizar el *fine-tuning* de los modelos, combinándolos si es necesario, hasta obtener una solución final.
7. Presentar la solución.
8. Lanzar, monitorizar y mantener el sistema.

En este caso, el problema está ya definido por los apartados previos. La obtención de los datos, sin embargo, supuso desde el principio un escollo a superar: la población objetivo es muy pequeña, y su acceso muy limitado. Desde el primer momento el equipo de psicólogos avisó de que con los medios disponibles sería prácticamente imposible obtener más de 20 o 30 registros, y que posiblemente ni siquiera podría alcanzarse esa cantidad, siendo imposible en cualquier caso saberlo con certeza hasta no disponer del sistema completo y en funcionamiento. Sin embargo, el proyecto no podía estar completamente acabado hasta disponer de los datos; un claro ejemplo de círculo vicioso. En este contexto, solo había una solución: utilizar para el primer análisis un conjunto de datos preexistente.

3.3.1 Conjunto de datos y preprocesamiento

El *Autistic Spectrum Disorder Screening Data for Toddlers* es un conjunto de datos recopilado por Fadi Thabtah de la Universidad de Auckland durante sus investigaciones, hecho público en julio de 2018 ([Thabtah, 2018a](#)). Contiene algo más de 1000 respuestas al Q-CHAT-10, binarizadas siguiendo

el mismo método que los autores del artículo original: en las primeras nueve preguntas las dos primeras opciones puntúan cero, mientras que las tres últimas puntúan uno; en la décima pregunta las tres primeras opciones puntúan cero y las dos últimas puntúan uno. La lista completa de atributos es:

- Respuestas binarizadas al Q-CHAT-10: diez columnas binarias 0/1.
- Edad del sujeto en meses: numérico, entero.
- Puntuación del Q-CHAT-10, siendo ≤ 3 no TEA y > 3 TEA: numérico, entero.
- Género del sujeto: binario 0/1.
- Raza del sujeto: cadena de texto.
- Nacido con ictericia: binario 0/1.
- Familiares con TEA: binario 0/1.
- Quién responde al test (madre, padre, cuidador...): cadena de texto.
- Rasgos autistas: binario 0/1.

Además, la Universidad de California en Irvine cuenta en su repositorio de datos con algo más de cien registros de respuesta al Q-CHAT-10, con los mismos 18 atributos del conjunto de Thabtah y el país de residencia del sujeto (UCI, 2017). Eliminando la columna del país ambos conjuntos pueden unirse, obteniendo cerca de 1200 resultados. Se ha comprobado que las distribuciones

probabilísticas de ambos conjuntos entran dentro de la normalidad, como puede verse en la Figura 3.1. De haberse encontrado alguna anomalía, por ejemplo un porcentaje de más del 90% de síes o noes en alguna respuesta, habría que haber valorado otras alternativas.

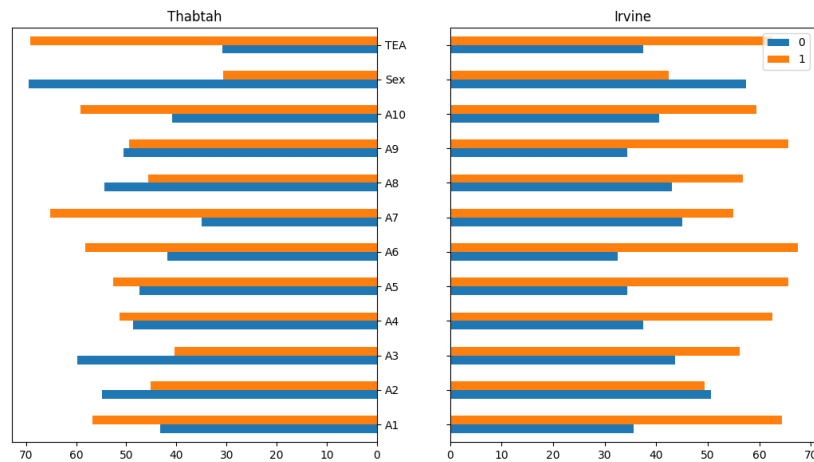


Figura 3.1: Distribuciones de ambos conjuntos de datos, por pregunta.

En el sistema real presentado aquí se recoge la edad y el género, pero no la raza ni los datos relacionados con ictericia, historia médica familiar ni propósitos del test, por lo que esas columnas serán eliminadas antes de aplicarlas al modelo. Además, Thabtah recomienda eliminar la columna de puntuación del Q-CHAT-10, ya que provoca *overfitting*, por lo que esa también será eliminada. Se decidió que los datos recogidos, fuese la cantidad que fuese, se añadirían al conjunto y se trabajaría con todos ellos. El conjunto no incluye datos nulos, por lo que solo hay que preocuparse por los que se

incorporen a partir de la extracción, que serán completados con la media de la columna.

Con esto, la lista estaría completa hasta el punto 4; a continuación, hay que elegir el modelo.

3.3.2 Modelo de aprendizaje automático

El problema es claramente de clasificación binaria supervisada: dados los datos, determinar si un sujeto pertenece a la clase TEA o a la clase no-TEA. Thabtah utiliza dos modelos, logistic regression y naive Bayes. Dado que el conjunto de datos utilizado será idéntico en un altísimo porcentaje, se propone la utilización de otros algoritmos con el fin de no solo evaluar el sistema en conjunto, sino también realizar un análisis comparativo aproximado con respecto a los resultados de Thabtah. Los modelos escogidos son:

- Support Vector Machine (SVM)
- Decision tree
- Random forests
- Nearest neighbor (KNN)
- Neural network (en concreto, un perceptrón multicapa o MLP)
- Logistic regression

Con esto, la lista estaría completa hasta el punto 6; solo resta presentar la solución y escoger el modo de lanzamiento del sistema.

3.3.3 Ingeniería de datos: lanzamiento y mantenimiento del sistema

Una vez seleccionado el mejor modelo, hay que entrar en la gestión de los datos. El entrenamiento inicial se realizará de forma *offline* con los datos disponibles previamente. Una vez finalizado el entrenamiento, el modelo se almacenará y se utilizará para predecir los resultados de los usuarios a medida que se recojan sus respuestas a las actividades. El ecosistema diseñado es bastante limpio, con datos binarizados y sin ruido; el preprocesado posterior a la extracción se encarga de que el flujo ETL (del inglés *Extraction, Transformation and Load*) sea rápido, directo y laminar.

Con esto, la lista de tareas estaría completa: el diseño del sistema está terminado, y es momento de desarrollarlo.

Capítulo 4

Desarrollo

Este capítulo describe la implementación técnica de la solución escogida. El servicio web será mostrado en detalle, incluyendo tanto el registro y la interfaz de usuario como la comunicación con el robot y la gestión de datos. Más brevemente, se comentarán las especificaciones de las actividades y del modelo clasificador.

4.1 Infraestructura

Una vez definida la infraestructura en forma de servicio web, el análisis continúa: ¿qué tecnología es la más adecuada? ¿Cómo seleccionar los diferentes módulos de forma que la integración entre todos sea óptima? Hay una serie de limitaciones a considerar:

- Robot NAO: el fabricante proporciona el SDK en varios lenguajes ([Soft-Bank Robotics, 2019b](#)): C++, Python, .NET, Java y Matlab. De estos, los dos con mayor comunidad detrás son C++ y Python.
- Sistema clasificador: la mayor parte de las librerías utilizadas para machine learning (numpy, scikit-learn) están escritas en Python.
- Limitaciones económicas: idealmente, el proyecto debería realizarse enteramente con herramientas Open Source, para minimizar costes.

El estudio de estas limitaciones arroja un ganador claro: Python. La opción de utilizar múltiples lenguajes e integrarlos mediante la creación de una API y *wrappers* se descartó de inmediato, ya que añadía dificultades al proyecto sin aportar nada de valor. El sistema, por tanto, quedó estructurado como se ve en la Figura 4.1.

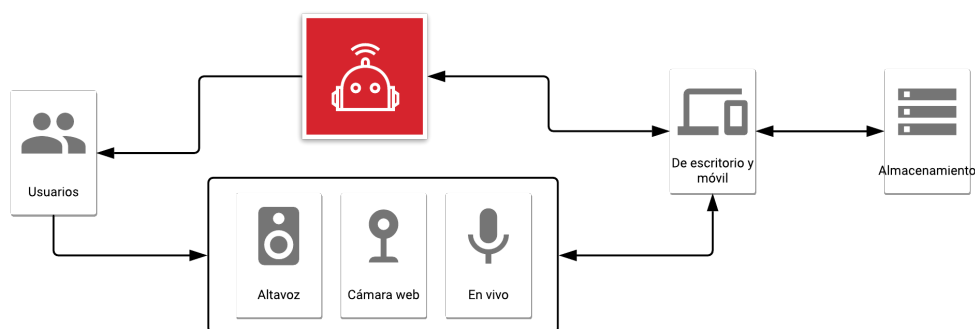


Figura 4.1: Diagrama del sistema.

El operario o terapeuta entrará al sistema desde un punto de acceso, ya

sea un ordenador portátil o dispositivo móvil. Ese punto de acceso estará conectado a una cámara web, un micrófono¹ y un altavoz, disponibles de forma nativa en la gran mayoría de equipos, y al robot Nao, mediante una conexión inalámbrica. A su vez, estará conectado con la base de datos centralizada donde se registrarán los resultados de las actividades. El dispositivo de acceso será el núcleo del sistema; el robot se limitará a actuar en función de las órdenes recibidas, y los dispositivos de entrada y salida recogerán los datos de forma automática en función de la actividad seleccionada. La interacción del robot y el usuario, por tanto, no será realmente bidireccional; el robot hablará, pero no recogerá la respuesta: será el resto de dispositivos quién se encargue de ello. Asimismo, toda la toma de decisiones se llevará a cabo en el servidor donde se encuentre alojado el sistema, una vez recogidos los datos; en base a ello, se enviará una señal al robot. De este modo, el sistema imitará el comportamiento bidireccional.

A la pregunta de por qué no se ha implementado una bidireccionalidad real, la respuesta es simple: exigiría que la extracción de datos y la toma de decisiones se llevara a cabo en el robot, pero los sensores integrados son de peor calidad y la potencia computacional es menor que la de un equipo externo. Por ello, se ha simulado; no hay diferencias a efectos prácticos, dado que no existe diferencia a ojos del usuario, y la calidad de los datos extraídos mejora sensiblemente.

¹En las actividades implementadas no se usan la cámara web ni el micrófono; se ha implementado la conectividad con vistas a futuras extensiones del sistema.

4.1.1 Sistema operativo, framework web e IDE

El sistema operativo escogido ha sido Linux (Ubuntu 16.04) debido a que es Open Source y por tanto, no exige el pago de licencias. Además, algunas de las herramientas necesarias están pensadas especialmente para ser usadas por línea de comandos, aspecto en el que Linux destaca.

Un framework es "una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación (...) se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta" ([Gutiérrez, 2005](#)). Un framework web, por tanto, es un conjunto de elementos que forman una estructura reutilizable que facilita el desarrollo de aplicaciones y sistemas web. Por supuesto, no es necesario un framework para desarrollar en un entorno web, siempre se puede partir de cero; sin embargo, reinventar la rueda no se encuentra entre los objetivos de este trabajo.

Una vez escogido el lenguaje, los web frameworks más convenientes por conocimientos previos disponibles y comunidad eran dos: Django ([Django Software Foundation, 2019](#)) o Flask ([Ronacher, 2019](#)). De entre ambos, se escogió Django porque permite diseñar y desarrollar proyectos más complejos que Flask. Si bien es cierto que para la magnitud de este TFM habría bastado con Flask, y posiblemente habría resultado más sencillo, no hay que olvidar que el trabajo forma parte de un proyecto mayor; las decisiones técnicas se han tomado siempre con vistas a una extensión futura. Además, Django es

más usado que Flask ([Arsenault, 2018](#); [Goel, 2018](#)) y la seguridad de tener una comunidad grande detrás ayuda a la supervivencia de los proyectos.

La estructura del proyecto, acorde a la filosofía de Django, consta de dos proyectos. El principal, *autismws*, define las urls desde las que acceder, la página principal, el registro de usuarios y los perfiles y en general todos aquellos componentes de uso común que pueden considerarse transversales. El segundo, *activities*, define el funcionamiento de las actividades. Las interfaces, programadas en HTML y JavaScript, se almacenan en una carpeta aparte llamada *templates*; existe también una carpeta anexa para las pruebas funcionales. La explicación detallada de las buenas prácticas y el funcionamiento interno de Django queda fuera del ámbito de este proyecto y no se detallará aquí; sin embargo, la estructura completa del sistema, así como el código, puede consultarse en el repositorio del apéndice [A](#).

En cuanto al entorno de desarrollo o IDE (del inglés *Integrated Development Environment*), PyCharm es el recomendado casi unánimemente ([Fincher, 2019](#); [Walker, 2019](#)). De nuevo, se puede usar un editor de texto como Sublime en Linux o Notepad++ en Windows, pero la pregunta es obvia: ¿para qué? El uso de una herramienta más potente y específica, como es PyCharm, facilita el trabajo de desarrollo.

4.1.2 Buenas prácticas en el desarrollo de software

Al desarrollar un producto de software hay elementos de planificación y control, ajenos al código, que son necesarios para dar solidez. Una lista no exhaustiva incluye:

- Control de versiones
- Pruebas unitarias
- Pruebas funcionales
- Modularidad
- Documentación del código
- Integración continua
- Entrega continua
- Logging

De nuevo, teniendo en cuenta que este proyecto se centra en la inteligencia artificial y no en la ingeniería de software, no se detallarán los pormenores de todos estos conceptos; simplemente se busca indicar que se han tenido en cuenta en el diseño. El control de versiones, indispensable para conservar la historia del proyecto y poder revertir cambios cuando existen varios colaboradores, ha sido Git; en concreto, se ha utilizado GitHub. Las pruebas

unitarias y funcionales se han llevado a cabo con los paquetes nativos de Python `unittest` y `selenium`, y se han automatizado gracias a las facilidades que ofrece PyCharm de modo que basta con pulsar un botón para ejecutar toda la batería de tests. También el logging se ha implementado con el sistema nativo de Django. No se ha implementado entrega ni integración continua, ya que la amplitud actual del proyecto no lo requería, pero si se ha desarrollado pensando en una inclusión futura: por ejemplo, se podría integrar Jenkins con GitHub y los tests funcionales de forma que un *commit* lanzara automáticamente los tests, mostrando al desarrollador si sus cambios son compatibles con lo anterior. En cuanto a documentación, se han utilizado los docstrings de Python y comentarios en código, y todo el proyecto se ha diseñado buscando la máxima modularidad, facilitando la extensión del sistema añadiendo nuevas actividades o archivando las ya no válidas.

Sobre las buenas prácticas en código, se han intentado respetar en todo momento los cinco principios que forman SOLID (*Single responsibility, Open-closed, Liskov substitution, Interface segregation and Dependency inversion*), manteniendo los métodos simples y concisos, con nombres claros y descriptivos, facilitando además las pruebas unitarias ([Martin, 1995, 1996a,b,c, 2009](#)).

4.1.3 Registro de usuarios

El registro se ha implementado usando `django-registration-redux` ([Bennett et al., 2019](#)), un módulo de Django que proporciona una estructura básica

de registro que puede modificarse luego para incorporar las características propias de cada proyecto. A las opciones básicas (usuario, dirección de correo electrónico y contraseña) se ha añadido una extra, el grupo, como puede verse en la Figura 4.2. Se han incluido dos opciones de grupo, paciente y terapeuta, de modo que el terapeuta puede registrarse en el sistema y dar de alta a sus pacientes. Internamente existe un tercer grupo, administrador, con vistas a un futuro soporte informático externo.

Password*

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation*

Enter the same password as before, for verification.

Group*

Submit

Figura 4.2: Pantalla de registro de nuevo usuario (detalle).

Para mayor flexibilidad, se ha optado por crear una página de perfil de usuario donde pueden modificarse los datos de registro generales (ver Figura 4.4) y la información de las cuatro preguntas del Q-CHAT-10 que no se

han implementado como actividades (ver Figura 4.5). La jerarquía es la de la Figura 4.3: un administrador puede editar cualquier perfil; un terapeuta puede editar su propio perfil y el de los sujetos sobre los que se realizarán las actividades; y por último, un usuario llano² únicamente puede editar su propio perfil. Se ha optado por no forzar la asociación entre un terapeuta y sus sujetos (es decir, que un sujeto solo pueda ser modificado por el terapeuta que tenga asignado) debido a que la especificación de requisitos es difusa en ese punto: es posible que existan sujetos pertenecientes a más de un terapeuta.

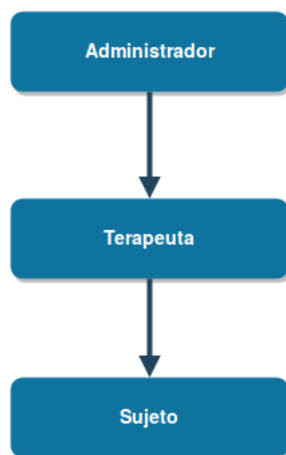


Figura 4.3: Jerarquía de los grupos de usuarios.

²Internamente los usuarios llanos son llamados pacientes. Sin embargo, el equipo de psicólogos se ha opuesto a dicha denominación, ya que posee una connotación subjetiva de *enfermo*, y el trastorno del espectro autista no es una enfermedad. Por lo tanto, se ha optado por modificar la nomenclatura en la memoria.

UNED AI Users Profile Log out

Perfil personal

Nombre

Apellidos

Email

Biografía

Datos relevantes del paciente.

Observaciones del terapeuta.

Figura 4.4: Datos generales de la pantalla de perfil de usuario.

Género*

Edad (en meses)*

¿Finge tu hijo/a al jugar?*

Seleccionar si el niño/a es capaz de fingir mediante el juego, por ejemplo cuidar de muñecas como si fueran niños, hablar por un teléfono móvil de mentira...

¿Sigue tu hijo/a la dirección de tu mirada?*

Seleccionar si el niño/a suele seguir la dirección de la mirada de la persona que está con él/ella.

Describirías sus primeras palabras como...*

Indicar la habilidad.

¿Suele tu hijo mirar a la nada sin propósito aparente?*

Indicar si se trata de un comportamiento frecuente.

¿Presenta el paciente signos de TEA?

Figura 4.5: Datos de las cuatro preguntas del Q-CHAT-10 no implementadas en el perfil del usuario.

4.1.4 Interfaz de usuario. Comunicación

Cuando el usuario se registra o accede al sistema se muestra la pantalla principal con todas las actividades, que puede verse en la Figura 4.6. En la cabecera de la página se encuentra el acceso al perfil, así como un botón para salir. Cada una de las actividades tiene un botón que muestra los detalles en un *pop-up* como el de la Figura 4.7, sin abandonar la página principal, y otro para iniciarla, que redirige a la pantalla de la actividad. Una vez allí, hay dos posibles resultados: actividad finalizada con éxito, o bien un error o interrupción de la misma. Ambas opciones redirigen a una pantalla de éxito/error desde la cual se vuelve a la pantalla principal. El flujo del programa se muestra en la Figura 4.8.

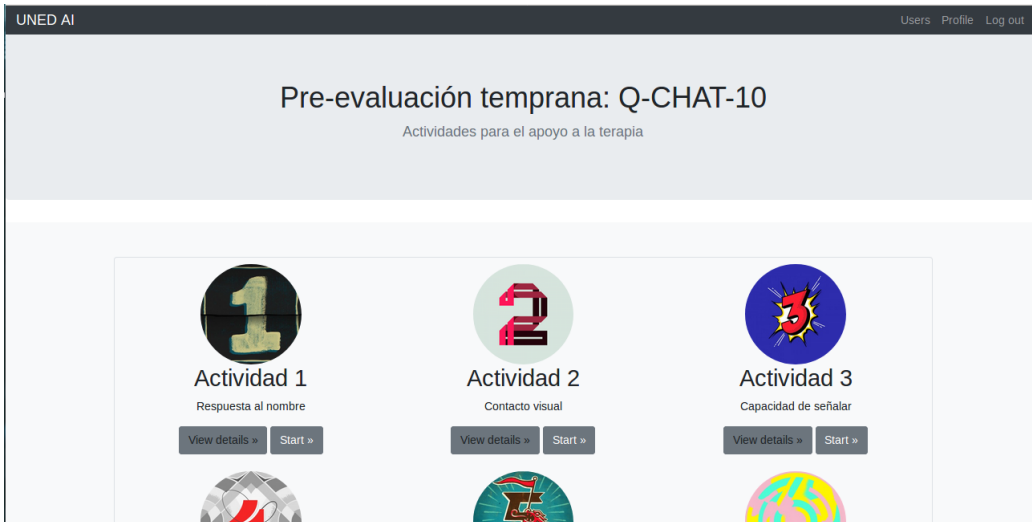


Figura 4.6: Menú de actividades.

Se ha diseñado una interfaz de usuario mínima, para evitar errores no

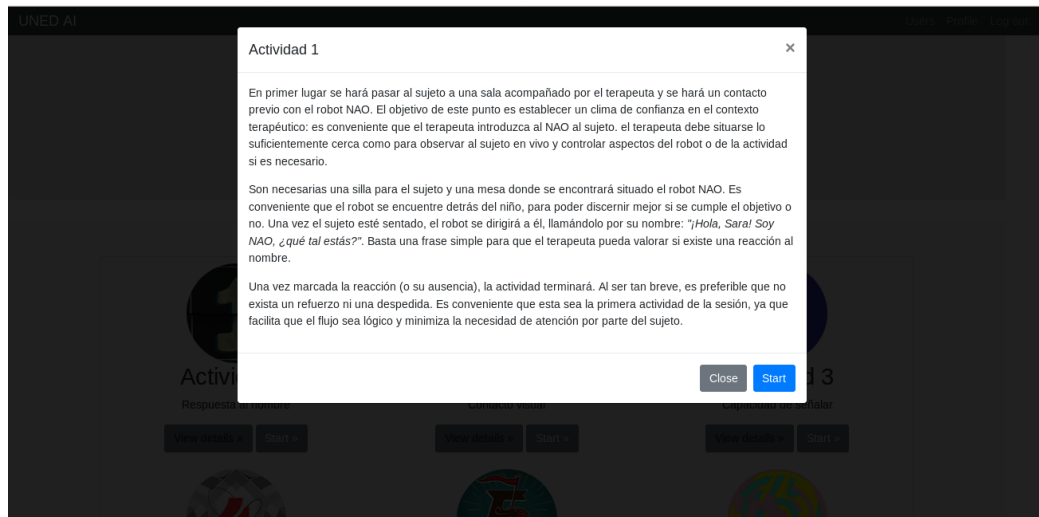


Figura 4.7: *Pop-up* con la descripción de la actividad.

deseados por parte del operario o terapeuta que maneje el sistema, consistente en tres bloques significativos:

- Selección del sujeto o paciente
- Botones para respuestas binarias: sí/no
- Botón de parada de emergencia
- Texto de apoyo

Cada una de las actividades es diferente, pero se ha conseguido que la interfaz sea válida para todas ellas; de este modo, el terapeuta no necesita familiarizarse con un entorno complejo. Aún así, la modificación es sencilla si en algún momento futuro hay que añadir o eliminar algún componente en

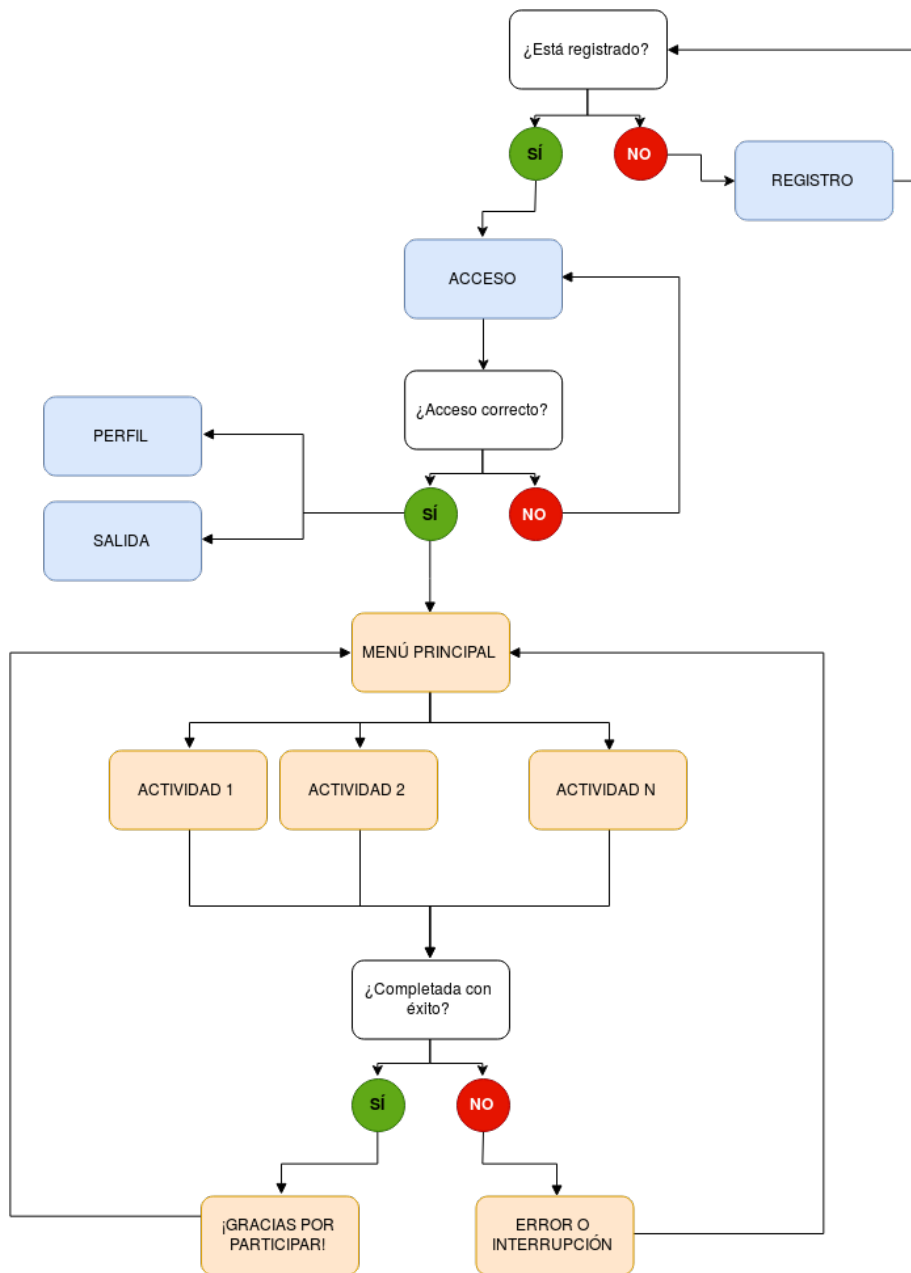


Figura 4.8: Flujo de ejecución del sistema.

nuevas actividades. En la Figura 4.9 se puede ver la interfaz estándar: un selector de usuarios llanos con botón de confirmación; dos botones binarios para respuestas de sí o no y un botón de parada de emergencia con un cuadro de texto para indicar, obligatoriamente, la razón de la parada. En la Figura 4.10 se ve la interfaz una vez seleccionado el usuario, con el texto de apoyo, que subtitula lo que Nao dice, debajo de los botones para respuestas binarias. En la Figura 4.11 se puede ver el aviso que salta si el terapeuta intenta detener la actividad sin escribir un motivo.



Figura 4.9: Interfaz de actividad con los componentes estándar.

Actividad 01

¿Te mira tu hijo/a cuando le llamas por su nombre?

The interface consists of three main panels:

- Patient Panel:** A dropdown menu with 'pat1' selected. Below it, the text reads: 'Username: pat1', 'Full name: David Beckham', and 'Joined at: June 28, 2018, 9:40 p.m.'. A blue 'Select' button is at the bottom.
- Yes - No Panel:** Two buttons, 'Yes' (green) and 'No' (red). Below them is the text: '¡Hola, David! Soy NAO. ¿qué tal estás?'.
- Stop activity - Danger! Panel:** A text input field and a red 'Stop' button.

Figura 4.10: Interfaz de actividad con el usuario seleccionado.

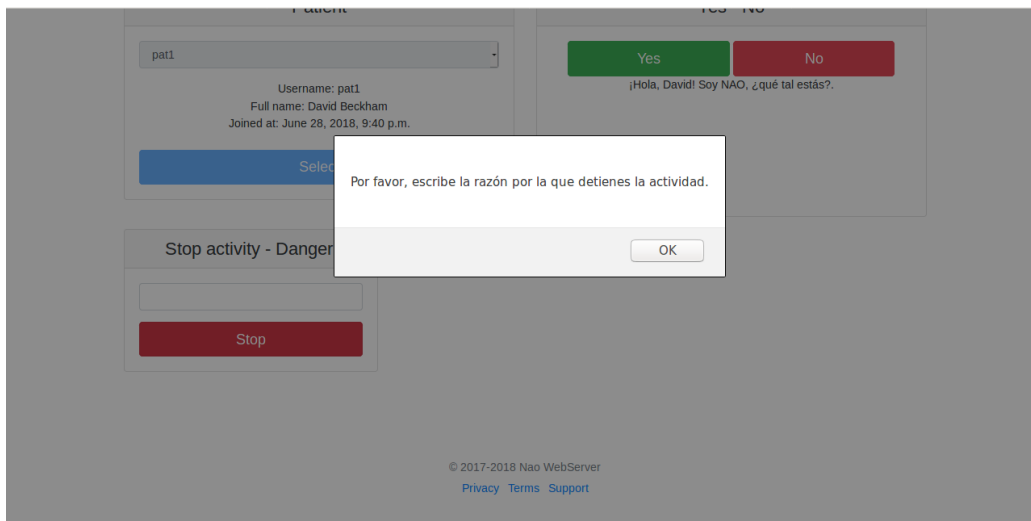


Figura 4.11: Aviso de error: la actividad no puede detenerse sin un motivo.

4.1.5 Comunicación con el robot

A lo hora de implementar la comunicación con el robot había únicamente dos opciones:

- Cargar previamente en el robot uno o varios programas, según las necesidades de cada actividad, y ordenar su ejecución desde el sistema.
- Enviar desde el sistema las órdenes directas: muévete; habla; siéntate.

La primera opción era más sencilla: la orden de ejecución es fácil de enviar de forma inalámbrica, y el sistema solo tendría que esperar un tiempo determinado para que el programa terminara su ejecución. Sin embargo, un análisis reveló inmediatamente que el plan tenía fisuras: si el robot iba a ser simplemente un actuador, el programa no podía basarse en las reacciones del sujeto; se desarrollaría, de principio a fin, independientemente de lo que pasara a su alrededor. Claro, es posible hacer que el programa reaccione ante el entorno, pero en ese caso ¿por qué no mover toda la toma de decisiones al robot? Ese debate estaba ya cerrado, y el centro neurálgico del sistema tenía que ser el entorno web. Diseñar los programas para que interactuasen con su entorno manteniendo la toma de decisiones en la web añadía demasiadas complicaciones. Por tanto, se decidió que las órdenes, simples y cortas, se enviasen desde el sistema. Con este fin se implementaron cuatro *scripts*:

- Script de inicio: hace que el robot se incorpore y permanezca en posición de espera.

- Script de conversación: hace que el robot pronuncie una o varias frases, con posibles tiempos de espera entre ellas, con o sin movimientos incorporados simultáneamente.
- Script de movimiento: hace que el robot cambie la pose a alguna de las predeterminadas. Tras un estudio de las actividades, se decidió que no era necesario crear más poses específicas.
- Script de fin: hace que el robot se sienta y permanezca en posición de ahorro de batería.

Al iniciar cualquiera de las actividades se lanza, automáticamente, el script de inicio. Al finalizarla, ya sea correctamente o con algún error, se lanza el script de fin. A lo largo de las actividades, en momentos concretos según las respuestas y la velocidad del sujeto, se ejecutan los scripts de conversación o movimiento, con parámetros definidos en cada caso. De este modo, toda la comunicación con el robot es más flexible, y en futuras actividades basta con definir los parámetros necesarios (pose y frase) para poder incorporar al robot, logrando de este modo una modularidad mucho mayor y una menor dependencia del robot, que ya no necesita tener programas precargados.

Cabe destacar que el SDK del robot Nao utilizado solo daba soporte para Python 2.7, mientras que el resto del sistema se ha realizado con Python 3.5. Para solucionar esto sin tener que realizar un *downgrade* a Python 2.7 de todo el sistema, se han instalado ambas versiones en el servidor y cada vez

que se recibe la orden de ejecutar un script, se envía el comando al intérprete de Python 2.7. De este modo se ha intentado respetar el principio SOLID de no depender, en lo posible, de implementaciones. El fragmento de código que se ocupa de ejecutar el script de conversación puede verse en el fragmento 4.1, mientras que el script en sí puede verse en el fragmento 4.2.

```
1 def nao_speech(request):
2     if not request.user.is_authenticated:
3         return HttpResponseRedirect("/login/")
4
5     if request.method == 'GET' and request.GET.get('text') is
6         not None:
7         script = "content/scripts/nao_speech.py"
8         text = request.GET.get('text')
9         animation = request.GET.get('animation')
10        if animation != 'none':
11            script = "content/scripts/nao_animated_speech.py"
12
13        process = subprocess.Popen([settings.PYTHON_PATH, script
14            , text], stdout=subprocess.PIPE)
15        output, error = process.communicate()
16
17        output_lines = output.decode('UTF-8').splitlines()
18        for l in output_lines:
19            logger.info(l)
20
21        data = {
22            'output_lines': "Recibido!"
23        }
```

```

23
24     else:
25         return HttpResponseRedirect("/")

```

Listing 4.1: Ejecución del script de conversación

```

1 from django.conf import settings
2
3 import sys
4 sys.path.append("/opt/pynaoqi-python2.7-2.1.2.17-linux64")
5 from naoqi import ALProxy
6
7 animated_speech = ALProxy("ALAnimatedSpeech", settings.NAOIP,
8     settings.NAOPORT)
9
10 animated_speech.say(str(sys.argv[1]), {"bodyLanguageMode": "
11     contextual"})

```

Listing 4.2: Script de conversación

4.2 Actividades

El concepto de gamificación o ludificación consiste en aplicar componentes lúdicos o de juego en contextos ajenos al juego (Deterding et al., 2011; Hamari et al., 2014; Kapp, 2012). Aunque es particularmente importante en la enseñanza, hay estudios que apuntan a una posible mejora en los tratamientos de personas autistas cuando se aplican técnicas gamificadoras (Ern, 2014; Nacke and Deterding, 2017). Con vistas a esto, se han diseñado las activida-

des de forma que sean interpretables como un juego, aunque realmente no lo sean.

La implementación de las actividades ha sido un proceso iterativo, realizada en paralelo con el diseño, intentando ceñir el trabajo al paradigma de las metodologías ágiles. Tras recibir un primer borrador del diseño inicial, se estudiaba cómo adaptarlo a la interfaz respetando en todo momento los requisitos, y se devolvía al equipo de psicólogos con los problemas que más probablemente iban a aparecer en la implementación; este proceso habitualmente duraba unas horas. En una breve reunión se discutían los problemas y se decidía si, a pesar del aumento en la complejidad, eran parámetros imprescindibles, o si por el contrario podían sustituirse o eliminarse. En la siguiente iteración, el diseño se modificaba en base a los cambios aceptados, y así sucesivamente; siguiendo este proceso, todas las actividades se resolvieron en un pequeño número de ciclos. Cabe destacar que, en todo momento, el equipo de psicólogos podía hacer sugerencias técnicas y viceversa; no había restricciones firmes. En general, con la infraestructura y la interfaz definidas y los scripts de comunicación con el robot implementados, las actividades no han exigido un gran esfuerzo técnico.

La descripción de las actividades realizada en el capítulo 3 de diseño hace innecesario extenderse más en este apartado; las seis se han realizado con apoyo de los scripts de movimiento y conversación, utilizando los gestos precargados de saludo, incorporación y despedida de la suite de Nao. Cada una de ellas corresponde con una dirección web diferente y tiene asociada

una tabla en la base de datos que almacena toda la información necesaria para el posterior procesamiento de los datos.

4.2.1 Estructura de la base de datos

La base de datos del sistema se ha diseñado de forma modular, buscando aislar al máximo cada una de las actividades para que conceptualmente puedan considerarse módulos independientes; de este modo, es sencillo sustituir, añadir o eliminar actividades del sistema, así como combinarlas para realizar estudios parciales. Además de las tablas propias de Django (migrations, session, auth_user, auth_group...) se ha creado una tabla para cada actividad.

4.3 Modelo clasificador

4.3.1 Modelado y entrenamiento

Los modelos se han implementado utilizando la librería scikit-learn. Como en cualquier proyecto de machine learning, lo primero ha sido limpiar los datos: muchas de las columnas del conjunto no resultaban de interés, y se han eliminado. También se cambiaron los nombres de algunas y se mapearon otras, para categorizar de forma binaria; por ejemplo, "Yes" se convirtió en 1, y "No" se convirtió en 0. Con ello, se pasó de unos datos con 18 atributos mezcla de categóricos, binarios y enteros:

Original data:

	Case_No	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age_Mons \
0	1	0	0	0	0	0	0	1	1	0	1	28
1	2	1	1	0	0	0	1	1	0	0	0	36
2	3	1	0	0	0	0	0	1	1	0	1	36
3	4	1	1	1	1	1	1	1	1	1	1	24
4	5	1	1	0	1	1	1	1	1	1	1	20

	Score	Sex	Ethnicity	Jaundice	Family_w_ASD \
0	3	f	middle eastern	yes	no
1	4	m	White European	yes	no
2	4	m	middle eastern	yes	no
3	10	m	Hispanic	no	no
4	9	f	White European	no	yes

	Who completed the test	Class/ASD Traits
0	family member	No
1	family member	Yes
2	family member	Yes
3	family member	Yes
4	family member	Yes

A unos mucho más manejables 12 atributos con once columnas binarias y una numérica:

Updated data:

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	Age	Sex	TEA
0	0	0	0	0	0	0	1	1	0	1	28	1	0
1	1	1	0	0	0	1	1	0	0	0	36	0	1
2	1	0	0	0	0	0	1	1	0	1	36	0	1
3	1	1	1	1	1	1	1	1	1	1	24	0	1
4	1	1	0	1	1	1	1	1	1	1	20	1	1

Una vez limpio, el conjunto de datos podía usarse directamente para alimentar un modelo, seleccionando la columna TEA como objetivo de la predicción y dividiéndolo en conjunto de test y validación en un 80% – 20%. Para la optimización de hiperparámetros se ha utilizado el *GridSearchCV* de scikit-learn, que analiza el modelo con diferentes parámetros y escoge el que mejor resultados produce usando cross-validation en el conjunto de test. El principal problema de esta técnica frente a otras como *RandomizedGridSearchCV* es que a mayor número de combinaciones, más tiempo necesita; la ventaja es que garantiza la combinación óptima. La selección de parámetros a valorar y su rango se ha realizado en función de las recomendaciones empíricas de la comunidad de Kaggle ([Howard](#)). Los parámetros analizados para cada uno de los modelos son:

Support Vector Machine (SVM)

- C: 1, 11, 21, 31, 41, 51, 61, 71, 81, 91, 101.
- gamma: 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.

- kernel: *rbf*.
- class_weight: *balanced*, None.

Los parámetros óptimos son:

```
*****
Training model: SVM
*****
Fitting 3 folds for each of 180 candidates, totalling 540 fits
Done 540 out of 540 | elapsed: 15.4s finished
Model with rank: 1
Parameters: {'C': 21, 'class_weight': 'balanced', 'gamma': 0.1,
            'kernel': 'rbf'}

SVM accuracy: 0.9655
SVM precision: 0.9936
SVM recall: 0.9571
```

Decision tree

- max_depth: 3, 5, None.
- max_features: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
- min_samples_leaf: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10.
- criterion: *gini*, *entropy*.

Los parámetros óptimos son:

```
*****
Training model: Decision tree
*****
Fitting 3 folds for each of 600 candidates, totalling 1800 fits
Done 1800 out of 1800 | elapsed:    4.2s finished
Model with rank: 1
Parameters: {'criterion': 'entropy', 'max_depth': None,
            'max_features': 3, 'min_samples_leaf': 1}

Decision Tree accuracy: 0.9009
Decision Tree precision: 0.9487
Decision Tree recall: 0.9080
```

Random forest

- `n_estimators`: 400, 800, 1000.
- `max_features`: *auto*, *sqrt*.
- `max_depth`: 40, 60, 80, 100, None.
- `min_samples_split`: 2, 5, 10.
- `min_samples_leaf`: 1, 2, 4.

Los parámetros óptimos son:

Training model: Random forest

Fitting 3 folds for each of 270 candidates, totalling 810 fits

Done 810 out of 810 | elapsed: 11.5min finished

Model with rank: 1

Parameters: {'max_depth': 40, 'max_features': 'auto',
 'min_samples_leaf': 1, 'min_samples_split': 2,
 'n_estimators': 400}

Random Forest accuracy: 0.9612

Random Forest precision: 0.9753

Random Forest recall: 0.9693

KNN

- n_neighbors: 5, 7, 10.
- leaf_size: 1, 3, 5.
- weights: *uniform, distance*.
- algorithm: *auto, kd_tree*.

Los parámetros óptimos son:

Training model: KNN

Fitting 3 folds for each of 36 candidates, totalling 108 fits

Done 108 out of 108 | elapsed: 4.7s finished

Model with rank: 1

Parameters: {'algorithm': 'auto', 'leaf_size': 3,
 'n_neighbors': 7, 'weights': 'uniform'}

K Nearest Neighbors accuracy: 0.9483

K Nearest Neighbors precision: 0.9748

K Nearest Neighbors recall: 0.9509

Neural network

- activation: *tanh*, *relu*.
- alpha: 0.0001, 0.001, 0.01.
- hidden_layer_sizes: (50,50,50), (50,100,50), (100,).
- learning_rate: *constant*, *adaptive*.
- solver: *sgd*, *adam*.

Los parámetros óptimos son:

Training model: Neural network

Fitting 3 folds for each of 72 candidates, totalling 216 fits

Done 216 out of 216 | elapsed: 2.0min finished

Model with rank: 1

Parameters: {'activation': 'tanh', 'alpha': 0.0001,
 'hidden_layer_sizes': (50, 50, 50),
 'learning_rate': 'constant', 'solver': 'adam'}

Neural Network accuracy: 0.9741

Neural Network precision: 1.0000

Neural Network recall: 0.9632

Logistic regression

- penalty: $l1$, $l2$.
- tol = 0.0001, 0.001, 0.01.
- C: 0.5, 1.
- solver: *liblinear*, *saga*.

Los parámetros óptimos son:

Training model: Logistic Regressor

```
*****
Fitting 3 folds for each of 24 candidates, totalling 72 fits
Done 216 out of 216 | elapsed: 2.0min finished
Model with rank: 1
Parameters: {'C': 1, 'penalty': 'l1', 'solver': 'liblinear', 'tol': 0.01}

Logistic regressor accuracy: 0.9526
Logistic regressor precision: 0.9634
Logistic regressor recall: 0.9693
```

Los resultados se analizarán en el capítulo 5; el modelo escogido se ha almacenado con el módulo *pickle*, incorporándolo al sistema:

```
1 from sklearn.externals import joblib
2
3 joblib.dump(chosen_model, 'model.pkl')
```

Listing 4.3: Almacenamiento del modelo

4.3.2 Preprocesado de los datos

Una vez entrenado el modelo puede hacerse el paso a producción y comenzar a predecir según las respuestas de los usuarios del sistema. Sin embargo, antes de alimentarlo con los datos extraídos es preciso realizar un preprocesado ligero, para que el formato de los datos sea el mismo que al entrenar. Por la implementación realizada se dispone de 6 respuestas binarias (1, 2, 3, 4,

7 y 9 del Q-CHAT-10 original), los datos de edad y género y la información previa referida a las cuatro preguntas que no se han convertido en actividades (5, 6, 8 y 10 del Q-CHAT-10 original); la única información a preprocesar es la de estas cuatro últimas preguntas. Las respuestas se han binarizado de acuerdo a los criterios del artículo original (Allison et al., 2012; Thabtah, 2018a): en 5, 6 y 8 las dos primeras opciones puntúan cero, mientras que las tres últimas puntúan uno; en la 10 las tres primeras opciones puntúan cero y las dos últimas puntúan uno.

4.3.3 Resultados del usuario

Si se dispone de permisos de terapeuta o administrador, desde la lista de usuarios (Figura 4.12) se puede acceder al perfil de los usuarios. En él, además de los datos generales ya detallados en el apartado 4.1.3, se muestran los resultados obtenidos en cada una de las pruebas, siendo *True* si se han observado signos típicos de TEA y *False* en caso contrario, como se ve en la Figura 4.13. Si alguna actividad no se ha completado, se mostrará el signo $-$; en ese caso, se indicará que *No hay datos suficientes para valorar a este usuario*. Por el contrario, si todas las actividades se han completado, se enviarán los datos al modelo para que pueda realizar la inferencia:

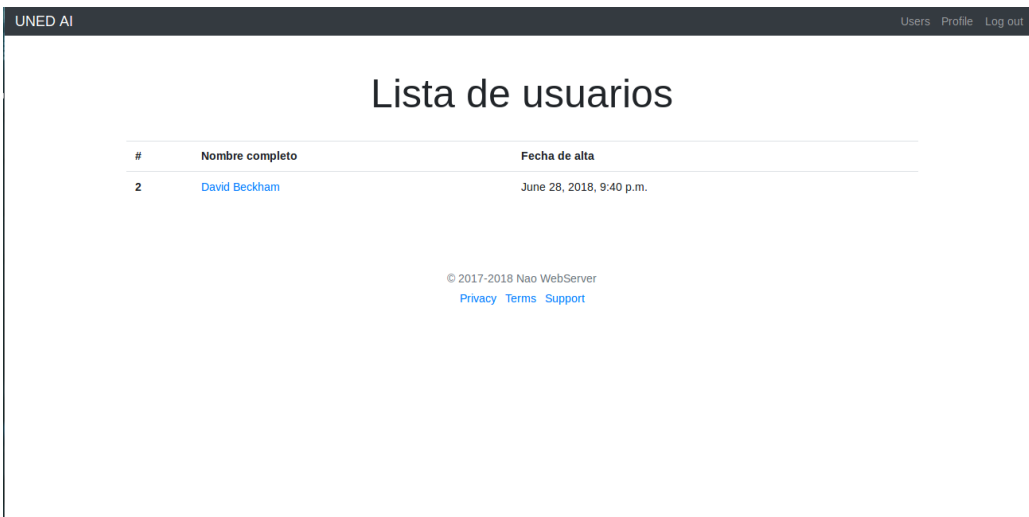


Figura 4.12: Lista de usuarios.



Figura 4.13: Resultados finales del usuario.

```

1 def get_user_prediction(pk) :
2
3     user = User.objects.get(pk=pk)

```

```

4     user_profile = UserProfile.objects.get(user_id=user.id)
5
6     a01 = get_answer(Activity01.objects.filter(patient_id=user.
7         id).first())
8     a02 = get_answer(Activity02.objects.filter(patient_id=user.
9         id).first())
10    a03 = get_answer(Activity03.objects.filter(patient_id=user.
11        id).first())
12    a04 = get_answer(Activity04.objects.filter(patient_id=user.
13        id).first())
14    a05 = get_answer(Activity05.objects.filter(patient_id=user.
15        id).first())
16    a06 = get_answer(Activity06.objects.filter(patient_id=user.
17        id).first())
18
19    ai01 = binarize_info(1, user_profile.q_ai_1)
20    ai02 = binarize_info(2, user_profile.q_ai_2)
21    ai03 = binarize_info(3, user_profile.q_ai_3)
22    ai04 = binarize_info(4, user_profile.q_ai_4)
23
24    age = user_profile.age
25    gender = 1 if user_profile.gender == 'F' else 0
26
27    if '-' in answers.values():
28        answers["RESULTADO FINAL"] = "No hay datos suficientes
29            para valorar a este usuario."
30
31    else:
32        user_data = [[a01, a02, a03, a04, ai01, ai02, a05, ai03,
33            a06, ai04, age, gender]]
34        answers["RESULTADO FINAL"] = predict_user(user_data)
35
36    return answers

```

```

26
27
28 def predict_user(user_data):
29
30     print(user_data)
31     model = joblib.load('content/models/model.pkl')
32     df = pd.DataFrame(user_data, columns=['A1', 'A2', 'A3', 'A4'
33         , 'A5', 'A6', 'A7', 'A8', 'A9', 'A10', 'Age', 'Sex'])
34
35     if model.predict(df)[0] == 0:
36         return 'El usuario no parece tener rasgos de TEA.'
37     else:
38         return 'El usuario presenta ciertos rasgos
39         característicos de TEA. Se sugiere realizar un
40         análisis en profundidad.'

```

Listing 4.4: Inferencia a partir de los datos del usuario.

4.3.4 Integración de los datos extraídos: actualización del modelo

La clasificación del usuario en TEA o no-TEA se realiza de forma automática, siendo visible en el perfil de usuario, bajo el título "¿Presenta el paciente signos de TEA?", como puede verse en la Figura 4.13. Sin embargo, el campo es modificable por el terapeuta: es posible que el terapeuta sepa que la clasificación ha sido incorrecta, o que considere que pese al resultado hay indicios suficientes de que existe una duda razonable, y en ese caso debe

poder corregirla; el conocimiento experto está por delante de la clasificación automática.

Para que los datos recogidos se incorporen al modelo se ha implementado un *cron job* mediante la librería APSchedule, que cada día a las 23.59 ejecuta el siguiente trabajo en el servidor donde esté alojado el sistema:

1. Extrae los datos de todos los sujetos pertenecientes al grupo pacientes.
2. Comprueba si el campo binario oculto *Añadido a la BBDD* es verdadero; si lo es, significa que los datos de dicho usuario ya han sido incorporados a la base de datos.
3. Si no ha sido añadido, comprueba si los datos están completos; podría ser que un sujeto no haya completado todas las actividades en un solo día, y en ese caso no habría que incorporar nada.
4. Si no ha sido añadido y los datos están completos, los copia a una lista y continúa con el siguiente paciente.
5. Una vez analizados todos los pacientes, se añade la lista de datos al conjunto preexistente, guardándolo como *data.csv*.
6. Con el nuevo dataset se entrena el modelo desde cero, y se almacena como *model.pkl*.

Al terminar, el modelo se habrá actualizado con todos los datos disponibles.

Capítulo 5

Evaluación. Análisis de resultados

Tras optimizar los hiperparámetros de cada uno de los modelos se han calculado las métricas de precisión, sensibilidad y especificidad. Hay varios motivos por los que se ha optado por utilizar estas medidas para la evaluación:

- El artículo original ([Thabtah, 2018a](#)) utiliza estas mismas métricas, lo que permite una comparación directa, aunque como se verá enseguida, dicha comparación solo proporciona una información aproximada.
- El equilibrio entre verdaderos negativos y verdaderos positivos (*precision-recall tradeoff*) es especialmente importante en el contexto de este trabajo. La métrica de sensibilidad es más importante que la de especificidad, por lo que es preferible usarlas antes que, por ejemplo, la métrica F1, que asume la misma importancia relativa para ambas; esto se explica en detalle a continuación.

Modelo	Precisión (<i>Accuracy</i>)	VP (<i>Precision</i>)	VN (<i>Recall</i>)
SVM	96.55	99.36	95.71
Decision tree	90.09	94.87	90.80
Random forests	96.12	97.53	96.93
KNN	94.83	97.48	95.09
Neural network	97.41	100	96.32
Logistic regression	95.26	96.34	96.93
Logistic regression (Thabtah)	97.94	98.00	97.35
Naive Bayes (Thabtah)	92.80	92.80	91.39

Tabla 5.1: Datos obtenidos en función del modelo.

Los resultados obtenidos se pueden ver en la tabla 5.1. Se incluyen dos filas con los datos de Thabtah a efectos comparativos. Debe recordarse que se fusionaron datos de dos fuentes con el fin de ampliar el conjunto de datos de entrenamiento todo lo posible; la mayoría de ellos, en torno a un 85 %, proceden de Thabtah, siendo el otro 15 % de la Universidad de Irvine. El conjunto de datos no es idéntico, por lo que el análisis comparativo debe tomarse tan solo como una aproximación.

El análisis de los datos no deja la más mínima duda: el mejor modelo es la red neuronal, un sencillo perceptrón multicapa, que posee los mejores resultados en los tres apartados. Pero es preciso detenerse un instante y pensar sobre el objetivo del modelo: ¿es acaso conseguir la máxima precisión? ¿Tienen la misma importancia el ratio de verdaderos positivos y el ratio de

verdaderos negativos? ¿Debe mantenerse un equilibrio perfecto entre ambos? La respuesta es no. Este modelo forma parte de un sistema de prediagnóstico: los resultados se utilizarán para llevar a cabo un seguimiento más detallado, o un análisis posterior en profundidad, o un estudio experto. **Por lo tanto, interesa que idealmente todos los usuarios con TEA sean correctamente clasificados, aunque esto conlleve una cierta cantidad de falsos positivos.** De lo contrario, posiblemente algunos usuarios con TEA serían incorrectamente clasificados como neurotípicos, privándoles así de la oportunidad de un diagnóstico posterior. El dato crítico es, por tanto, el ratio de verdaderos negativos (sensibilidad o recall): tiene que ser lo más alto posible. Y con esta métrica, el mejor modelo es random forests, empatado con logistic regression.

¿Cuál escoger entre estos dos? Random forests, por la interpretabilidad de los resultados. Aunque la red neuronal, SVM o logistic regression hubieran arrojado un mejor ratio de verdaderos negativos, random forests podría, por la interpretabilidad, ser aún así la mejor opción. Mientras que los tres primeros funcionan como *cajas negras* (es decir, es complicado explicar las razones por las cuales un usuario es clasificado como TEA o no-TEA), random forests proporciona mucha información útil para entender por qué ocurre la clasificación: es muy sencillo extraer el peso relativo que tiene cada característica, usando el parámetro `feature_importances_`:

```
1 for name, score in zip(df.columns, model_forest.  
    feature_importances_):
```

```
2 print(name, score)
```

Listing 5.1: Peso relativo de las características

Los pesos relativos de cada una de las preguntas son:

```
A1    0.07586443622186195
A2    0.06840572374515735
A3    0.04134026570487984
A4    0.07068724614436704
A5    0.14330543184377942
A6    0.13282806932656827
A7    0.13423164864867243
A8    0.061925549019377864
A9    0.11729853908215723
A10   0.04091860024282477
Age   0.09439596199170101
Sex   0.018798528028652225
```

Cuya visualización gráfica se expone en la Figura 5.1. Puede observarse que el sexo del paciente no tiene apenas influencia, mientras que las preguntas 5, 6, 7 y 9 son especialmente importantes. Es llamativo comprobar que la edad tiene bastante más importancia que las preguntas 3 y 10, que apenas aportan información al modelo. Usando el coeficiente de correlación de Pearson se confirma esta tendencia, como se ve en la Figura 5.2; la pregunta 10 en concreto apenas tiene correlación con el resto. Los datos disponibles son

escasos como para adoptar medidas, pero posiblemente el futuro de este proyecto pase por eliminar las preguntas con menos influencia.

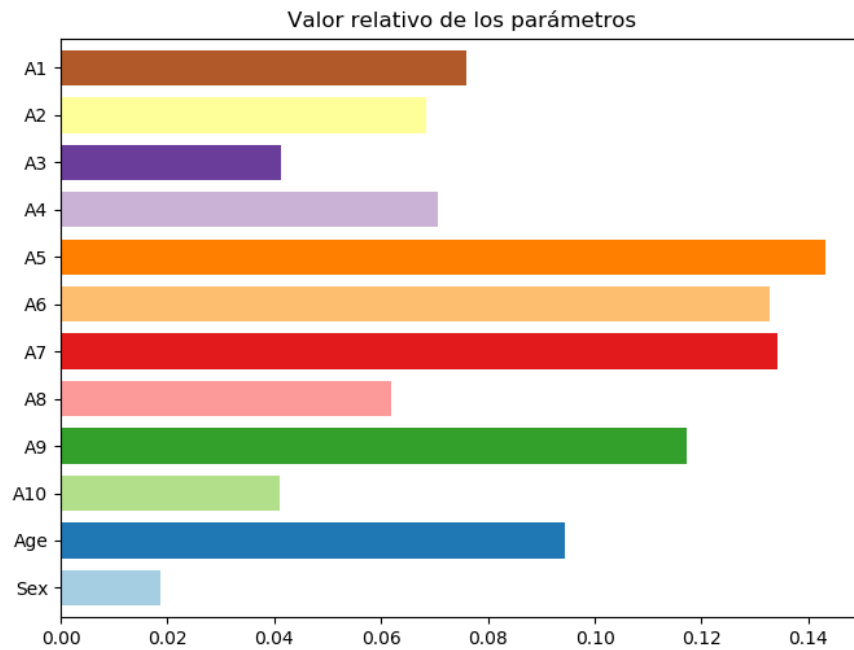


Figura 5.1: Valor relativo de los parámetros.

En cuanto al apartado técnico, la usabilidad ha sido evaluada en un entorno controlado de laboratorio según las heurísticas de Nielsen ([Nielsen, 1995](#)):

- Visibilidad del estado de la web: no hay errores 404, los resultados de las acciones son claros y los links no están caídos ni causan bucles.
- Compatibilidad y concordancia con el usuario: Las nomenclaturas y

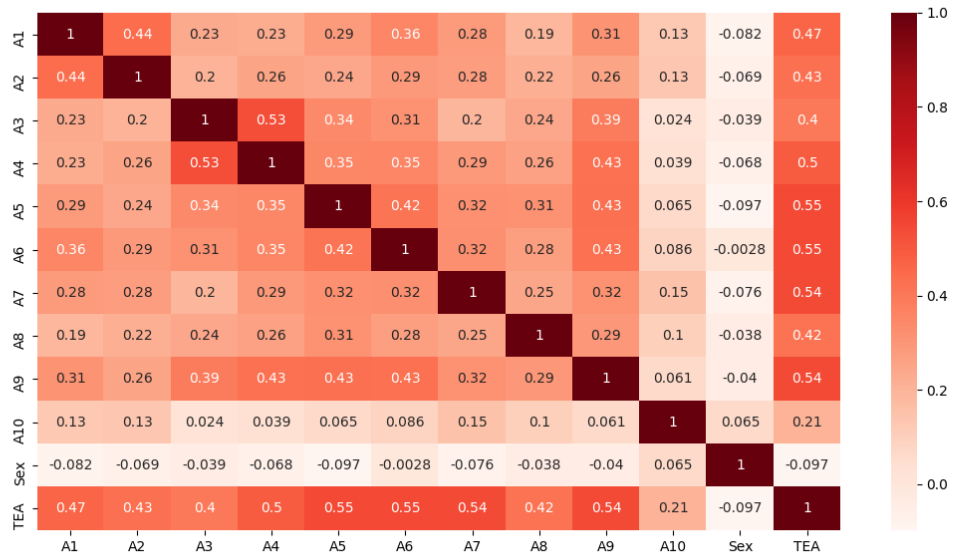


Figura 5.2: Correlación entre los diferentes parámetros.

el estilo se han acordado con el equipo de psicólogos para adaptarlas al lenguaje técnico correcto (por ejemplo, sustituyendo el nombre "paciente" por "sujeto").

- Control y libertad para el usuario: hay opciones de cancelar y una navegación intuitiva; el usuario tiene libertad, dentro de los límites del sistema: por ejemplo, para salir de una actividad antes de tiempo debe indicar el motivo por el cuál la detiene, ya que es un dato de interés para el análisis del usuario.
- Consistencia y estandarización: las páginas siguen el mismo patrón visual y los iconos usados son estándar.

- Prevención de errores: habilitada por medio del sistema de tests, integrado en el sistema y que comprueba que las acciones permitidas al usuario no conllevan errores.
- Reconocimiento en vez de memorización: los elementos comunes no cambian de lugar entre pantallas, la información es clara y directa, sin mucho texto a no ser que esté específicamente solicitado (como en los pop-ups).
- Eficiencia y flexibilidad de uso: los atajos de teclado están habilitados, así como las teclas más comunes: tabulador e intro.
- Estética y diseño: se ha utilizado el diseño de bootstrap4, adaptado especialmente para web y móvil.
- Ayuda al usuario: cuando ocurre un error, se le lleva a la pantalla de error y se le da opción a volver al menú principal.
- Ayuda y documentación: el punto más débil; sin embargo, los pop-ups proporcionan información sobre cada una de las actividades.

Se han realizado pruebas con niños neurotípicos y no se ha encontrado ningún problema reseñable. Sin embargo, es conveniente la presencia de un operario y un terapeuta en pruebas en un entorno real; hasta ese momento no se puede considerar el sistema como maduro. Las pruebas realizadas con varios dispositivos simultáneos indican que el sistema no presenta problemas de rendimiento; sin embargo, la escalabilidad de la solución puede verse comprometida porque la inferencia se lleva a cabo cargando el modelo con cada

petición de cliente. El proyecto se ha diseñado a pequeña escala, por lo que en principio no debería haber ningún problema, pero si en un futuro se quiere ampliar la base de usuarios simultáneos, es posible que haya que rediseñar ese módulo como una API REST.

Capítulo 6

Conclusiones. Trabajos futuros

6.1 Conclusiones

Se ha implementado con éxito el test Q-CHAT-10 de prediagnóstico del espectro del trastorno autista en un servicio web con un robot Nao como actuador principal y un modelo clasificador basado en el algoritmo de random forests. En el contexto del desarrollo de software e implementación, el sistema es usable, intuitivo y accesible, y cumple con todos los requisitos fijados. En el contexto científico, los resultados obtenidos son similares a los de la revisión bibliográfica actual, y permiten realizar una clasificación en tiempo real tan pronto como el usuario termina las actividades. Los datos obtenidos se incorporan al sistema, que se reentrena y aprende en función de las nuevas entradas.

El avance principal que aporta este TFM es la combinación, por primera vez, de las herramientas de detección temprana del autismo con los robots como asistentes para terapia para formar un sistema completo de prediagnóstico. Otro elemento novedoso es que la implementación del Q-CHAT-10 está basada en la interacción con el sujeto en vez de en respuestas de terceros; la extracción de datos es mucho más compleja, pero ha podido automatizarse casi por completo. Con el mismo esfuerzo que en implementaciones previas basadas en terceros, el terapeuta dispone de una fuente de información mucho más rica.

6.2 Trabajos futuros

Aunque no sea un trabajo futuro sino previo, es importante destacar que este TFM surge de una idea más ambiciosa: diseñar desde cero, sin la base del Q-CHAT-10 o el Q-CHAT-25, una herramienta de pre-diagnóstico. Llegaron a implementarse varias actividades notablemente más complejas que las descritas aquí; por ejemplo, en una de ellas Nao pedía al niño que formase una palabra a partir de nueve letras sueltas; una vez completada la tarea, Nao eliminaba una letra de la raíz de la palabra formada, sustituyéndola por otra de frecuencia de aparición similar en el español, y le pedía al niño que formase otra palabra, evaluando su flexibilidad cognitiva. Este trabajo no se ha descartado, sigue en marcha; pero la falta de datos y de población susceptible de formar parte del estudio hizo que hubiera que enfocar el TFM hacia otra

de las áreas con relación con el trastorno del espectro autista¹.

Limitando el estudio de los trabajos futuros al ámbito de este trabajo, hay uno que destaca por su obviedad: solo se han implementado en forma de actividad gamificada seis de las diez preguntas del Q-CHAT-10; aunque la complejidad sea mayor, sería interesante tratar de implementar las diez preguntas. Con ayuda del equipo de psicólogos se ha realizado un estudio previo de las cuatro actividades restantes:

- *Actividad 5. ¿Juega tu hijo/a a juegos de roles?* Se sugiere presentarle al niño unos cuantos juguetes, como coches o muñecos, y observar su reacción. Sin embargo, el papel del Nao no está claro; es totalmente secundario.
- *Actividad 6. ¿Sigue tu hijo/a la dirección de tu mirada?* Se sugiere que Nao juegue con el niño con algún juguete y que posteriormente coja el objeto y se aleje poniéndolo en alto. Nao debe preguntarle dónde está mientras mira al lugar en el que lo ha colocado.
- *Actividad 8. Describirías sus primeras palabras como...* Por desgracia, por la propia naturaleza de la pregunta será necesaria la intervención de los padres o cuidadores en este caso.
- *Actividad 10. ¿Suele tu hijo mirar a la nada sin propósito aparente?* Es un hecho transversal; podría sacarse por mera observación a lo largo

¹Si se desea consultar el código de ese proyecto anterior, el repositorio es <https://github.com/ruromgar/autismws>.

de la sesión, mientras se realizan las demás actividades. De nuevo, Nao no tendría ningún papel aquí.

En el apartado técnico hay varias posibles mejoras o añadidos. Por ejemplo, no se ha implementado entrega ni integración continua, ya que la amplitud actual del proyecto no lo requería, pero el desarrollo se ha realizado pensando en su futura inclusión. Se podría integrar Jenkins con GitHub y los tests funcionales de forma que un *commit* lanzara automáticamente los tests, mostrando al desarrollador si sus cambios son compatibles con lo anterior.

Por último, mencionar como línea de investigación la implementación del sistema en colegios de niños con TEA, para poder recoger datos reales en vivo y comprobar el impacto en un contexto menos controlado que un laboratorio.

Bibliografía

Carrie Allison, Bonnie Auyeung, and Simon Baron-Cohen. Toward brief “red flags” for autism screening: the short autism spectrum quotient and the short quantitative checklist in 1,000 cases and 3,000 controls. *Journal of the American Academy of Child & Adolescent Psychiatry*, 51(2):202–212, 2012.

American Psychiatric Association. *Diagnostic and statistical manual of mental disorders (DSM-5®)*. American Psychiatric Pub, 2013.

Cody Arsenault. Top 10 Front-End Frameworks of 2018. <https://www.keycdn.com/blog/front-end-frameworks>, 2018. Consultado: 02-12-2018.

Len Bass, Paul Clements, and Rick Kazman. *Software architecture in practice*. Addison-Wesley Professional, 2003.

James Bennett, Andrew Cutler, et al. django-registration-redux 2.5 documentation. <https://django-registration-redux.readthedocs.io/en/latest/index.html>, 2019. Consultado: 06-06-2019.

Daniel Bone, Matthew S Goodwin, Matthew P Black, Chi-Chun Lee, Kartik Audhkhasi, and Shrikanth Narayanan. Applying machine learning to

facilitate autism diagnostics: pitfalls and promises. *Journal of autism and developmental disorders*, 45(5):1121–1136, 2015.

Alessandro Crippa, Christian Salvatore, Paolo Perego, Sara Forti, Maria Nobile, Massimo Molteni, and Isabella Castiglioni. Use of machine learning to identify children with autism and their motor abnormalities. *Journal of autism and developmental disorders*, 45(7):2146–2156, 2015.

Sebastian Deterding, Miguel Sicart, Lennart Nacke, Kenton O’Hara, and Dan Dixon. Gamification. using game-design elements in non-gaming contexts. In *CHI’11 extended abstracts on human factors in computing systems*, pages 2425–2428. ACM, 2011.

Django Software Foundation. Django. The web framework for perfectionists with deadlines. <https://www.djangoproject.com/>, 2019. Consultado: 06-06-2018.

M Duda, R Ma, N Haber, and DP Wall. Use of machine learning for behavioral distinction of autism and adhd. *Translational psychiatry*, 6(2):e732, 2016.

Anna M Ern. The use of gamification and serious games within interventions for children with autism spectrum disorder. B.S. thesis, University of Twente, 2014.

Michael Feathers. *Working Effectively with Legacy Code: WORK EFFECT LEG CODE _p1*. Prentice Hall Professional, 2004.

Huanghao Feng, Anibal Gutierrez, Jun Zhang, and Mohammad H Mahoor. Can nao robot improve eye-gaze attention of children with high functioning autism? In *2013 IEEE International Conference on Healthcare Informatics*, pages 484–484. IEEE, 2013.

Jon Fincher. Python IDEs and Code Editors (Guide). <https://realpython.com/python-ides-code-editors-guide/#python-specific-editors-and-ides>, 2019. Consultado: 06-06-2018.

Aurélien Géron. *Hands-on machine learning with Scikit-Learn and Tensor-Flow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc.", 2017.

Carlo Ghezzi, Mehdi Jazayeri, and Dino Mandrioli. *Fundamentals of software engineering*. Prentice Hall PTR, 2002.

Aman Goel. Top 10 Web Development Frameworks in 2018. <https://hackr.io/blog/top-10-web-development-frameworks-in-2018>, 2018. Consultado: 02-12-2018.

Javier J Gutiérrez. ¿Qué es un Framework web? *No available http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf*, 2005.

Juho Hamari, Jonna Koivisto, Harri Sarsa, et al. Does gamification work?-a literature review of empirical studies on gamification. In *HICSS*, volume 14, pages 3025–3034, 2014.

Kazuo Hirai, Masato Hirose, Yuji Haikawa, and Toru Takenaka. The development of honda humanoid robot. In *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No. 98CH36146)*, volume 2, pages 1321–1326. IEEE, 1998.

Jeremy Howard. Kaggle. <https://www.kaggle.com/>. Consultado: 02-06-2019.

Luthffi Idzhar Ismail, Syamimi Shamsudin, Hanafiah Yussof, Fazah Akhtar Hanapiah, and Nur Ismarrubie Zahari. Estimation of concentration by eye contact measurement in robot-based intervention program with autistic children. *Procedia Engineering*, 41:1548–1552, 2012a.

Luthffi Idzhar Ismail, Syamimi Shamsudin, Hanafiah Yussof, Fazah Akhtar Hanapiah, and Nur Ismarrubie Zahari. Robot-based intervention program for autistic children with humanoid robot nao: initial response in stereotyped behavior. *Procedia Engineering*, 41:1441–1447, 2012b.

Kenji Kaneko, Kensuke Harada, Fumio Kanehiro, Go Miyamori, and Kazuhiko Akachi. Humanoid robot hrp-3. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2471–2478. IEEE, 2008.

Karl M Kapp. *The gamification of learning and instruction*. Wiley San Francisco, 2012.

Ann Le Couteur, Gyles Haden, Donna Hammal, and Helen McConachie. Diagnosing autism spectrum disorders in pre-school children using two

- standardised assessment instruments: the adi-r and the ados. *Journal of autism and developmental disorders*, 38(2):362–372, 2008.
- Robert C Martin. Principles of ood. *línea*]. Available: <http://butunclebob.com/ArticleS.UncleBob.PrinciplesOfOod>. [Último acceso: 29 Agosto 2016], 1995.
- Robert C Martin. The interface segregation principle: One of the many principles of ood. *C PLUS PLUS REPORT*, 8:30–36, 1996a.
- Robert C Martin. The liskov substitution principle. *C++ Report*, 8(3):14, 1996b.
- Robert C Martin. The open-closed principle. *More C++ gems*, 19(96):9, 1996c.
- Robert C Martin. Getting a SOLID start, 2009.
- Carla A Mazefsky and Donald P Oswald. The discriminative ability and diagnostic utility of the ados-g, adi-r, and gars for children in a clinical setting. *Autism*, 10(6):533–549, 2006.
- Mohd Azfar Miskam, Mohd Aminullah Che Hamid, Hanafiah Yussof, Syami Shamsuddin, Norjasween Abdul Malik, and Siti Nora Basir. Study on social interaction between children with autism and humanoid robot nao. In *Applied mechanics and materials*, volume 393, pages 573–578. Trans Tech Publ, 2013.

Mohd Azfar Miskam, Nur Fareeza S Masnin, Mohd Hazwan Jamhuri, Syamimi Shamsuddin, Abdul Rahman Omar, and Hanafiah Yussof. Encouraging children with autism to improve social and communication skills through the game-based approach. *Procedia Computer Science*, 42:93–98, 2014a.

Mohd Azfar Miskam, Syamimi Shamsuddin, Mohd Ridzuan Abdul Samat, Hanafiah Yussof, Husna Ahmad Ainudin, and Abdul Rahman Omar. Humanoid robot nao as a teaching tool of emotion recognition for children with autism using the android app. In *2014 International Symposium on Micro-NanoMechatronics and Human Science (MHS)*, pages 1–5. IEEE, 2014b.

Lennart E Nacke and Christoph Sebastian Deterding. The maturing of gamification research. *Computers in Human Behaviour*, pages 450–454, 2017.

Jakob Nielsen. How to conduct a heuristic evaluation. *Nielsen Norman Group*, 1:1–8, 1995.

Yu Ogura, Hiroyuki Aikawa, Kazushi Shimomura, Hideki Kondo, Akitoshi Morishima, Hun-ok Lim, and Atsuo Takanishi. Development of a new humanoid robot wabian-2. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006.*, pages 76–81. IEEE, 2006.

Ill-Woo Park, Jung-Yup Kim, Jungho Lee, and Jun-Ho Oh. Mechanical design of humanoid robot platform khr-3 (kaist humanoid robot 3: Hubo).

In *5th IEEE-RAS International Conference on Humanoid Robots, 2005.*, pages 321–326. IEEE, 2005.

Armin Ronacher. Flask. A Python microframework. <http://flask.pocoo.org/>, 2019. Consultado: 06-06-2018.

Winston W Royce. Managing the development of large software systems: concepts and techniques. In *Proceedings of the 9th international conference on Software Engineering*, pages 328–338. IEEE Computer Society Press, 1987.

Syamimi Shamsuddin, Hanafiah Yussof, Luthffi Ismail, Fazah Akhtar Hanapiah, Salina Mohamed, Hanizah Ali Piah, and Nur Ismarrubie Zahari. Initial response of autistic children in human-robot interaction therapy with humanoid robot nao. In *2012 IEEE 8th International Colloquium on Signal Processing and its Applications*, pages 188–193. IEEE, 2012a.

Syamimi Shamsuddin, Hanafiah Yussof, Luthffi Idzhar Ismail, Salina Mohamed, Fazah Akhtar Hanapiah, and Nur Ismarrubie Zahari. Humanoid robot nao interacting with autistic children of moderately impaired intelligence to augment communication skills. *Procedia Engineering*, 41:1533–1538, 2012b.

Syamimi Shamsuddin, Hanafiah Yussof, Mohd Azfar Miskam, A Che Hamid, Norjasween Abdul Malik, and Hafizan Hashim. Humanoid robot nao as hri mediator to teach emotions using game-centered approach for children

- with autism. In *HRI 2013 Workshop on Applications for Emotional Robots*, 2013.
- Mary Shaw, David Garlan, et al. *Software architecture*, volume 101. prentice Hall Englewood Cliffs, 1996.
- SoftBank Robotics. Nao the humanoid robot. <https://www.softbankrobotics.com/emea/en/nao>, 2019a. Consultado: 28-05-2019.
- SoftBank Robotics. SDKs nao software. <http://doc.aldebaran.com/1-14/dev/sdk.html>, 2019b. Consultado: 20-05-2019.
- Adriana Tapus, Andreea Peca, Amir Aly, Cristina Pop, Lavinia Jisa, Sebastian Pinteau, Alina S Rusu, and Daniel O David. Children with autism social engagement in interaction with nao, an imitative robot: A series of single case experiments. *Interaction studies*, 13(3):315–347, 2012.
- Fadi Thabtah. Autism spectrum disorder screening: machine learning adaptation and dsm-5 fulfillment. In *Proceedings of the 1st International Conference on Medical and Health Informatics 2017*, pages 1–6. ACM, 2017.
- Fadi Thabtah. An accessible and efficient autism screening method for behavioural data and predictive analyses. *Health informatics journal*, page 1460458218796636, 2018a.
- Fadi Thabtah. Machine learning in autistic spectrum disorder behavioral research: A review and ways forward. *Informatics for Health and Social Care*, pages 1–20, 2018b.

Fadi Thabtah, Firuz Kamalov, and Khairan Rajab. A new computational intelligence approach to detect autistic features for autism screening. *International journal of medical informatics*, 117:112–124, 2018.

UCI. UCI Machine Learning Repository. <https://archive.ics.uci.edu/ml/datasets.php>, 2017. Consultado: 14-07-2019.

Aaron Walker. 7 Best Python IDEs for 2019. <https://learn.g2.com/python-ide>, 2019. Consultado: 06-06-2019.

Dennis Paul Wall, J Kosmicki, TF Deluca, E Harstad, and Vincent Alfred Fusaro. Use of machine learning to shorten observation-based screening and diagnosis of autism. *Translational psychiatry*, 2(4):e100, 2012.

Jin'ichi Yamaguchi, Eiji Soga, Sadatoshi Inoue, and Atsuo Takanishi. Development of a bipedal humanoid robot-control method of whole body cooperative dynamic biped walking. In *Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C)*, volume 1, pages 368–374. IEEE, 1999.

Apéndice A

Anexo 1: Código

El código completo del servicio web puede consultarse en el siguiente repositorio: <https://github.com/ruromgar/autismws/tree/tfm>.