

Escuela Politécnica Superior

19
20

Trabajo fin de máster

Sistema de aprendizaje adaptable para personas con TEA



David Doyáguez Sánchez

Escuela Politécnica Superior
Universidad Autónoma de Madrid
C/ Francisco Tomás y Valiente nº 11

Universidad Autónoma de Madrid

Escuela Politécnica Superior



Proyecto para la obtención del título de
Máster en Investigación e Innovación en
Inteligencia Computacional y Sistemas
Interactivos
por la Universidad Autónoma de Madrid



Tutor del trabajo fin de máster:

Germán Montoro Manrique



Sistema de aprendizaje adaptable para personas con TEA

David Doyágüez Sánchez

Todos los derechos reservados.

Queda prohibida, salvo excepción prevista en la Ley, cualquier forma de reproducción, distribución comunicación pública y transformación de esta obra sin contar con la autorización de los titulares de la propiedad intelectual.

La infracción de los derechos mencionados puede ser constitutiva de delito contra la propiedad intelectual (*arts. 270 y sgts. del Código Penal*).

DERECHOS RESERVADOS

© 1 de Junio de 2020 por UNIVERSIDAD AUTÓNOMA DE MADRID
Francisco Tomás y Valiente, n.º 1
Madrid, 28049
Spain

David Doyágüez Sánchez

Sistema de aprendizaje adaptable para personas con TEA

David Doyágüez Sánchez

C\ Francisco Tomás y Valiente N.º 11

IMPRESO EN ESPAÑA – PRINTED IN SPAIN

A mi familia y amigos

Big stories need human stakes.

Aaron Swartz

AGRADECIMIENTOS

A lo largo de la vida de una persona, existen ciertos periodos en los que se lucha solo. El periodo que acaba con la elaboración de este trabajo, sin embargo, no ha sido uno de ellos.

Es por esto que recuerdo a todas las personas que me han apoyado durante la realización de este trabajo, y deseo que todas las personas que no mencione, y que sepan que deben estar aquí, se sientan también partícipes de esta nueva obra y cuenten con mi gratitud.

En primer lugar, agradecer a Germán Montoro, director de este trabajo, que haya conseguido lo imposible, motivarme para continuar con mi etapa académica. Y no sólo eso, si no por permitirme contar siempre con su experiencia, paciencia, consejo, apoyo y dedicación.

También quiero agradecer a Javier Gómez, profesor de la Escuela, compañero del Grupo de Herramientas Interactivas Avanzadas, por sus sabios consejos, apoyo, ánimos y por motivarme a mejorar este trabajo.

Gracias también al Grupo de Herramientas Interactivas Avanzadas por acogerme con cariño y valorar mi trabajo allí, y en especial a Ruth Cobos, Pilar Rodríguez y Francisco Jurado. Es un placer trabajar con todos vosotros y espero seguir cerca próximamente.

Quiero agradecer al Grupo de Modelado e Ingeniería del Software por haberme enseñado muchas de las herramientas necesarias para la realización de este trabajo.

Y, por supuesto, al resto de profesores que me han apoyado durante la realización de este trabajo (y que vienen apoyándome durante mi vida académica), Gonzalo Martínez, Carlos Santacruz, Luis Lago. Gracias por enseñarme a tener esperanza, a cuestionarlo todo, y a no rendirme nunca.

A mis amigos de la universidad, aunque muchos ya han acabado, pero siempre están ahí, y a mis amigos de fuera de la universidad, que siguen ahí, resilientes al paso de los años. Siempre pacientes y comprensivos cuando antepongo el trabajo a todo lo demás, gracias.

Por último, pero no por ello menos importante, a mi familia, en especial a mi madre, mi padre y mi hermano, por su apoyo, fe, paciencia y cariño.

Gracias a todos por no tener que luchar solo. Esta lucha ha terminado, y he aquí el resultado.

RESUMEN

En este trabajo se presenta un sistema de aprendizaje adaptable para personas con TEA. El TEA o trastorno del espectro del autismo es una condición compleja que no se manifiesta de igual manera para todo el mundo, pero que suele llevar asociada dificultades de comunicación e intereses restringidos.

Esta condición complica la educación de las personas que la padecen, que pueden ver sus funciones ejecutivas alteradas, es decir, pueden tener complicaciones para seguir instrucciones del profesorado y mantener la atención durante las sesiones de aprendizaje, reduciéndose su aprovechamiento de las mismas.

Para esto se presenta un sistema que permite modelar al usuario de forma precisa por medio de cuestionarios, que podrán ser generados fácilmente por los docentes o expertos en TEA. Para conseguir que esta generación sea posible para los expertos, se ha creado una herramienta de autor que hace esta tarea más sencilla.

Estos cuestionarios serán utilizados para adaptar automáticamente una herramienta de aprendizaje para Android de sumas sencillas, de tal manera que concuerde con los intereses del alumno con TEA. Para realizar la adaptación se hace uso de un servidor web gracias al que se pueden almacenar y transmitir los datos de los usuarios y las respuestas a los cuestionarios entre la herramienta de autor y la herramienta de aprendizaje.

Este sistema se ha diseñado e implementado en colaboración con un colegio de educación especial, lo que ha permitido que el sistema se adapte a las necesidades del colectivo al que va dirigido y sea compatible con la operativa del mismo, cumpliendo con una serie de requisitos de usabilidad, ubicuidad y seguridad.

Por último, aunque debido a las recientes circunstancias excepcionales no ha sido posible probarlo con usuarios reales en el colegio tal y como estaba previsto, se han realizado una serie de evaluaciones sobre los requisitos, una evaluación heurística, y una evaluación de expertos para comprobar que el resultado final cumple con las expectativas, y demostrar que puede ser una buena herramienta para mejorar la educación de personas con TEA.

PALABRAS CLAVE

TEA, autismo, adaptable, personalizable, adaptación, personalización, app, herramienta

ABSTRACT

In this project it is presented an adaptable learning system made for people with ASD. ASD or autism spectrum disorder is a complex condition that manifests differently for each person, but it can bring communication difficulties and restricted interests.

This condition complicates education of people that suffer it, because they can see their executive functions altered, namely, they can have complications to perform instructions given by their teachers and to keep attention while teaching sessions are going on, impacting their performance on those sessions.

In order to approach this issue, a system that can model the user in a precise way via questionnaires is presented. These questionnaires can be easily generated by teachers or experts using a specific authoring tool that has been created.

These questionnaires will be used to automatically adapt an easy additions Android learning tool, so it goes along with the student with ASD's preferences. In order to perform the adaptation, a web server is used to store and to transmit users' data and questionnaires between the authoring tool and the learning tool.

This system has been designed and implemented in collaboration with a special education school, and that has made possible for the system to be adapted to the school's needs and also to be compatible with its operative, complying with a series of usability, ubiquity and security requirements.

Finally, although it has not been possible to test the system with real users in the school, a series of evaluations over the requirements, a heuristic evaluation and an evaluation made by experts have been done to prove that the final result meets the expectations and that it can be a good tool to improve education of people with ASD.

KEYWORDS

ASD, autism, adaptable, customizable, adaptation, customization, app, tool

ÍNDICE

1	Introducción	1
1.1	Motivación	1
1.2	Alcance	2
1.3	Estructura del documento	3
2	Estado del Arte	5
3	Diseño	13
3.1	Requisitos	14
3.2	Casos de uso	16
3.3	Arquitectura	16
3.3.1	Herramienta de autor	16
3.3.2	Servidor	18
3.3.3	Aplicación móvil	18
3.3.4	Adaptabilidad de la aplicación	19
4	Desarrollo	21
4.1	Herramienta de autor	21
4.1.1	Meta-modelo de conjuntos de preguntas	21
4.1.2	Editor gráfico de conjuntos de preguntas	23
4.1.3	Transformación a JSON del conjunto de preguntas	23
4.1.4	Sincronización automática con el servidor	24
4.1.5	Acceso al servidor	24
4.1.6	Modularidad de la herramienta	25
4.1.7	Mejoras en la interfaz	25
4.2	Servidor	26
4.2.1	Interfaz de administración	26
4.2.2	API REST	26
4.2.3	Patrón Modelo-Vista-Controlador	26
4.2.4	Modularidad del software servidor	28
4.3	Aplicación móvil	29
4.3.1	Interfaz de la aplicación	29
4.3.2	Mecánica del sistema de aprendizaje	29
4.3.3	Adaptabilidad del sistema	30

4.3.4 Idempotencia y adaptatividad del sistema	31
5 Integración, pruebas y resultados	33
5.1 Trazabilidad de requisitos	33
5.2 Evaluación heurística de usabilidad	47
5.3 Evaluación básica por parte de expertos	48
6 Conclusiones y trabajo futuro	51
6.1 Conclusiones	51
6.2 Trabajo futuro	52
6.2.1 Mejoras en el trabajo	52
6.2.2 Nuevos desarrollos que hagan uso de este trabajo	53
Bibliografía	58
Apéndices	59

LISTAS

Lista de algoritmos

Lista de códigos

Lista de cuadros

Lista de ecuaciones

Lista de figuras

3.1	Diagrama de casos de uso	17
3.2	Diagrama de casos de uso orientado a la arquitectura	20
4.1	Diagrama del metamodelo	22
5.1	Página web de creación de nuevo usuario	34
5.2	Interfaz de la herramienta de autor	35
5.3	Creando un nuevo conjunto de preguntas	35
5.4	Editando un conjunto de preguntas	36
5.5	Accediendo al listado de preguntas desde la web	37
5.6	Accediendo al listado de preguntas desde la app	38
5.7	Simulando preguntas desde la interfaz web	39
5.8	Menú principal de la aplicación	40
5.9	Listado de usuarios en la aplicación	40
5.10	Instrucciones en el juego	41
5.11	Recompensa visual del juego	42
5.12	Instrucciones en minúscula en el juego	43
5.13	Diferentes colores y objetos en el juego	44
5.14	Juego con todos los objetos iguales	45
5.15	Juego con todos los objetos desordenados	46
5.16	Juego mejorado con el feedback de expertos	49

Lista de tablas

2.1	Tabla de número de resultados por consulta y base de datos	7
2.2	Tabla de número de resultados por consulta acotados	7
2.3	Tabla de número de resultados por consulta quitando duplicados.....	7
2.4	Tabla de número de resultados por consulta descartando artículos fuera del ámbito...	8
2.5	Segunda tabla de número de resultados por consulta y base de datos	9
2.6	Segunda tabla de número de resultados por consulta eliminando duplicados.....	9
2.7	Segunda tabla de número de resultados por consulta eliminando artículos fuera del ámbito	9

Lista de cuadros

INTRODUCCION

El autismo ha sido definido como un trastorno del desarrollo neurológico, caracterizado por una serie de dificultades para comunicarse, para entender la conducta de los demás y de relacionarse, al igual que para pensar y comportarse de manera flexible. En el año 2013 se publica el DSM-V con una nueva definición para el diagnóstico del autismo, Trastorno del Espectro del Autismo, o TEA para abreviar, en el que se describen una serie de alteraciones que consisten en:

- 1.- Dificultades persistentes de comunicación e interacción social en múltiples contextos.
- 2.- Patrones restringidos y repetitivos de conducta, intereses o actividades.

Para darse un caso de TEA, estos déficits deben ser persistentes (no aparecen y desaparecen con el paso del tiempo) y manifestarse en una variedad de contextos (no se limitan exclusivamente al colegio o al hogar).

Las características que definen el TEA se manifiestan de forma muy diferente en cada persona, y eso presenta un problema a la hora de aplicar terapias o de abordar actividades de enseñanza, debido a que no se pueden generalizar las soluciones a todas las personas con TEA.

1.1. Motivacion

En el ámbito educativo, los dispositivos móviles tienen cada vez más importancia, dadas las ventajas que proporcionan en cuanto a ubicuidad, flexibilidad y capacidad para poder personalizar la experiencia educativa de los alumnos [1].

Es precisamente esta capacidad de personalización la que convierte a los dispositivos móviles en una herramienta muy potente para mejorar la educación de las personas con TEA. Si se puede adaptar la experiencia educativa a las necesidades particulares de una persona con TEA, de manera que sea posible comunicarse bidireccionalmente con ella, el aprendizaje será mejor [2].

Para realizar esta comunicación con el estudiante, es esencial entender sus preferencias. Muchos sistemas de aprendizaje actuales se basan en la adaptatividad para mejorar la experiencia, es decir, recolectan datos de uso para mejorar su funcionamiento y adaptarlo al usuario [3].

Esta aproximación, aunque es muy válida para personas neurotípicas, no es la mejor opción para personas con TEA debido al problema de *arranque frío*, que es la falta de datos iniciales sobre el usuario por parte del sistema, y lleva a un funcionamiento inicial del mismo no personalizado [4]. En el caso de las personas con TEA, y en particular en el caso de los alumnos jóvenes, es necesario evitar este problema, y ofrecer una experiencia adaptada desde el principio por varios motivos:

- 1.– Educativos: En el caso de alumnos muy jóvenes, es necesario que el aprendizaje se produzca en los tiempos adecuados para ello.
- 2.– Económicos: Los dispositivos inteligentes no son baratos, y realizar una inversión en uno de estos dispositivos genera una serie de expectativas que deben cumplirse. Además, se han conocido casos de usuarios con TEA que han dañado dispositivos debido a la frustración que les producían ciertas sesiones.
- 3.– Temporales: El tiempo que una persona con TEA pasa utilizando un producto destinado a mejorar su educación, es tiempo que no pasa con su familia, es por esto que el producto debe ser efectivo desde el principio y no se puede abusar del tiempo del usuario.

1.2. Alcance

Es por todo esto que se propone un sistema adaptable para personas con TEA, que permita que reciban una experiencia educativa personalizada en función de sus intereses y preferencias personales, y que optimice la inversión económica y temporal que realizan las familias y los centros de educación especial.

Para ello, el sistema modelará al usuario mediante preguntas de opción múltiple, que tendrán como objetivo obtener información sobre el estado del usuario y que podrán ser definidas por el centro educativo, y aplicará esta información para adaptar desde el primer momento una herramienta de aprendizaje de sumas sencillas, con el objeto de que los alumnos obtengan competencias básicas en aritmética.

Esta adaptación se realizará mediante un sistema de objetivos, es decir, unos elementos asociados a una o varias respuestas en particular (previamente definidas por el centro), que serán interpretados por la aplicación móvil para realizar las adaptaciones en tiempo de ejecución de la herramienta de aprendizaje.

Estas adaptaciones consistirían en cambiar diversos aspectos de la forma en la que se presenta la información en la aplicación, de modo que concuerde con los intereses (u otros factores definidos por el centro) de los usuarios, lo que permitiría que la sesión de aprendizaje fuera más amena y efectiva.

Este sistema estará conformado por tres elementos:

- 1.– Una herramienta de autor que permitirá a los docentes del centro generar conjuntos de preguntas sin necesidad de saber programar.
- 2.– Una aplicación móvil que permitirá a los docentes contestar esas preguntas en relación a un alumno en concreto, y que a la vez permitirá a los alumnos interactuar con una herramienta de aprendizaje de sumas

sencillas.

3.– Un servidor web que mantendrá los datos de los alumnos del centro y que permitirá probar los conjuntos de preguntas generados antes de contestarlos en la aplicación.

El sistema será extensible y modular, con el objetivo de que este trabajo sirva también de base para futuros desarrollos, reduciendo así el coste de mejorar y ampliar este trabajo en el futuro, a costa de que el tiempo de desarrollo actual sea mayor y, por lo tanto, el alcance del sistema sea menor, dado el tiempo limitado del que se dispone para la realización del trabajo.

1.3. Estructura del documento

Esta memoria consta de los siguientes capítulos:

- Estado del Arte: En el que se detalla el panorama actual de la tecnología en el ámbito del TEA, haciendo incisos en los aspectos que más se asemejan a este trabajo.
- Diseño: En el que se explican las decisiones de diseño tomadas para modelar un sistema modular, extensible y que se adapte a las necesidades de los usuarios.
- Desarrollo: En el que se detalla la implementación llevada a cabo en base al diseño anteriormente explicado.
- Integración, pruebas y resultados: En el que se explica por qué el trabajo realizado es válido y qué métodos de evaluación han sido utilizados.
- Conclusiones: En el que se recapitula sobre lo que se ha hecho, para qué sirve, y qué queda por hacer.

ESTADO DEL ARTE

Para el desarrollo de este trabajo se ha realizado una revisión de la literatura, que se ha considerado imprescindible para poder entender qué tecnología hay implementada en la actualidad con respecto al tema que vamos a tratar. Identificar qué desarrollos o métodos existen es importante porque pueden ayudar a mejorar la calidad del resultado final, permitiendo identificar tanto qué métodos funcionan y se están utilizando con éxito para mejorar el desarrollo de las funciones ejecutivas en personas con TEA, como para entender qué métodos no funcionan.

Una parte importante de este trabajo es el de implementar un sistema modular y extensible que permita modelar al usuario con TEA a partir de una serie de preguntas y respuestas, que deben poder ser definidas por un equipo de profesionales que no tienen por qué tener conocimientos de informática, y que dada la generalidad requerida para este sistema también es posible que existan desarrollos similares aplicados a otros ámbitos.

Para realizar esta revisión de la literatura, se han seguido una serie de protocolos estandarizados [5, 6], de forma que la revisión se ha realizado de la siguiente forma:

- Se ha buscado literatura en las bases de datos ACM, IEEE Xplore, Elsevier y EBSCO.
- Se han incluido estudios primarios tales que:
 - Traten de métodos que permitan a personas con TEA desarrollar sus funciones ejecutivas.
 - Traten de métodos para identificar a través de preguntas las áreas en las que una persona con TEA necesita de más apoyo.
 - Traten de métodos que permitan a usuarios sin TEA y no programadores definir una secuencia de preguntas de forma sencilla, para permitir posteriormente una evaluación personalizada de usuarios con TEA.
- Se han excluido estudios primarios tales que:
 - Sean anteriores al año 2009. Esto es debido a que los smartphones aún no contaban con la suficiente popularidad. [7]
 - Que no hayan sido probados en usuarios reales.
- Una vez identificados los estudios primarios, se ha procedido a su análisis, buscando los siguientes elementos:
 - Breve descripción del método utilizado.

- Tipo de técnica de evaluación.
- Tipo de análisis.
- Número de usuarios que han participado en el estudio
- Efectividad del método (porcentaje de éxito del método).
- Ventajas asociadas al método.
- Desventajas asociadas al método.
- Casos particulares en los que el método no es aplicable.
- El método ha sido implementado o no en forma de aplicación móvil.

Teniendo en cuenta estos criterios, se han realizado las siguientes consultas para cada base de datos:

- 1.– ((ASD OR autism OR autist) AND (executive))
- 2.– ((ASD OR autism OR autist) AND (smartphone OR tablet OR smart) AND (executive))
- 3.– ((ASD OR autism OR autist) AND (smartphone OR tablet OR smart) AND (question OR questions) AND (classify OR model))
- 4.– ((define) AND (questions) AND (interface) AND (non-programmer OR inexperienced OR non-technical))
- 5.– ((custom OR personalized) AND (evaluation OR (questions AND answers)) AND (non-programmer OR inexperienced OR non-technical))
- 6.– ((user modelling) AND (question OR questions) AND (answer OR answers) AND ((adapt OR modify) interface))

Como algunas bases de datos arrojan muchos resultados para algunas consultas, se ha tratado de acotar la búsqueda utilizando una serie de consultas más específicas:

- 7.– ((ASD OR autism OR autist) AND (smartphone OR tablet OR smart) AND (executive) AND (emotional regulation) AND (improve) AND (teach))
- 8.– ((tool to generate (questionnaire OR questionnaires)) to (adapt OR modify) (interface OR therapy OR app)))

Ejecutando las consultas, se han identificado los siguientes estudios primarios en cada base de datos (nótese que se han separado en dos ámbitos de búsqueda, uno para el ámbito de TEA y tecnología, y otro para el ámbito de desarrollo de cuestionarios automatizados por parte de profesionales no programadores y su uso para modelar usuarios y adaptar interfaces):

	ACM	IEEE Xplore	Elsevier	EBSCO - ERIC
Consulta 1	13	5	337	231
Consulta 2	3	1	131	1
Consulta 3	1	0	136	0
Consulta 7	94	0	0	0
Consulta 4	29	1	46325	0
Consulta 5	890	1	3140	0
Consulta 6	482	13	13415	0
Consulta 8	265160	0	15899	0

Tabla 2.1: Número de resultados obtenidos ejecutando cada consulta en las bases de datos

Aún habiendo intentado generar consultas más específicas, para el tema de modelado del usuario a través de preguntas no se ha logrado acotar la búsqueda. Se han tenido en cuenta algunos autores recomendados por expertos, de modo que finalmente se han leído los siguientes artículos por consulta:

C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	Otras fuentes
13	2	1	1	15	10	8	0	12

Tabla 2.2: Número de resultados por consulta acotados

Al eliminar artículos duplicados obtenemos que en total, para cada consulta tenemos el siguiente número de artículos:

C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	Otras fuentes
11	1	0	1	13	8	8	0	12

Tabla 2.3: Número de resultados por consulta sin contar duplicados

Después de leer esos artículos, se han descartado otra serie de artículos, o bien por no tener finalmente una aplicación al caso actual, o no haber sido probados empíricamente, de forma que se tendrán en cuenta finalmente el siguiente número de artículos para realizar el análisis:

C 1	C 2	C 3	C 4	C 5	C 6	C 7	C 8	Otras fuentes
2	1	0	0	2	2	3	0	2

Tabla 2.4: Número de resultados por consulta cribando artículos fuera del ámbito

De la revisión anterior, hemos podido comprobar que en el ámbito de terapias orientadas a trabajar las funciones ejecutivas destacan los programas que guían al usuario descomponiendo las tareas que se desea que aprendan en subtareas más sencillas [8, 9].

En la misma línea, también destacan los sistemas calendarios, que permiten al usuario organizar de forma fácil las tareas que tienen que realizar al día o a la semana, de manera que puedan tener un seguimiento del progreso, tanto ellos como sus terapeutas [10].

Para que el aprendizaje sea efectivo, es necesario darle a los usuarios con TEA un *feedback* para que puedan saber si lo están haciendo bien, y si fallan corregir su actividad. Hay estudios en los que se demuestra que la importancia de esta retroalimentación aumenta cuanto más joven es el usuario, y que se puede influir en el aprendizaje de la auto-determinación usando *feedbacks* erróneos [11].

El auge de nuevas tecnologías móviles y el denominado *Internet de las Cosas* está permitiendo aplicar nuevas terapias al ámbito del TEA, de modo que se pueden usar muñecos conectados a dispositivos, y que se pueden tocar para proporcionar mayor sensación de inclusión en la terapia [12]; también se pueden analizar los datos de los sensores de acelerómetros o giroscopios, de los que se pueden descubrir cosas interesantes, como que a los usuarios más jóvenes con TEA les resulta más complicado sostener una tablet [13].

Esta nueva tendencia del *Internet de las Cosas* está acercando la tecnología al día a día de gente con pocos conocimientos de diseño y desarrollo de software, de manera que nace una nueva necesidad de permitir al usuario final tener el poder de configurar el comportamiento de estos sistemas. Esta tendencia se alinea con la necesidad que presentan los usuarios con TEA de recibir terapias personalizadas, y por lo tanto de la necesidad de los terapeutas de poder crear estas personalizaciones de un modo ameno, sin que el uso de la tecnología sea un impedimento y consuma mucho tiempo.

Para esto, han demostrado ser de utilidad herramientas que presentan un sistema de depuración, para que el usuario pueda probar su desarrollo antes de desplegarlo [14].

El paradigma de desarrollo más extendido en este tipo de herramientas es el visual, es decir, programas que permiten realizar configuraciones mediante el dibujo de diagramas [15], aunque también existen soluciones que utilizan programación basada en eventos usando asistentes [16]; por último, con las explicaciones adecuadas también es posible usar funcionalidades simplificadas de lenguajes de programación tradicionales [17].

Todo esto tiene aplicación en un campo como el de los sistemas adaptativos, que pueden usarse para mejorar el aprendizaje en general [18] y para recomendar contenido personalizado, lo que tiene

también interesantes implicaciones en el diseño de los sistemas, habiendo llegado a desarrollarse conceptos como el de *learning objects* y estándares para definirlos como el *IEEE LOM* [19].

Mediante estas referencias iniciales, se han podido extraer conceptos interesantes, como el concepto de *End-User Development* o de *learning objects*, conceptos que pueden ser usados en nuevas consultas para ampliar la bibliografía:

9.– (EUD) AND (ASD OR autism)

10.– (learning object) AND (ASD OR autism)

Con las que se han obtenido los siguientes resultados:

	ACM	IEEE Xplore	Elsevier	EBSCO - ERIC
Consulta 9	1	0	8	0
Consulta 10	1968	32	5549	9

Tabla 2.5: Número de resultados por consulta y base de datos

Cribando duplicados y artículos que no tenían nada que ver, se han leído los siguientes artículos por consulta:

Consulta 9	Consulta 10
1	55

Tabla 2.6: Número de resultados por consulta eliminando los duplicados

Después de leer esos artículos, se han descartado algunos por no tener suficiente relación con el tema. Por lo tanto, al final se tendrán en cuenta el siguiente número de artículos para el análisis:

Consulta 9	Consulta 10
1	15

Tabla 2.7: Número de resultados por consulta eliminando los artículos fuera del ámbito

Mediante este nuevo análisis, se han podido descubrir herramientas específicas para crear contenidos y terapias para personas con TEA.

Un ejemplo muy representativo es el de los cuentos interactivos, que los terapeutas pueden modificar a placer para que se adapte a los intereses de niños con TEA. Esto es muy importante porque cada uno tiene unos intereses muy particulares, y para que la terapia sea efectiva es necesario mantener al niño atento. También se da la capacidad a los terapeutas de controlar la sesión de forma remota, de manera que se puede pausar, resumir o reiniciar e incluso programar un *feedback* del cuento al niño en tiempo real [20].

La capacidad de ajustar parámetros en tiempo real es una característica que parece ser deseable dado que muchos desarrollos lo han implementado, de forma que se pueden adaptar mundos virtuales [21] o incluso juegos físicos [22] en base a la habilidad del usuario.

En esta línea, de nuevo encontramos que los entornos de desarrollo visuales parecen ser efectivos para implementar mecánicas personalizadas en terapias [23].

También se han encontrado más ejemplos de aplicaciones que permiten a usuarios con TEA gestionar sus tareas pendientes [24] o acostumbrarse a llevar hábitos de vida saludables [25].

Este tipo de aplicaciones producen un gran impacto positivo en la integración de las personas con TEA en la sociedad, pero para lograr una mejor integración es esencial trabajar la autorregulación emocional [26]. Los nuevos avances en realidad aumentada y reconocimiento facial permiten a las personas con TEA que lo necesitan aprender a reconocer las emociones [27], una habilidad que puede ayudarles mucho a formar un círculo de personas de confianza en las que apoyarse cuando lo necesiten.

En ese sentido, se han hecho también estudios de cómo personas de confianza pueden ayudar a las personas con TEA en asuntos de la vida cotidiana haciendo uso de redes sociales [28]. Lo que se intenta es que la persona con TEA no se aisle, y que pueda contactar con gente de confianza en situaciones que considere necesarias, que pueden ser incluso hasta de peligro [29].

Las personas con TEA también pueden sufrir aislamiento en el ámbito laboral, por lo que existen métodos para ayudarles no sólo a trabajar habilidades deseables en el mundo laboral, como por ejemplo la programación [30], sino que también hay maneras de ayudarles a trabajar sus habilidades sociales [31], emocionales, conversacionales [32] o incluso económicas [33].

Todas estas intervenciones que tienen como fin mejorar las competencias de las personas con TEA requieren de una gran capacidad de adaptación y personalización, y se vuelve necesario mantener un registro del progreso de las terapias, registros que pueden ser usados por sí mismos como terapias [34]. Gracias a los registros de las terapias se pueden adaptar las intervenciones a corto plazo o incluso sobre la marcha [35, 36], mejorando el efecto de las mismas.

En definitiva, se puede comprobar que en la investigación en TEA actualmente hay numerosos usos de las nuevas tecnologías, y en concreto de los dispositivos móviles como herramientas para mejorar el aprendizaje, dada la potencia y flexibilidad que permiten. Sin embargo, esta potencia y flexibilidad aumentan la complejidad de implementación de estos sistemas, que en este caso son utilizados por educadores y expertos en TEA que suelen ser ajenos a los aspectos más técnicos.

Es por esto que es necesario realizar herramientas de autor que permitan a estos expertos adecuar el sistema a las necesidades particulares de los usuarios. Estas herramientas para adaptar sistemas no sólo se utilizan en el ámbito educativo y en el TEA, sino que también se dan en otros ámbitos en auge, como por ejemplo el *Internet de las Cosas*. En este ámbito la investigación es muy extensa, y

se ha podido observar cierta prevalencia de las herramientas de autor visuales, lo que ha influido en la decisión de crear una herramienta de autor basada en diagramas para este trabajo.

En la investigación se ha observado también que existen sistemas que requieren de la intervención del experto durante la ejecución para realizar adaptaciones. Esto tiene mucho sentido para una sesión individual, pero si lo que se quiere es realizar una sesión en un aula con múltiples usuarios de, por ejemplo, un colegio de educación especial, esto no es posible. Por eso para este trabajo se ha determinado que el modelado del usuario debe ser previo, y dada la necesidad de que este modelado sea preciso, tiene sentido el uso de cuestionarios creados previamente por los expertos.

DISEÑO

En este trabajo se ha desarrollado un sistema de aprendizaje adaptable para personas con TEA. Más en concreto, este sistema permitirá a los usuarios con TEA aprender a realizar sumas sencillas a través de una aplicación para Android.

Para poder modelar al usuario de forma precisa a partir de cuestionarios, que pueden ser variables en función del criterio de los expertos, se ha implementado una herramienta de autor que permite generar estos cuestionarios mediante la creación de diagramas, es decir, con la herramienta se diseñan los cuestionarios como si fueran diagramas.

Estos diagramas se almacenan en un servidor en línea, de forma que puedan ser accedidos de forma ubicua. Además, este servidor proporciona una interfaz web que permite gestionar los usuarios y probar los cuestionarios desarrollados con la herramienta de autor antes de responderlos en la aplicación móvil.

La aplicación móvil tiene una doble función, modelar al usuario con TEA contestando las preguntas y ofrecer una herramienta de aprendizaje de sumas sencillas con la que puedan aprender y esté adaptada a sus necesidades desde el primer momento en el que empiezan a utilizarla.

Esta adaptación es posible al tener en las respuestas a los cuestionarios la información necesaria para realizar la adaptación. Los autores de los cuestionarios no sólo definen la secuencia de preguntas (que, como veremos más adelante, es un grafo) si no que definen también las posibles respuestas y la adaptación que se va a dar en función de dichas respuestas (por medio de un sistema de objetivos que también explicaremos más adelante).

En definitiva, el conjunto de preguntas y respuestas permiten modelar al usuario de forma precisa, y además en ese modelo se especifican las adaptaciones que se van a realizar mediante un sistema de objetivos integrado en los conjuntos de preguntas, y que es interpretable por la aplicación para hacer efectiva la adaptación.

3.1. Requisitos

Gracias a la colaboración de un colegio de educación especial y al estado del arte, se han elucidado los siguientes requisitos funcionales:

RF-1: Capacidad para poder adaptar sesiones de aprendizaje a las necesidades particulares de cada alumno.

RF-2: Capacidad para poder agregar usuarios al sistema.

RF-3: Capacidad para poder agregar cuestionarios de forma sencilla, que luego podrán ser respondidos de forma ubicua.

RF-4: Capacidad para poder definir qué adaptaciones se aplicarán en la sesión a partir de las respuestas a esos cuestionarios.

RF-5: Capacidad para que la persona que los define, o un tercero, pueda probar los cuestionarios y ver qué adaptaciones se aplicarían dadas ciertas respuestas.

RF-6: Capacidad para poder seleccionar tanto el usuario con TEA como el cuestionario a los que se va a asociar una respuesta.

RF-7: Capacidad para poder ejecutar una sesión de aprendizaje que consista en resolver sumas sencillas seleccionando el usuario con TEA al que se le va a aplicar y el estado en el que se encuentra, que ha sido previamente modelado mediante un conjunto de respuestas a un cuestionario.

La implementación de estos requisitos, según el colegio, son compatibles con la operativa actual de los centros de educación especial y permitirían que el sistema pudiera ser usado durante las clases.

Existen otros requisitos, igualmente importantes pero más relacionados con la terapia o sesión que se va a dar al alumno con TEA. En concreto, la sesión podrá ser adaptada en los siguientes términos generales:

RF-8: Capacidad de dar instrucciones en texto o no darlas.

RF-9: Capacidad de dar las recompensas en formato animado, audio, animada y audio a la vez, o ninguna.

RF-10: Capacidad de que las letras sean todas en mayúscula o no.

RF-11: Capacidad de que el color de fondo pueda ser personalizado.

Estos requisitos encuentran su razón de ser en que todos los alumnos con TEA tienen una serie de preferencias personales, que pueden ser utilizadas en todas las sesiones para conseguir un mejor aprovechamiento de las mismas. Si queremos que nuestro sistema sea extensible y reutilizable (para que el mismo software sea usable por personas distintas), es necesario tener en cuenta estas

preferencias.

Sin embargo, en este TFM se ha desarrollado una aplicación para aprender a sumar, y para lograr ese objetivo concreto también es necesario conseguir que la sesión de aprendizaje de sumas sencillas sea adaptable. En términos particulares, la sesión podrá ser adaptada de las siguientes maneras:

RF-12: Capacidad de que los números vayan acompañados de objetos o no.

RF-13: En caso de que haya objetos, capacidad de poder definir el tipo de los objetos, que pueden ser: dedos, bolas, coches, dinosaurios o planetas.

RF-14: Capacidad de que siendo los objetos de una misma categoría sean todos los objetos iguales o no.

RF-15: Capacidad de que los objetos se muestren en orden o no.

Estos requisitos son necesarios debido a que cada alumno con TEA tiene un modelo de aprendizaje muy particular, y lo que funciona con un alumno puede no funcionar con otro.

También se han recopilado los siguientes requisitos no funcionales, que están relacionados con la operativa de los colegios de educación especial:

RNF-1: Separación de la funcionalidad utilizada por los usuarios con TEA de la funcionalidad de los docentes.

RNF-2: Protección de la configuración realizada por los docentes.

RNF-3: Sincronización de configuraciones entre dispositivos.

3.2. Casos de uso

Cabe destacar que el sistema presentado en este trabajo, a pesar de estar orientado a a su uso con personas con TEA, tiene una serie de funcionalidades que pueden ser aplicadas a otros ámbitos, en concreto toda la funcionalidad de generación de cuestionarios y de gestión de usuarios. Es por esto que se ha utilizado una terminología genérica a la hora de diseñar el sistema.

A partir de los requisitos del apartado anterior, se crea un diagrama de casos de uso mostrado en la figura 3.1.

Entrando más en detalle en el uso del sistema, la operativa sería una secuencia tal que:

- 1.– Se da de alta a la organización (en este caso el colegio) en el sistema.
- 2.– Se da de alta a un usuario administrador para esa organización (en este caso la directora del colegio).
- 3.– El administrador de la organización da de alta de los autores en el sistema (en este caso los docentes del colegio).
- 4.– El administrador da de alta a los clientes (en este caso los estudiantes del colegio) en el sistema.
- 5.– El autor genera el cuestionario.
- 6.– El autor selecciona al estudiante concreto con el que se va a trabajar.
- 7.– El autor, el estudiante, o un tercero en representación del estudiante contesta a las preguntas.
- 8.– El estudiante interactúa con la aplicación.

Generalizando los casos de uso se puede lograr que el resultado final sea también generalizable a distintos ámbitos.

3.3. Arquitectura

Una vez elucidados los requisitos y estando claros los casos de uso, se procede al diseño del sistema. Dados los requisitos de ubicuidad, se opta por seguir una arquitectura de tipo cliente-servidor, de forma que los datos puedan ser accesibles desde múltiples dispositivos. Se genera otro diagrama de casos de uso más enfocado a mostrar la arquitectura del sistema (figura 3.2).

Podemos comprobar que existen tres elementos que componen el sistema, y que se van a explicar más en detalle a continuación.

3.3.1. Herramienta de autor

La herramienta de autor permite la generación de cuestionarios en forma de diagramas por parte de los autores, dado que ya se comprobó en el estado del arte que la generación de diagramas era una buena aproximación para acercar los desarrollos a personas sin conocimientos de programa-

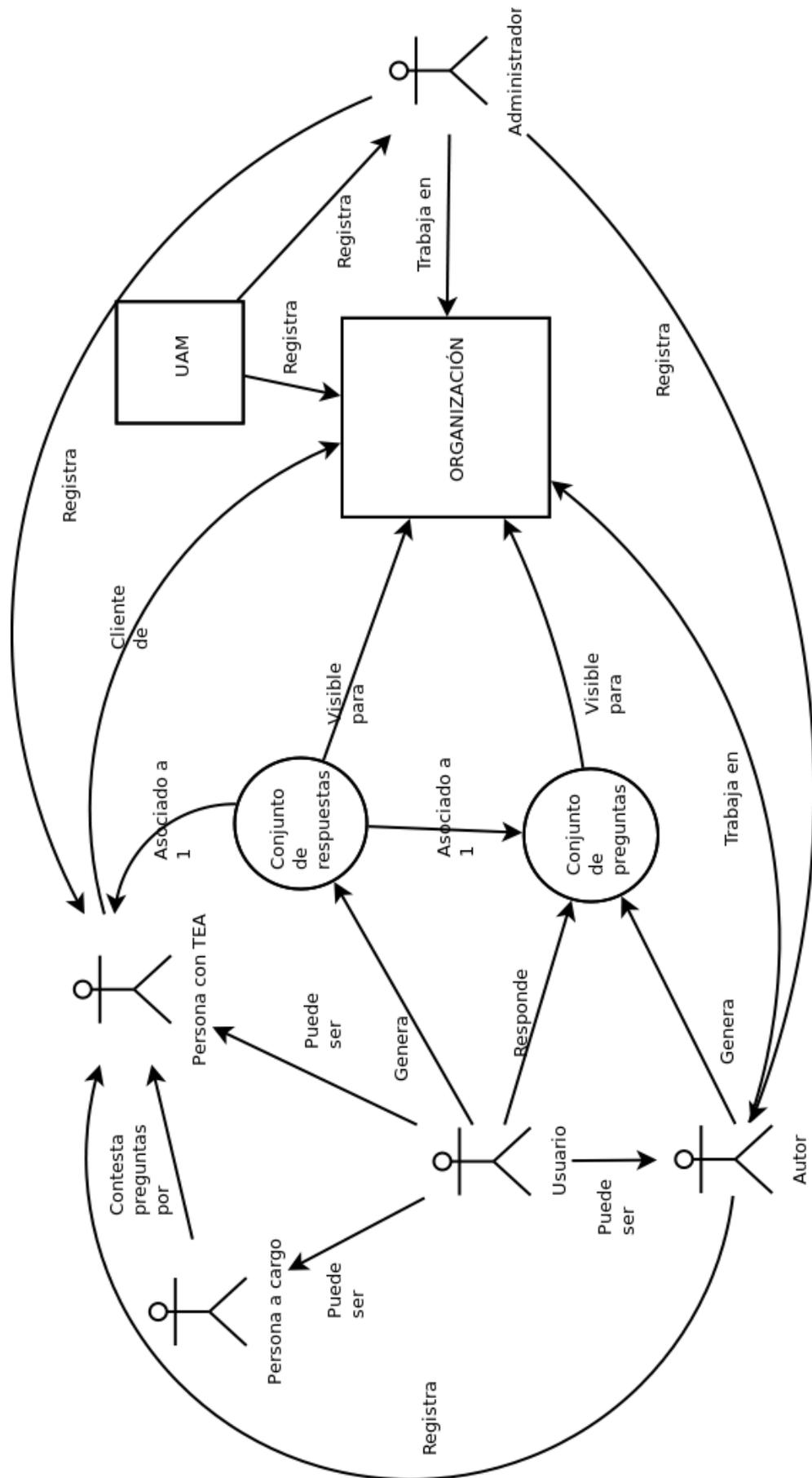


Figura 3.1: Diagrama de casos de uso

ción [15, 23]. Esta herramienta, si bien no es imprescindible, es muy importante porque permite a los autores generar sus propios conjuntos de preguntas sin necesidad de escribir código, lo que permite disponerla a cualquier usuario así como hacerlo de forma mucho más rápida. Esto es importante, porque al ser sencillos de generar podrán generar más, y podrán escoger uno u otro dependiendo del contexto en el que se encuentren los estudiantes o del criterio de los expertos. Por ejemplo, puede tener sentido utilizar un conjunto de preguntas distinto para modelar a un estudiante de mayor edad. Además, los cuestionarios se han planteado como un grafo (en el que cada nodo es una pregunta y cada rama es una transición entre dos preguntas por medio de una respuesta, estando las posibles respuestas contenidas como subnodos del nodo pregunta) y no como una secuencia, lo que permite que se puedan evitar preguntas innecesarias, reduciendo así el tamaño de los cuestionarios y optimizando el tiempo requerido para rellenarlos.

3.3.2. Servidor

El elemento central es el servidor. El servidor mantiene los datos de los usuarios, las preguntas y las respuestas accesibles a los demás elementos que conforman el sistema, de forma que se puedan crear, listar, leer, actualizar y descargar. Este software servidor es el que mantiene la lógica de los usuarios siguiendo lo estipulado en la figura 3.1.

El software servidor se ha diseñado pensando en que el sistema puede llegar a escalar y utilizarse en distintas organizaciones a la vez. Para optimizar costes, el diseño permite mantener los datos de usuarios, cuestionarios y respuestas de múltiples organizaciones en una misma base de datos, siempre evitando que los datos de una organización sean accesibles por otra.

Para dar el poder de gestión a la organización, nace el concepto de *administrador de una organización*. Un administrador de una organización será creado por el administrador del sistema (de la UAM en este caso), y este administrador de organización será el encargado de agregar los clientes (usuarios con TEA) y autores (docentes y expertos). Estos autores podrán a su vez agregar clientes. Esto por ejemplo se adaptaría a la realidad de que un colegio está dividido en cursos y grupos, de forma que en este caso en particular la directora del colegio de educación especial sería la administradora de la organización, y los autores serían los docentes, que podrían introducir los datos de sus alumnos, repartiendo mejor la carga de trabajo.

Además, este software servidor permite probar los cuestionarios simulando la aplicación móvil online en el navegador web de los expertos.

3.3.3. Aplicación móvil

La aplicación móvil tiene una doble función. Por una parte, permite a los estudiantes aprender mediante una aplicación de sumas sencillas adaptada a sus intereses. Por otra parte, permite a los

autores seleccionar un estudiante con el que se va a trabajar y responder al conjunto de preguntas para ese estudiante (o utilizar el último conjunto de respuestas para ese estudiante). Esta funcionalidad está separada, de manera que los estudiantes no sean capaces de cambiar esta configuración fruto de las respuestas.

La herramienta de aprendizaje consiste en una serie de sumas sencillas que son presentadas una a una al usuario con TEA. Se presentan tres opciones, de las que sólo una es la correcta. El usuario con TEA deberá tocar en la correcta para recibir una *recompensa*, que no es más que un estímulo auditivo o visual (según decidan los expertos) que se presenta antes de mostrar la siguiente suma.

3.3.4. Adaptabilidad de la aplicación

Se han tratado los tres elementos principales que conforman la arquitectura, pero una cuestión que todavía no se ha abordado y que no se aprecia en los diagramas es cómo se ha diseñado la interacción entre los cuestionarios y la aplicación, interacción que hará efectiva la adaptabilidad de esta aplicación.

El principal problema que se encuentra es que las preguntas son variables, esa es la razón de ser de la herramienta de autor. Entonces, al ser variables las preguntas, la aplicación no puede tener conocimiento previo de esas mismas preguntas ni de sus respuestas asociadas. Obtener este conocimiento en tiempo de ejecución exclusivamente a partir de las preguntas y las respuestas implicaría probablemente un *parsing* complicado y que no estaría exento de errores.

Para dar solución a este problema, nace el concepto de objetivo. Se ha denominado objetivo a una situación que puede darse o no, y a la que se llega a través de haber contestado a determinadas preguntas de una forma concreta, esto es, habiendo dado una serie concreta de respuestas a una serie de preguntas. Para diseñar esto, tenemos que el conjunto de preguntas tiene definidos una serie de objetivos que forman parte del grafo, y están asociados a una o más respuestas, de modo que un objetivo se activa cuando se contesta con alguna de las respuestas que están asociadas a ese objetivo. Estos objetivos son finitos, estáticos y conocidos previamente por el autor.

La aplicación lleva implementados esos objetivos, de forma que al interpretar el grafo de preguntas es capaz de conocer qué objetivos están activados (es decir, se están dando) para un conjunto de respuestas en particular, y adaptarse en función de esos objetivos.

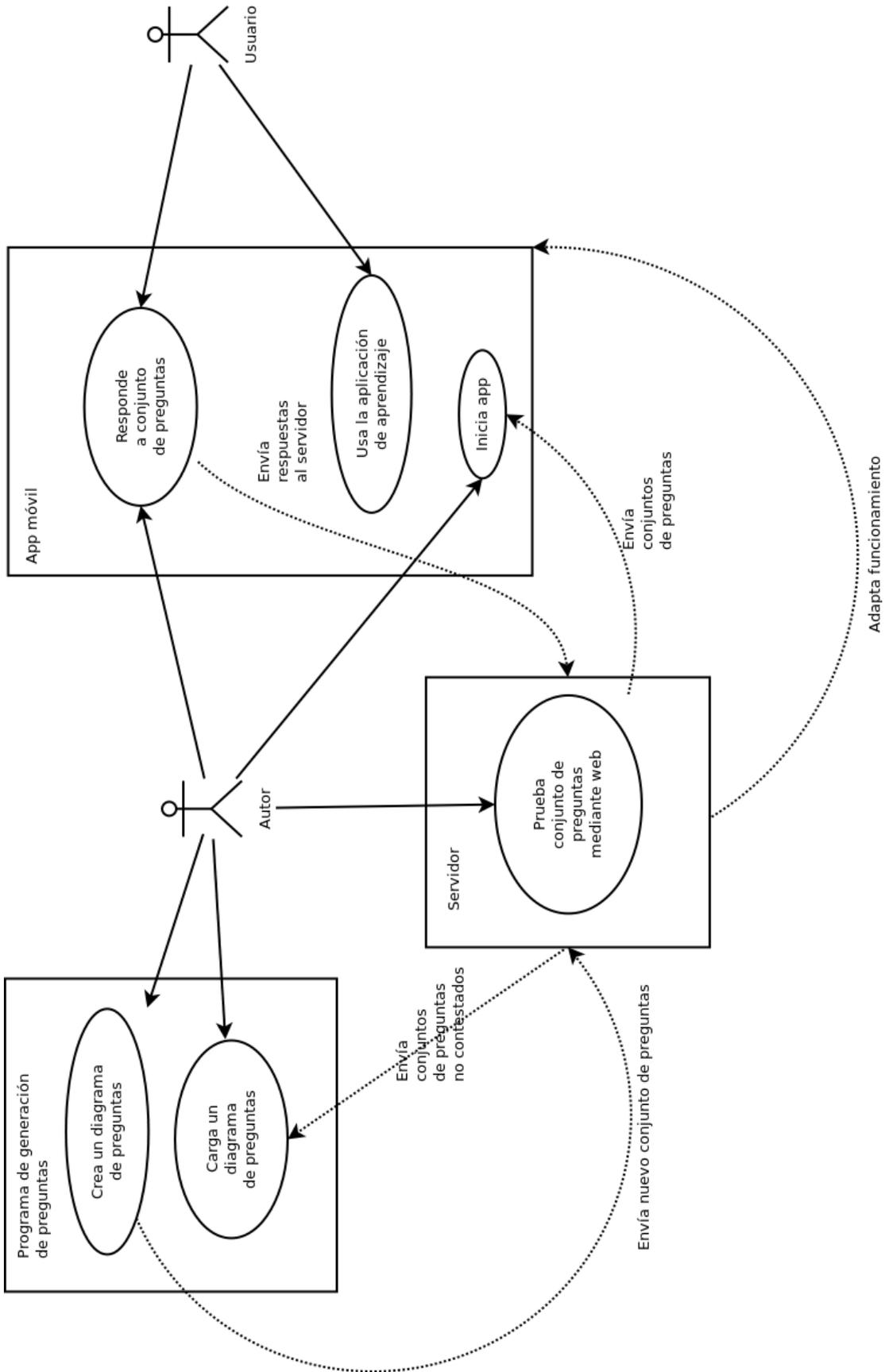


Figura 3.2: Diagrama de casos de uso orientado a la arquitectura

DESARROLLO

A continuación se va a explicar el proceso de desarrollo y los detalles de la implementación del sistema. Se ha seguido una metodología incremental para poder ir validando el sistema según se iba avanzando con el desarrollo.

4.1. Herramienta de autor

Lo primero que se ha empezado a desarrollar ha sido la herramienta de autor, a la que se ha denominado Quartz. Esta herramienta se ha basado en Eclipse, que es el estándar *de facto* en lo que a ingeniería basada en modelos se refiere [37].

La ingeniería basada en modelos ofrece una flexibilidad inestimable a la hora de abordar un desarrollo en el que hay involucrada una herramienta de autor [38].

4.1.1. Meta-modelo de conjuntos de preguntas

Lo primero que se hace es definir el meta-modelo. El meta-modelo es la especificación general a la que se adhieren todos los conjuntos de preguntas en nuestro sistema. Para definir este meta-modelo se ha hecho uso de Epsilon [39], una distribución particular de Eclipse que permite definir meta-modelos de forma gráfica. El meta-modelo queda como se puede ver en la figura 4.1.

Este meta-modelo, aunque lo hemos definido de forma gráfica, es transformado por Epsilon automáticamente a formato Ecore (basado en XML), que puede ser leído por el resto de herramientas de Eclipse.

En la figura se puede observar que podemos tener distintos tipos de preguntas: numéricas, textuales y genéricas. Todas ellas son de opción múltiple, es decir, una pregunta soporta tener varias respuestas, que no necesariamente tienen que ser del mismo tipo (es decir, una pregunta puede tener como posible respuesta a la vez un valor numérico, un valor de texto libre o un valor predefinido en una frase). Este meta-modelo es fácilmente extensible por medio de Epsilon.

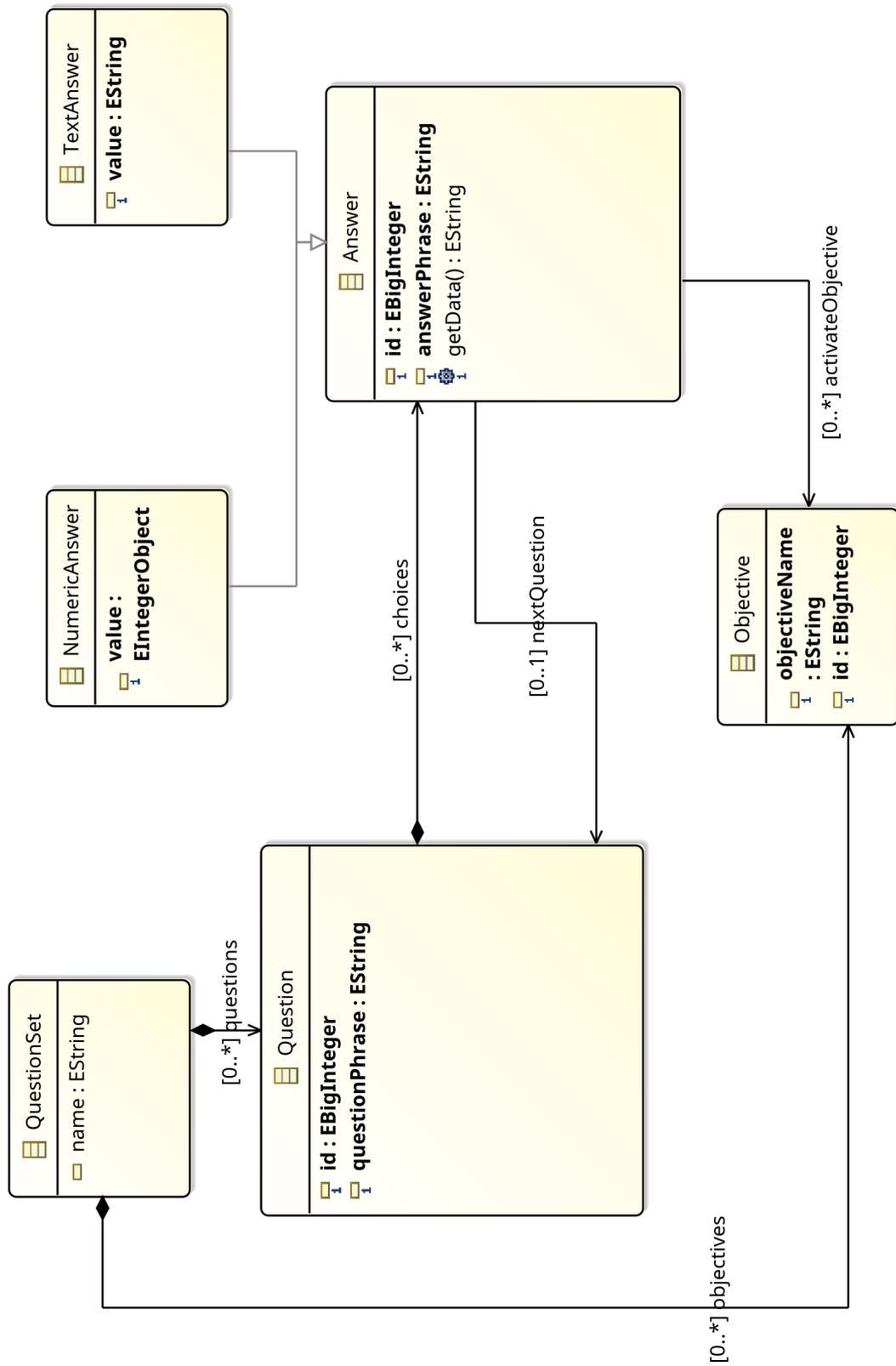


Figura 4.1: Meta-modelo de los conjuntos de preguntas

4.1.2. Editor gráfico de conjuntos de preguntas

Una vez definido el meta-modelo, el siguiente paso es la creación del editor gráfico. Como ya se mencionó en el capítulo de diseño, las preguntas están pensadas para ser modeladas como un grafo, por lo tanto la forma natural de crear conjuntos de preguntas es creando grafos o diagramas.

Es en esta tarea en la que podemos aprovechar la potencia de la ingeniería basada en modelos. Para crear el editor de diagramas utilizaremos otra herramienta de Epsilon, denominada EuGENia [40]. Esta herramienta permite, dado un meta-modelo, generar automáticamente la base de un editor gráfico, que podrá a su vez generar modelos (en nuestro caso conjuntos de preguntas) que estén basados en dicho meta-modelo.

Sin embargo, EuGENia no hace todo el trabajo, lo que genera es un plugin para Eclipse con una interfaz y una funcionalidad básica que posteriormente se ha tenido que retocar para adaptarse a los requisitos.

Esta fase de retocado ha consistido principalmente en conseguir que se asignaran automáticamente identificadores únicos a cada elemento (para lo que ha habido que familiarizarse con el código Java generado por EuGENia) y personalizar el estilo de los grafos en el editor para que los nodos de preguntas y de objetivos no tuvieran la misma forma, y también para que las conexiones entre respuestas y objetivos fueran no dirigidas mientras que las conexiones entre respuestas y preguntas sí que fueran dirigidas.

4.1.3. Transformación a JSON del conjunto de preguntas

Los conjuntos de preguntas generados por el editor de diagramas se guardan en dos ficheros. Uno con extensión `.quartz_diagram` que contiene los datos de posición gráfica de los elementos en el editor, y otro con extensión `.quartz` que contiene los datos del conjunto de preguntas en formato XMI.

El formato XMI es una extensión de XML que permite almacenar modelos en un formato legible. Sin embargo, como se utilizan tecnologías web para implementar tanto el servidor como la aplicación móvil, se decidió realizar una transformación a JSON, que es un formato mucho más manejable que XML para JavaScript.

Para hacer esta transformación, aprovechamos de nuevo la ingeniería basada en modelos usando una herramienta de Epsilon denominada Epsilon Generation Language o EGL para abreviar. EGL permite, mediante la definición de plantillas transformar un modelo a texto, o lo que es lo mismo, generar una representación textual de un modelo basándose en una plantilla.

Era deseable que esta transformación fuera transparente para el usuario, porque se tenía conocimiento previo de que los usuarios finales de la herramienta de autor no conocían el concepto de archivo, por lo que no podrían entender el cambio de formatos ni acordarse de realizar esta transfor-

mación cada vez que se modificara el diagrama. Para hacer esto transparente, se programa un plugin de Eclipse que contenga la plantilla EGL y que automáticamente la ejecute cada vez que se guarde el diagrama. De este modo, cada vez que el usuario hace un cambio en el diagrama, se regenera su representación en JSON.

Hacer esto automático tuvo una serie de implicaciones muy importantes para el desarrollo, porque el editor basado en Eclipse no detectaba los cambios en los archivos a no ser que estuvieran dentro de un proyecto de Eclipse, por lo que hubo que reimplementar toda la funcionalidad referente a la creación de nuevos diagramas.

4.1.4. Sincronización automática con el servidor

Como ya se ha comentado antes, los autores a los que va dirigido esta herramienta requerían la menor interacción posible con aspectos de transmisión de los datos. Es por esto que, para evitar el trasiego de archivos, se vuelve necesario implementar una sincronización automática con el servidor. Cada vez que el usuario guarde el diagrama, automáticamente ese diagrama se sube al servidor.

Para implementar esto, se usó un método similar al de la transformación a JSON. Eclipse permitía detectar los cambios en un archivo de forma que cuando el usuario guardara el diagrama, se subiera al servidor mediante una conexión HTTP. Se programa un plugin de Eclipse que cuando detecta un cambio en un archivo con determinada extensión dentro del proyecto, manda una petición PUT con el archivo a un *endpoint* del servidor, que se encarga de guardarlo en un lugar adecuado.

De igual manera que se permite una carga, también debe permitirse una descarga. Para hacer esto, se reimplementa la carga de archivos programando otro plugin de Eclipse que pida al servidor una lista de diagramas descargables para la organización. Al seleccionar uno de la lista, se descarga y se abre automáticamente.

4.1.5. Acceso al servidor

Toda esta comunicación con el servidor debe seguir cierta lógica. En la fase de diseño se determinó que un mismo servidor podía manejar distintas organizaciones, lo que hace necesario que los autores se autentiquen para poder saber qué conjuntos de preguntas podrán editar (los que hayan sido creados por miembros de su organización).

Para lograr esto desde la herramienta, se programa un plugin de Eclipse que al iniciar la herramienta, hace que aparezca una ventana en la que se debe introducir el nombre de usuario y la contraseña, que se enviarán a un *endpoint* del servidor para conseguir un token. Este token se guarda en el depósito de claves seguro de Eclipse, para evitar a los usuarios el tener que introducir usuario y contraseña cada vez que inicien la herramienta.

4.1.6. Modularidad de la herramienta

La dirección del servidor a la que nos conectamos tiene un carácter variable. Para realizar pruebas utilizamos una dirección, pero en el momento en el que desplegamos el servidor, esta dirección cambia. Para hacer que esto sea sencillo de modificar, se programa un plugin de Eclipse que contiene la dirección del servidor.

La herramienta en sí se ha implementado como un conjunto de plugins de Eclipse que se agrupan en grupos denominados *features* o características. En concreto se han establecido dos características:

- Una característica denominada QuartzFeature, que contiene el meta-modelo, el editor de diagramas, las plantillas EGL y las funciones de carga y descarga automáticas.
- Una característica denominada QuartzAlenta.branding que contiene el plugin con la dirección del servidor y una pantalla de carga personalizada con el logo del colegio.

Estas dos características se han agrupado en lo que se denomina un *producto* de Eclipse, que permite generar una instancia de Eclipse con las características deseadas y para la plataforma elegida (que puede ser Windows, Linux o MacOS), de forma que pueda ejecutarse como si se tratara de un programa cualquiera.

Si se desea cambiar algún comportamiento de la herramienta o desactivar alguna opción, no es necesario reescribir el código de nuevo ni crear un nuevo proyecto, basta con quitar el plugin que no queramos de la *feature* o agregar una nueva *feature*. Incluso podríamos distribuir el programa bajo otro nombre para otra organización creando una nueva característica *branding* y creando otro producto de Eclipse con la nueva característica y la característica QuartzFeature.

4.1.7. Mejoras en la interfaz

Al ser la herramienta una serie de plugins que se ejecutan sobre Eclipse, y ser Eclipse una herramienta de desarrollo para programadores, la herramienta hereda algunas funcionalidades, menús y características de Eclipse que no son necesarias y que complican la interfaz de cara al usuario.

Por eso se ha programado un plugin de Eclipse que al ejecutar la herramienta oculta automáticamente todos los menús innecesarios, dejando exclusivamente un menú de control con los botones para cargar y descargar los conjuntos de preguntas.

4.2. Servidor

En este trabajo se ha desarrollado un servidor web por medio de Django, un *framework* web en Python que permite agilizar el proceso de desarrollo [41]. Django sigue el patrón modelo-vista-controlador, siendo el modelo los datos en una base de datos, la vista el conjunto de la interfaz web y de *endpoints* de la API REST y el controlador la lógica de negocio que conecta esos dos elementos.

El servidor web tiene una doble funcionalidad en el sistema. La primera es la de mostrar una interfaz de administración para los usuarios (que no debe confundirse con la interfaz de administración por defecto que proporciona Django), mientras que la segunda es la de proporcionar una API REST con el objetivo de que la herramienta de autor y la aplicación móvil puedan operar ubicuamente.

4.2.1. Interfaz de administración

La interfaz de administración permite a los usuarios, tanto administradores como autores, gestionar clientes para su organización. La interfaz soporta que dicha organización sea creada previamente por el superusuario (que sería el administrador del servidor en el que se desplegara).

Aparte, la interfaz permite a los autores probar los conjuntos de preguntas que desarrollen usando la herramienta. Esto es factible al realizarse la subida del conjunto de preguntas de forma transparente, de manera que los autores sólo tienen que seleccionar el conjunto de preguntas que quieren probar, y se les presenta una interfaz similar a la de un dispositivo móvil con la que pueden ver qué objetivos se van activando según van respondiendo preguntas.

4.2.2. API REST

Se implementa una API REST usando una biblioteca de Django denominada Django Rest Framework. Esta biblioteca permite simplificar el desarrollo y evitar posibles errores dado que abstrae mucha funcionalidad como por ejemplo la autenticación por medio de *tokens*. Aunque esta biblioteca simplifica mucho el desarrollo y la depuración de las APIs que se desarrollan con ella, tiene cierto coste de aprendizaje, al introducir conceptos nuevos externos a Django como son los *serializadores*, los *enrutadores* y los *adaptadores*.

4.2.3. Patrón Modelo-Vista-Controlador

En Django, el *framework* utilizado para el desarrollo del software servidor fomenta el uso del patrón modelo-vista-controlador o MVC. El proceso de desarrollo es el siguiente:

Modelo

Primero se crea el modelo. El modelo se define como una serie de clases de Python, que se trasladan a una base de datos relacional (en nuestro caso se ha utilizado SQLite por su facilidad de configuración) mediante un proceso que en Django se denomina *migración*. Nuestro modelo se compone de las siguientes clases:

- **Organization:** Las organizaciones son conjuntos de datos que están aislados de otros conjuntos de datos, y además tienen un administrador que las dirige y una serie de usuarios, que pueden ser autores y clientes.
- **Question:** Los conjuntos de preguntas tienen un nombre, están asociados a una organización en particular y se componen de un fichero de diagrama con la posición visual de cada elemento en la herramienta de autor, un fichero modelo que es el que tiene los datos del conjunto de preguntas en formato XML, y un fichero JSON que es una representación del modelo en formato JSON.
- **Person:** Las personas son datos adicionales sobre la persona que hay detrás de cada usuario. En esta clase se especifica el rol de la persona en el sistema, el género, la organización a la que pertenece, la fecha de nacimiento y la foto de perfil. Esta clase está asociada con la clase `User` proporcionada por Django, que es la que contiene los datos de autenticación.
- **Answer:** Los conjuntos de respuestas tienen una fecha de creación y están asociados a un conjunto de respuestas y a un usuario en particular, y contienen una cadena de caracteres con los identificadores de las respuestas seleccionadas separadas por comas.

Vista-Controlador

Una vez creado el modelo, que puede irse modificando con el paso del tiempo gracias al potente sistema de migraciones de Django, se crean lo que en Django se denominan vistas. Una vista no es más que una URL que se puede visitar y que muestra algo. En Django una vista es o bien una función que devuelve una serie de datos, o una clase. En este trabajo hemos utilizado funciones para la interfaz de administración, y clases para la API REST (que también se considera como un conjunto de vistas en Django). Sin embargo, estas vistas no son vistas en el patrón MVC, si no que corresponden con el controlador, porque contienen la lógica en Python que une el modelo con lo que ve el usuario.

Para la interfaz de administración, entran en juego lo que realmente serían las vistas del patrón MVC, que es lo que en Django se denomina plantilla. Una plantilla es un fichero HTML con unas etiquetas de Django especiales embebidas que tienen una funcionalidad similar a PHP. Para simplificar el desarrollo se ha hecho uso de Bootstrap para dar un estilo profesional y cuidado a la interfaz.

La API REST, como se ha comentado previamente, asume la función de controlador y actúa como enlace entre el modelo y dos sistemas externos, uno la aplicación móvil, y otro la herramienta de autor. Para mantener la seguridad del sistema, es requisito que tanto la aplicación como la herramienta de autor dispongan de un *token* para autenticarse contra la API. En este caso el *token* es estático, es decir, cada usuario lleva asociado un *token* que no varía durante el tiempo de vida del sistema. Esto permite simplificar mucho ciertos aspectos de implementación de la aplicación y de la herramienta de autor, que requieren de un único procedimiento de *login* porque el token se puede almacenar localmente de

forma segura en el caso de la herramienta de autor, y en el caso de la aplicación no se requiere de *login* al mantener un usuario específico para la aplicación (que asume el rol de autor), y por lo tanto poder conocer de antemano (en tiempo de desarrollo) el *token* asignado.

La API REST, además, se apoya en una biblioteca de Django denominada Django Rest Framework (DRF), que permite definir fácilmente *endpoints* CRUD (CReate Update Delete). Fácilmente en el sentido de que es una biblioteca muy expresiva y en pocas líneas de código se pueden definir APIs muy potentes, a costa de agregar complejidad al sistema y de requerir una gran inversión de tiempo en el estudio previo de DRF.

4.2.4. Modularidad del software servidor

El software servidor está implementado como una *app Django*. En Django, una *app* es un código que puede utilizarse desde otra aplicación, pudiendo convivir varias aplicaciones a la vez (de hecho DRF es también una *app Django*), y siendo posible cambiar ciertos aspectos de la misma por medio de *sobreescritura*.

Sobreescritura en Django es un concepto análogo a la sobreescritura de métodos en Java, sólo que en Django es posible sobreescribir cualquier cosa a excepción del modelo. De esta forma, es posible cambiar el estilo de la interfaz de administración sobreescribiendo las plantillas, o cambiar el comportamiento de la API REST sobreescribiendo la URL correspondiente y agregándole otra vista distinta.

4.3. Aplicación móvil

Una vez implementado el servidor con su API, el último paso es el de implementar la aplicación móvil. Para implementar la aplicación se ha hecho uso de un *framework* denominado Apache Cordova [42]. Apache Cordova permite crear aplicaciones que pueden ser compiladas para múltiples plataformas evitando la complejidad de programarlas nativamente, y permitiendo usar HTML, CSS y JavaScript para implementar la aplicación.

4.3.1. Interfaz de la aplicación

La interfaz de la aplicación de sumas sencillas, a la que se ha denominado *Sumarte* está compuesta por las siguientes pantallas:

- Menú principal: El menú principal contiene un botón con forma de candado que dará acceso a la funcionalidad restringida (es decir, la funcionalidad destinada a los autores). Contiene un botón en el centro que al ser pulsado dará inicio a la aplicación de sumas.
- Menú de selección de usuario: Al pulsar el botón candado del menú principal, aparece un diálogo en el que se pide contestar a una pregunta. Como muchas persona con TEA tienen dificultades para la comunicación, esto hará muy complicado que consigan acceso a esta interfaz, mientras que evitará a los autores el tener que memorizar una contraseña. Una vez contestada la pregunta, aparecerá el menú de selección de usuario, en el que aparecen las fotos de perfil de los estudiantes de la organización junto a su nombre y apellidos, mostradas por orden alfabético.
- Menú de selección de pregunta: Al seleccionar un estudiante en el menú, aparece otro menú con los conjuntos de preguntas de la organización.
- Menú de contestación de pregunta: Al seleccionar una pregunta en el menú, si existe una respuesta para ese usuario y ese conjunto de preguntas, se muestra un diálogo que informa de esta situación, y da la opción de volver a contestar al conjunto de preguntas o, por el contrario, utilizar la última respuesta vinculada a ese estudiante para ese conjunto.
- Menú de aprendizaje: Una vez contestadas las preguntas, se vuelve automáticamente al menú principal. En este menú principal hay un botón que al pulsarlo hace que comience la aplicación de sumas, adaptado con las respuestas escogidas en acciones anteriores (el estudiante y conjunto de respuestas contestado se guarda en la memoria del dispositivo para que no sea necesario volver a seleccionar un estudiante y un conjunto en cada ejecución de la aplicación).

4.3.2. Mecánica del sistema de aprendizaje

La mecánica del sistema de aprendizaje es sencilla. Se presenta al estudiante una suma sencilla y tres posibles opciones, de las que sólo una es correcta. El estudiante debe tocar en la opción correcta para pasar a la siguiente suma. El proceso acaba cuando lo decida la persona a cargo del estudiante.

4.3.3. Adaptabilidad del sistema

A pesar de la sencillez de la mecánica, el adaptar esta mecánica a partir del sistema de objetivos presentado en el capítulo de diseño no es una tarea trivial. Para poder explicar lo que se ha hecho, se va a explicar en detalle cuáles han sido las adaptaciones implementadas, para después centrarse en cómo se hace efectiva esta adaptación. Las adaptaciones implementadas han sido:

- Mostrar instrucciones
- Mostrar una recompensa sonora al acertar
- Mostrar una recompensa visual al acertar
- Mostrar el texto en mayúsculas
- Mostrar el fondo de la aplicación en rojo
- Mostrar el fondo de la aplicación en azul
- Mostrar el fondo de la aplicación en verde
- Mostrar el fondo de la aplicación en blanco
- Mostrar el fondo de la aplicación en rosa
- Mostrar el fondo de la aplicación en amarillo
- Mostrar dedos de las manos acompañando a los números, de forma que el número de dedos sea igual al número que acompañan.
- Mostrar pelotas acompañando a los números, de forma que el número de pelotas sea igual al número que acompañan.
- Mostrar coches acompañando a los números, de forma que el número de coches sea igual al número que acompañan.
- Mostrar dinosaurios acompañando a los números, de forma que el número de dinosaurios sea igual al número que acompañan.
- Mostrar planetas acompañando a los números, de forma que el número de planetas sea igual al número que acompañan.
- Hacer que estos objetos sean siempre el mismo o que aleatoriamente se varíe la imagen (por ejemplo, si hay tres dinosaurios, que no sean los tres dinosaurios iguales).
- Hacer que estos objetos se muestren en fila o no.

Es decir, se trata de adaptaciones en forma de presentación de la información, mientras que la mecánica siempre es la misma. Las adaptaciones se han implementado mediante funciones. Cuando el sistema interpreta las respuestas para ver qué objetivos están activos, por cada objetivo activo ejecuta una función que está asociada a ese objetivo y el resultado de esa función es que se produce un cambio en el sistema, que dura hasta que se empieza una nueva interacción con una configuración distinta.

4.3.4. Idempotencia y adaptatividad del sistema

El sistema implementado en este trabajo es adaptable, es decir, se puede modificar su interfaz para adaptarla a distintos tipos de usuarios, que se han modelado previamente mediante la contestación a un determinado conjunto de respuestas y, en última instancia mediante la activación de ciertos objetivos.

Sin embargo, las funciones que se ejecutan para llevar a cabo estos objetivos se han implementado de forma que sean idempotentes, esto es, que aunque se ejecuten varias veces, el estado del sistema no varíe con esas sucesivas ejecuciones de esa misma función. Esto permite ejecutar estas funciones en mitad del proceso de aprendizaje, porque el éxito de su ejecución no depende de ejecuciones (o no-ejecuciones) previas.

Si podemos ejecutar las funciones a mitad del proceso de aprendizaje y, por lo tanto, cambiar el estado del sistema a mitad de la interacción, tenemos que es posible hacer que el sistema sea adaptativo, es decir, que en función de determinados parámetros (como por ejemplo, el desempeño del usuario en el sistema) este se vaya adaptando a sus necesidades. Esto, aunque no se ha implementado para este trabajo por el alcance del mismo, convierte la interacción, aparte de en adaptable, en *potencialmente adaptativa*.

INTEGRACIÓN, PRUEBAS Y RESULTADOS

Una vez desarrollado el sistema, se procede a evaluar hasta que punto se han cumplido los requisitos. La forma evidente de hacerlo es realizando pruebas con usuarios, pero dada la situación actual por la incidencia de la COVID-19, ha sido imposible realizar pruebas con usuarios, tal y como estaba inicialmente previsto.

5.1. Trazabilidad de requisitos

Aún así, podemos realizar una serie de pruebas sin usuarios con el fin de determinar el nivel de cumplimiento de los requisitos en el sistema final. Para hacer esto, vamos a inspeccionar los requisitos funcionales que se han implementado en el sistema final:

RF-1: Capacidad para poder adaptar sesiones de aprendizaje a las necesidades particulares de cada estudiante.

Se trata del requisito principal, que se cumple mediante el sistema en su conjunto. Las sesiones de aprendizaje se adaptan modelando al usuario, contestando a una serie de preguntas que pueden definir los expertos mediante una herramienta de autor, pudiendo coexistir distintas formas de modelar al usuario en el sistema (distintos conjuntos de preguntas). La adaptación se hace efectiva mediante el sistema de objetivos descrito en el capítulo de desarrollo, esto es, la aplicación interpreta el conjunto de preguntas y las respuestas dadas para un usuario concreto y de ahí se averigua qué objetivos deben estar activados.

RF-2: Capacidad para poder agregar usuarios al sistema.

Se cumple este requisito mediante la interfaz de administración del servidor mostrada en la figura 5.1.

RF-3: Capacidad para poder agregar cuestionarios de forma sencilla, que luego podrán ser respondidos de forma ubicua.

Se cumple este requisito mediante la herramienta de autor, trabajando en sincronía con el servidor.

Quartz Salir

Resumen
Organizaciones
Usuarios
Preguntas

Mi Perfil

Nuevo usuario de ATest2

Nombre de usuario

Contraseña

Correo electrónico

Nombre de pila

Apellido

Fecha de nacimiento

Género

Rol

Figura 5.1: Página web de creación de nuevo usuario

Para crear un cuestionario sólo es necesario pulsar sobre el botón de nuevo diagrama (figura 5.2) y escribir el nombre que se le quiere dar al nuevo diagrama (figura 5.3).

La herramienta de autor presenta una serie de herramientas que permiten a los usuarios autores posicionar en un lienzo las preguntas y las respuestas, tal y como se muestra en la figura 5.4.

Estos conjuntos de preguntas o cuestionarios se guardan automáticamente en el servidor, y podemos acceder a un listado de las mismas a través de la web (figura 5.5) y la aplicación (figura 5.6).

RF-4: Capacidad para poder definir qué adaptaciones se aplicarán en la sesión a partir de las respuestas a esos cuestionarios.

Se cumple este requisito mediante el sistema de objetivos que se pueden agregar a los diagramas.

RF-5: Capacidad para que la persona que los define o un tercero pueda probar los cuestionarios y ver qué adaptaciones se aplicarían dadas ciertas respuestas.

Se cumple este requisito mediante la interfaz de administración web del servidor, que proporciona una interfaz especial mediante la que se puede probar un cuestionario previamente seleccionado (figura 5.7).

RF-6: Capacidad para poder seleccionar tanto el usuario con TEA como el cuestionario a los que se va a asociar una respuesta.

Se cumple este requisito mediante la interfaz restringida de la aplicación, a la que pueden acceder

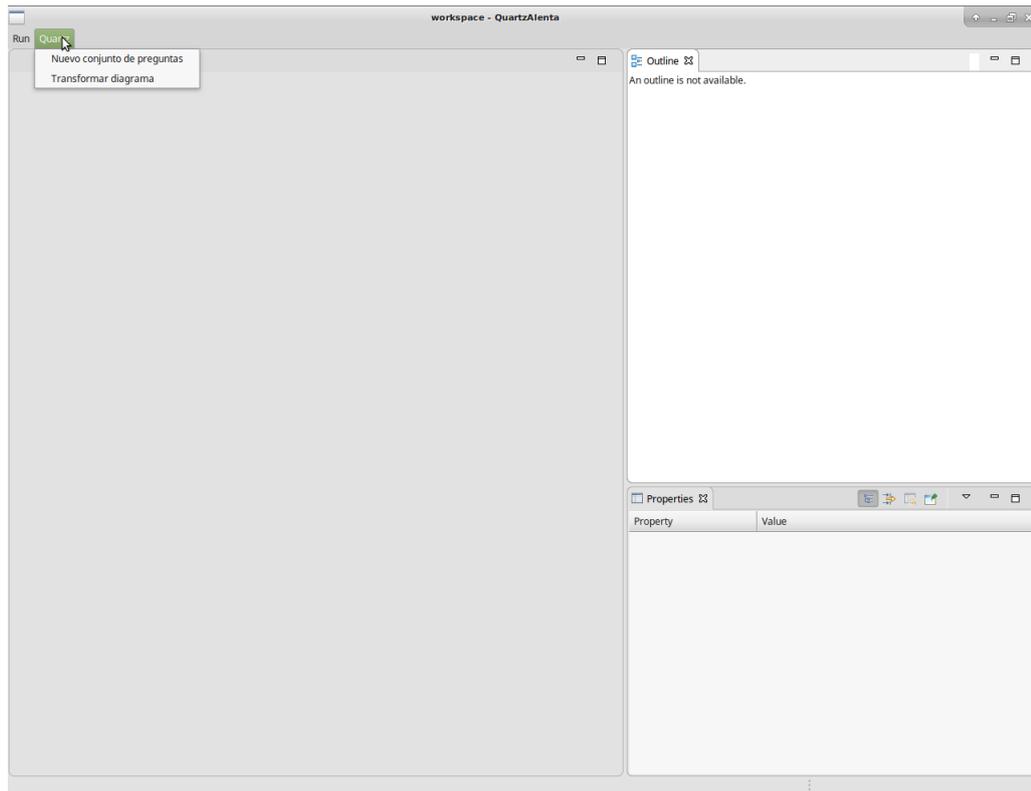


Figura 5.2: Interfaz de la herramienta de autor

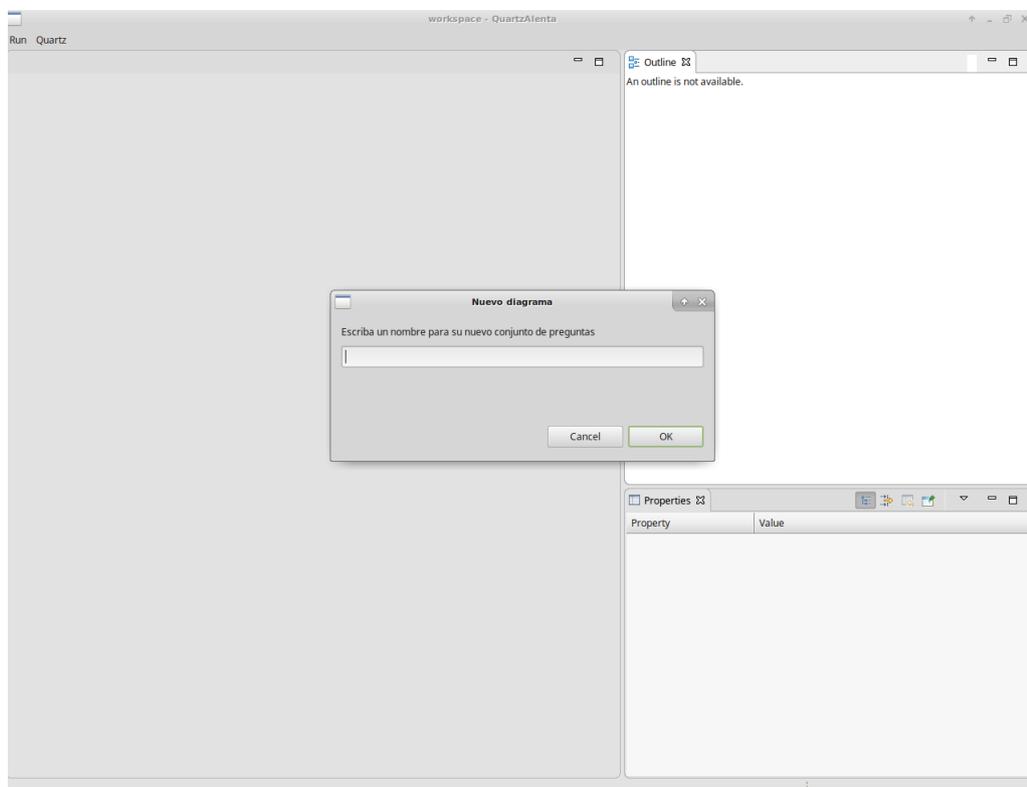


Figura 5.3: Creando un nuevo conjunto de preguntas

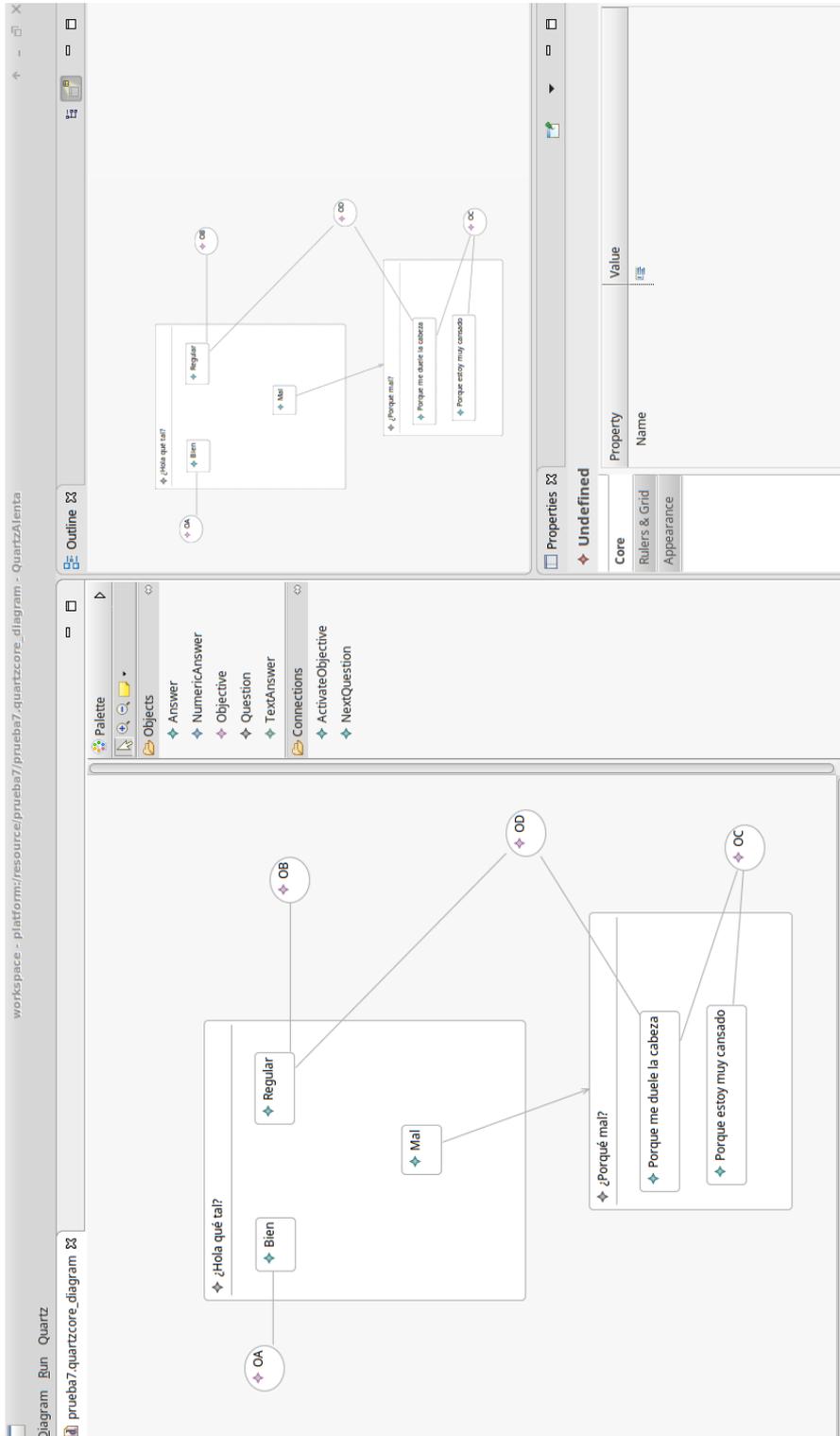


Figura 5.4: Editando un conjunto de preguntas

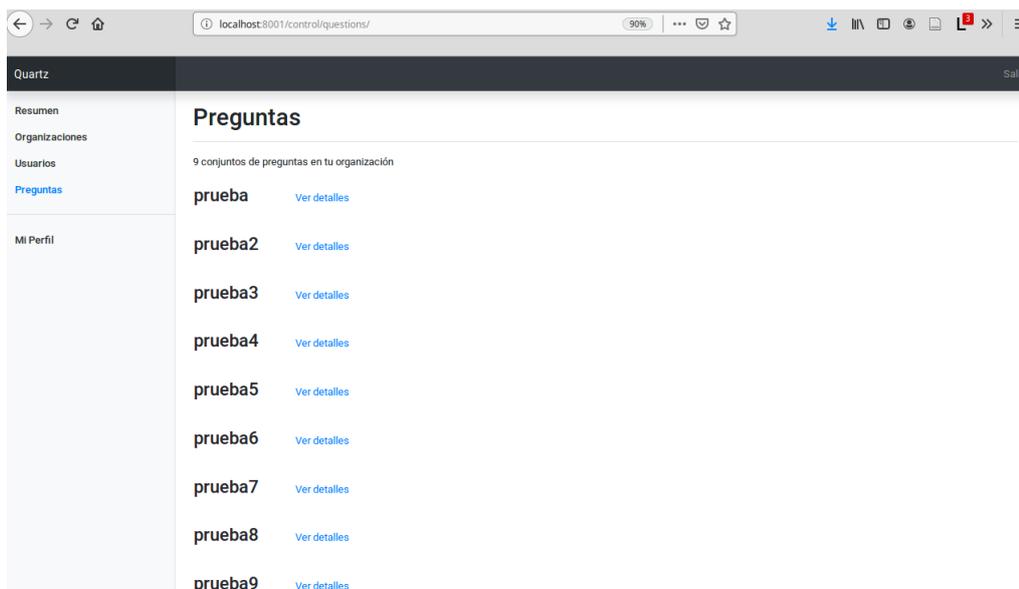


Figura 5.5: Accediendo al listado de preguntas desde la web

personas sin TEA mediante una pregunta de seguridad. Al iniciar la aplicación, se presenta al usuario con un menú principal (figura 5.8)

RF-7: Capacidad para poder ejecutar una sesión de aprendizaje que consista en resolver sumas sencillas seleccionando el usuario con TEA al que se le va a aplicar y el estado en el que se encuentra, que ha sido previamente modelado mediante un conjunto de respuestas a un cuestionario.

Se cumple este requisito mediante el sistema de aprendizaje mediante sumas sencillas, que permite seleccionar un usuario en concreto de una lista antes de proceder a contestar un cuestionario para ese usuario, tal y como se muestra en la figura 5.9.

RF-8: Capacidad de dar instrucciones en texto o no darlas.

Se cumple este requisito en el sistema de aprendizaje mediante sumas sencillas, proporcionando un objetivo *instrucciones* que, cuando se activa, muestra un texto con instrucciones (figura 5.10).

RF-9: Capacidad de dar las recompensas en formato animado, audio, animada y audio a la vez, o ninguna.

Se cumple este requisito en el sistema de aprendizaje mediante sumas sencillas, proporcionando los objetivos *recompensa-animada* y *recompensa-audio*. La recompensa animada consiste en un símbolo que parpadea una vez antes de pasar a la siguiente pregunta, tal y como se muestra en la figura 5.11.

RF-10: Capacidad de que las letras sean todas en mayúscula o no.

Se cumple este requisito en el sistema de aprendizaje mediante sumas sencillas a través del objetivo *mayúsculas*, de forma que cuando está activado se muestran las instrucciones en mayúscula (figura

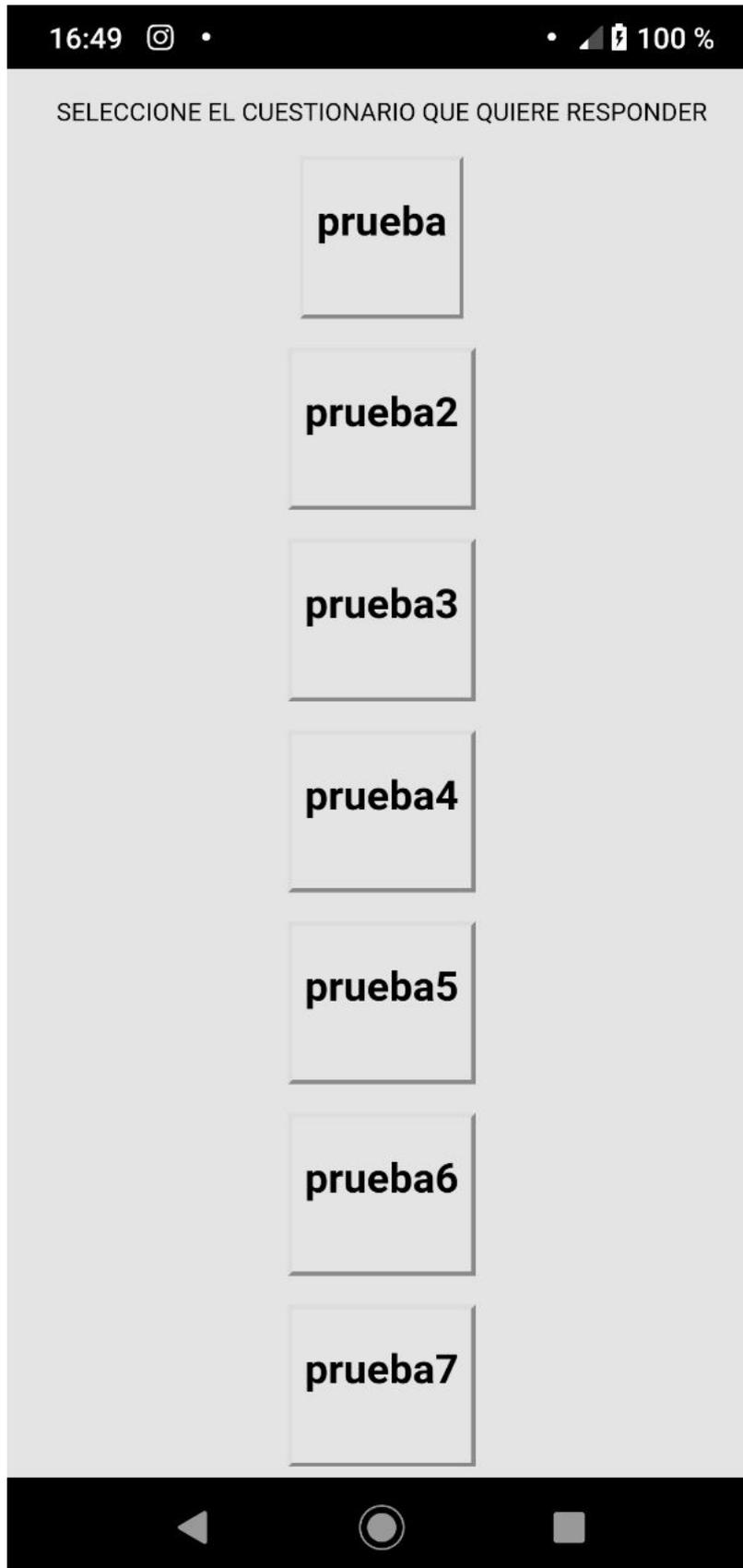


Figura 5.6: Accediendo al listado de preguntas desde la app

The screenshot shows a web browser window with the URL `localhost:8001/control/questions/details/18`. The page title is "Detalles del conjunto de preguntas prueba9". The navigation menu includes "Resumen", "Organizaciones", "Usuarios", "Preguntas" (highlighted), and "Mi Perfil".

The main content area features a tablet simulation displaying a question: "Quieres una recompensa?" with three options: "Si, animada", "Si, sonora", and "Si, animada y sonora". A "Reiniciar" button is located to the right of the tablet.

Below the tablet, there are two tables:

Preguntas y respuestas en esta sesión:

Pregunta	Respuesta
Sabes algo de sumas sencillas?	Si
Que color te gusta mas?	Rojo
Preferes en mayusculas o en minusculas?	MAYUSCULAS
Quieres acompañar los numeros con objetos? SI	Dedos
Quieres que sean distintos, o todos iguales?	Todos iguales
En orden o en desorden?	Orden

Objetivos activados en esta sesión:

Objetivo	Activado
fondo-rojo	✓
fondo-verde	
fondo-rosa	
fondo-azul	
fondo-blanco	
fondo-amarillo	
instrucciones	
mayusculas	✓
objetos-dedos	✓
objetos-bolas	
objetos-planetas	
objetos-coches	
objetos-dinosaurios	
objetos-distintos	
objetos-desordenados	
recompensa-animada	
recompensa-audio	

Figura 5.7: Simulando preguntas desde la interfaz web



Figura 5.8: Menú principal de la aplicación

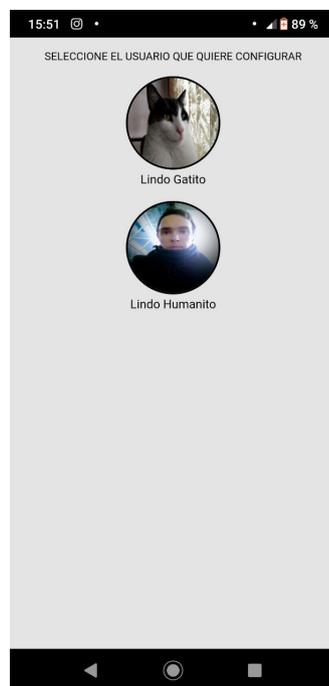


Figura 5.9: Listado de usuarios en la aplicación

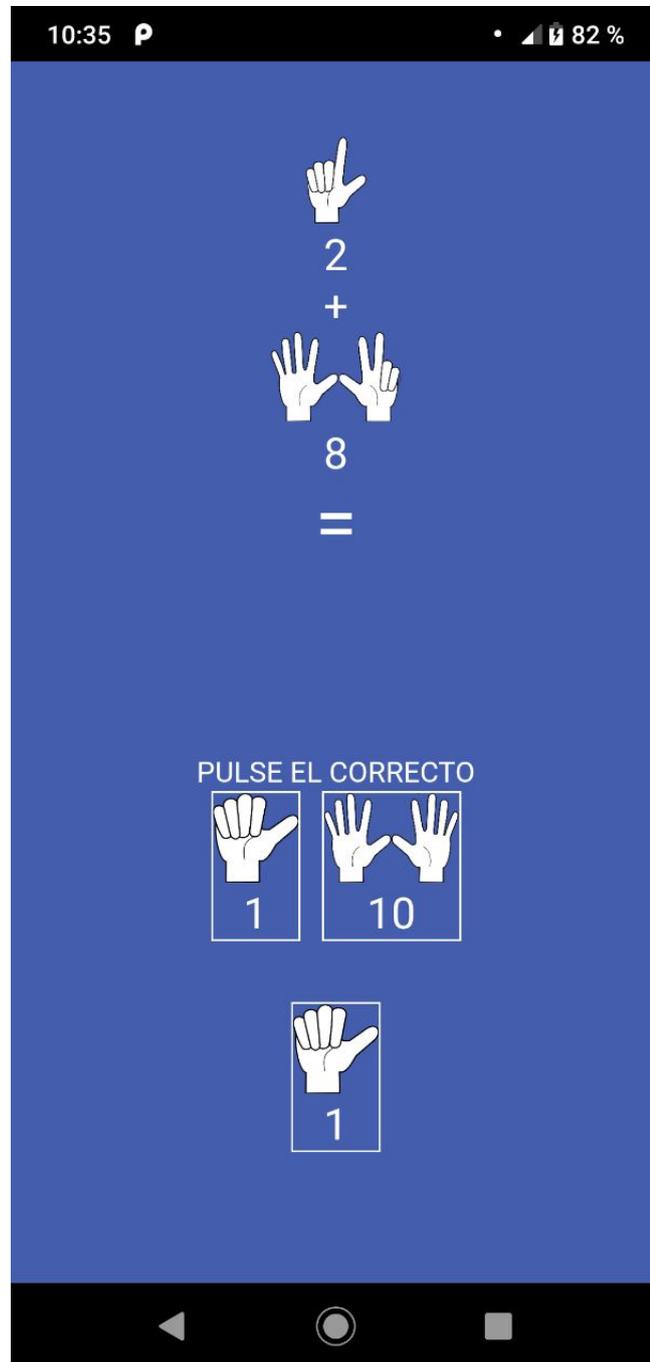


Figura 5.10: Instrucciones en el sistema



Figura 5.11: Recompensa visual del sistema

5.10) y cuando no está activado se muestran las instrucciones en minúscula (figura 5.12).

RF-11: Capacidad de que el color de fondo pueda ser personalizado.

Se cumple este requisito en el sistema de aprendizaje mediante sumas sencillas proporcionando los objetivos *fondo-rojo*, *fondo-azul*, *fondo-verde*, *fondo-blanco*, *fondo-rosa* y *fondo-amarillo*, que de activarse quedarían como se muestra en la figura 5.13.

RF-12: Capacidad de que los números vayan acompañados de objetos o no.

Se cumple este requisito en el sistema de aprendizaje mediante sumas sencillas con los objetivos *objetos-dedos*, *objetos-bolas*, *objetos-coches*, *objetos-dinosaurios* y *objetos-planetas*.

RF-13: En caso de que haya objetos, capacidad de poder definir el tipo de los objetos, que pueden ser: dedos, bolas, coches, dinosaurios o planetas, que de activarse quedarían como se muestra en la figura 5.13.

Se cumple este requisito por medio de los objetivos mencionados en el RF-12.

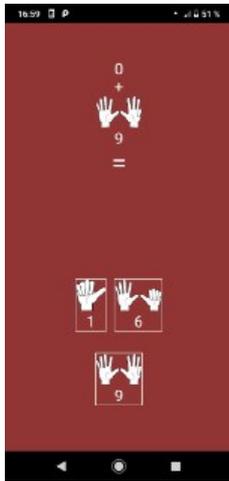
RF-14: Capacidad de que siendo los objetos de una misma categoría sean todos los objetos iguales o no.

Se cumple este requisito por medio del objetivo *objetos-distintos*. En caso de estar desactivado, todos los objetos serán iguales, como se puede comprobar en la figura 5.14.

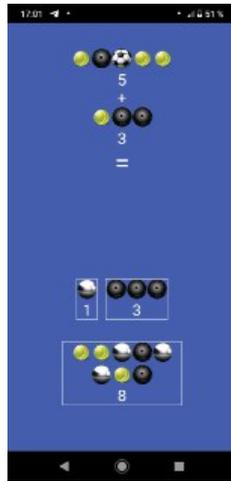
RF-15: Capacidad de que los objetos se muestren en orden o no.



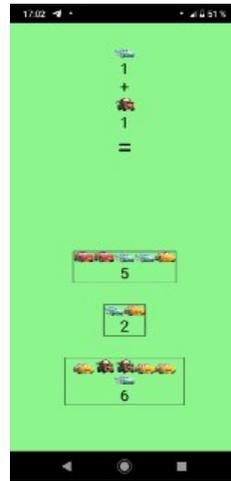
Figura 5.12: Instrucciones en minúscula en el sistema



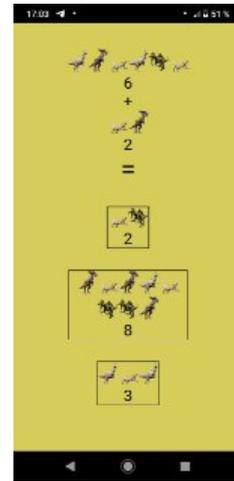
(a) Fondo rojo y dedos



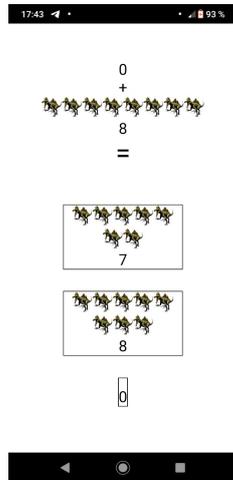
(b) Fondo azul y pelotas



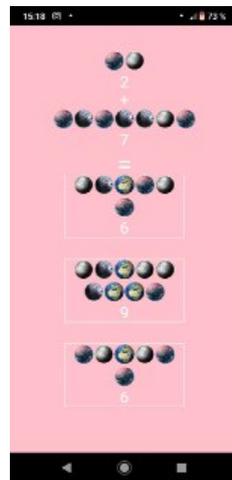
(c) Fondo verde y coches



(d) Fondo amarillo y dinosaurios



(e) Fondo blanco y dinosaurios



(f) Fondo rosa y planetas

Figura 5.13: En estas figuras se muestran diferentes combinaciones de colores y objetos, siendo independientes los colores del fondo con los objetos, y pudiendo darse otras combinaciones.

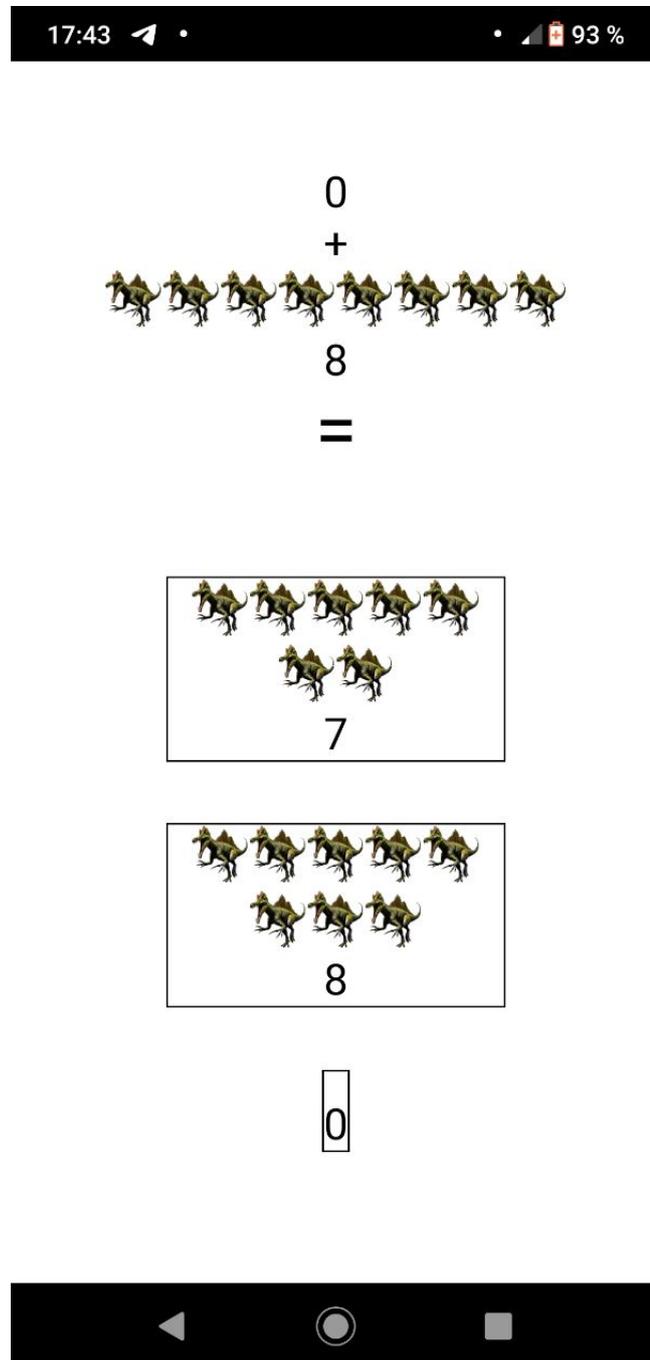


Figura 5.14: Sistema con todos los objetos iguales

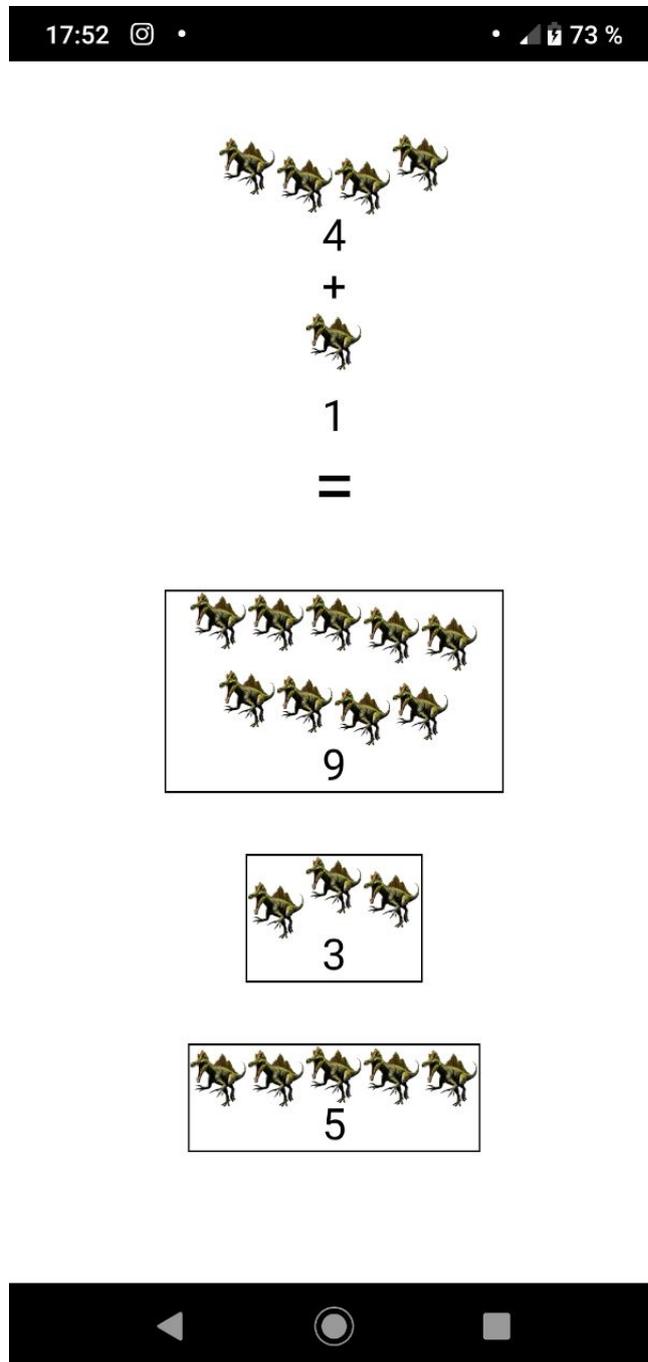


Figura 5.15: Sistema con todos los objetos desordenados

Se cumple este requisito por medio del objetivo *objetos-desordenados*, como se puede comprobar en la figura 5.15.

5.2. Evaluación heurística de usabilidad

Para poder suplir la falta de pruebas con usuarios reales, sobrevenida por las circunstancias actuales, se ha realizado una evaluación heurística de usabilidad. Como se trata de un software que va a ser utilizado por expertos de un campo concreto, se han usado los Principios Heurísticos para el Diseño Centrado en el Usuario de Mayhew [43], que consiste en evaluar los siguientes puntos:

- 1.– Compatibilidad del producto con los procesos del usuario: Se debe determinar si el producto se adapta al uso que el usuario hará de él.
- 2.– Consistencia: Se debe determinar si el producto no es vulnerable a errores.
- 3.– Familiaridad: Se debe determinar si el usuario está familiarizado con un sistema similar.
- 4.– Simplicidad: Se debe determinar si el sistema es simple.
- 5.– Manipulación directa: Se debe determinar si el usuario maneja directamente los datos del sistema.
- 6.– Control: Se debe determinar si el usuario tiene en todo momento control del sistema.
- 7.– WYSWYG: Se debe determinar si el usuario trabaja con el aspecto real del documento.
- 8.– Flexibilidad: Se debe determinar si el sistema es flexible, puede ser usado por cualquier tipo de usuarios.
- 9.– Sensibilidad y feedback: Se debe determinar si el sistema interactúa con el usuario, por ejemplo da mensajes de error cuando algo falla.
- 10.– Tecnología invisible: Se debe determinar si la tecnología subyacente utilizada se mantiene invisible al usuario.
- 11.– Protección: Se debe determinar si los datos del usuario quedan a salvo de posibles intrusos.
- 12.– Facilidad de aprendizaje: Se debe determinar si el sistema es sencillo de aprender o posee funcionalidades muy complejas.

Dados los puntos anteriores, el resultado de la evaluación es que el sistema es compatible con los procesos de un colegio de educación especial, al haber tenido en cuenta la operativa determinada por uno de ellos desde fases tempranas de diseño.

Durante la fase de desarrollo no se han detectado errores no recuperables, si bien la interfaz de la herramienta de autor puede ser lenta al cargar. Esto es debido a que está basada en Eclipse, que es un programa que produce un gran impacto en los recursos del sistema en el que se ejecuta.

Al estar basada en la creación de diagramas, y ser esta la principal acción que llevarán a cabo los usuarios, creemos que la acción de dibujar en sí resultará familiar a los usuarios, pero podría ser que algunas particularidades del editor fueran ajenas a los usuarios.

Se ha puesto un especial hincapié en que el sistema sea simple, pero la complejidad de gestionar los datos de los usuarios y de crear elementos complejos como son cuestionarios hace que el sistema

tenga ciertas limitaciones en este sentido.

Por ejemplo, el usuario no manipula directamente los datos de los diagramas, que al final son ficheros de texto, sino que se hace una transformación de una visualización en forma de diagrama a varios formatos. Estas transformaciones y posterior sincronización agregan complejidad al sistema, y pueden ser una potencial fuente de fallos que no se hayan podido detectar en la fase de desarrollo, aunque se haya puesto un particular esfuerzo en que todos estos procesos complejos sean invisibles para el usuario.

Sin embargo, el sistema ha sido desarrollado de forma que en caso de errores se informe al usuario, y que la terminología usada sea independiente del contexto del usuario, de forma que pueda ser usado por un experto en TEA o por un abogado, por ejemplo, y siempre cumpliendo con unos mínimos de seguridad, por ejemplo usando el almacenamiento seguro de Eclipse o utilizando los sistemas de autenticación seguros de Django.

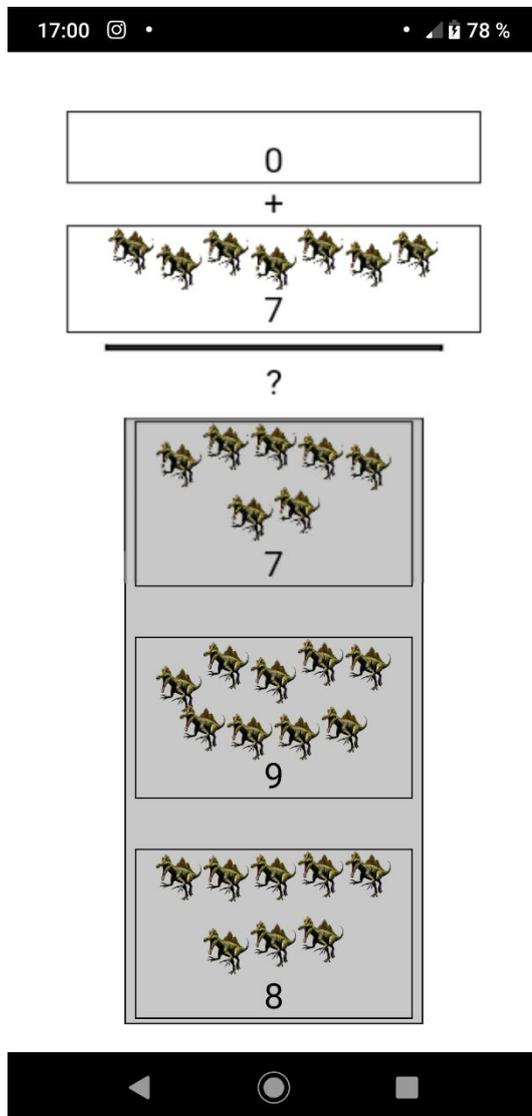
Aunque no se ha podido comprobar con usuarios, el sistema ha sido desarrollado sin suponer conocimiento previo del usuario sobre programación, así que aunque al principio no se aprovechara al 100 % toda la capacidad del sistema, la funcionalidad del mismo es limitada, por lo que no debería ser muy complicado que los usuarios aprendieran a utilizarlo y se acostumbraran rápidamente.

5.3. Evaluación básica por parte de expertos

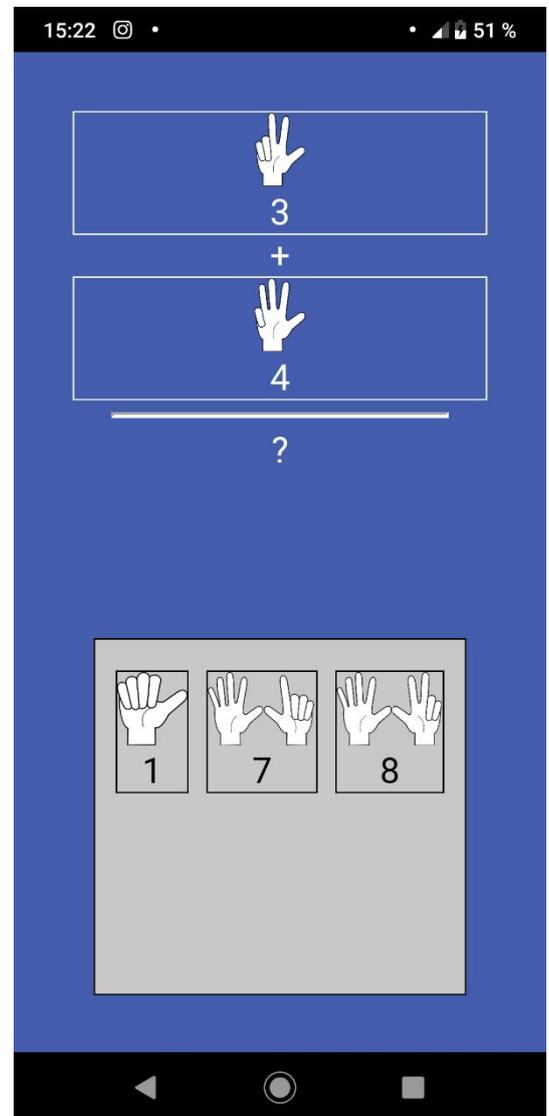
Una vez finalizado el desarrollo, se enviaron una serie de capturas mostrando la interfaz de la aplicación móvil a la directora del colegio de educación especial. Proporcionó un *feedback* que consistió en los siguientes puntos:

- Confirmar que las opciones de adaptabilidad eran adecuadas.
- Confirmar que la mecánica del juego era válida.
- Sugerir el encuadre de números y objetos en los sumandos de la operación, de modo que quede más clara la asociación de números con objetos.
- Sugerir la sustitución del símbolo igual por una línea horizontal, para que sea similar a una suma en vertical.
- Sugerir la adición de un signo de interrogación debajo de la línea horizontal, de modo que quede más claro que hay que pulsar sobre una acción.
- Sugerir la diferenciación de los posibles resultados con un color de fondo neutro.

Como parte de la fase de evaluación, se han implementado esos cambios (figura 5.16) que han sido verificados de nuevo por la directora del colegio de educación especial.



(a) Fondo blanco y dinosaurios



(b) Fondo azul y dedos

Figura 5.16: En estas figuras se muestran diferentes combinaciones de colores y objetos, en el sistema mejorado gracias al feedback de expertos

CONCLUSIONES Y TRABAJO FUTURO

6.1. Conclusiones

En este trabajo se ha implementado un sistema de aprendizaje adaptable para personas con TEA. Este sistema está compuesto por una herramienta de autor, un servidor web y una aplicación móvil.

El objetivo del trabajo ha sido implementar un sistema de aprendizaje adaptable para colegios de educación especial con estudiantes con TEA, con el fin de ayudar a que los alumnos con TEA puedan aprender en este caso a realizar sumas sencillas, permitiendo a los docentes adaptar la aplicación fácilmente y de forma consistente a través de la generación de cuestionarios, para lo que se les proporciona una herramienta de autor.

Esta herramienta de autor se ha programado con la intención de que sea lo más sencilla posible, y para conseguirlo se ha realizado una investigación sobre el estado del arte, descubriendo que un ámbito en el que se usan con éxito herramientas que permiten personalizar sistemas es el ámbito del *Internet de las Cosas*, un ámbito en auge.

Se ha descubierto que una buena manera de permitir a los usuarios realizar desarrollos es por medio de diagramas (hecho que se ve reforzado al poder ser modelados los cuestionarios como grafos), y esta idea se ha visto implementada en este trabajo, siendo la herramienta de autor un generador de diagramas. Implementar un generador de diagramas es una tarea compleja, por lo que para aligerar el desarrollo y permitir una alta capacidad de extensión en el sistema, se ha desarrollado siguiendo el paradigma de *desarrollo basado en modelos*, que mediante el uso de herramientas como Eclipse, Epsilon y EuGENia han permitido generar una herramienta altamente modular, y que, si bien no ha sido posible realizar pruebas de usuario con ella, permite generar cuestionarios a personas ajenas a la programación, lo que es fundamental para el funcionamiento del sistema.

La generación de estos diagramas o conjuntos de preguntas, requiere de un almacenamiento online de los mismos, de forma que se puedan transmitir rápidamente a todos los docentes y que puedan utilizarse de forma sencilla para adaptar el sistema de aprendizaje mediante sumas sencillas. Para hacer esto se vuelve imprescindible tener una arquitectura cliente-servidor, en la que la herramienta y

el sistema toman el papel de cliente. El software servidor se ha implementado con Django, un popular *framework* web que permite aligerar el desarrollo y que permite también que el software sea integrado fácilmente en futuros desarrollos.

Este software servidor tiene dos partes fundamentales, una interfaz que permite gestionar de forma sencilla los usuarios y visualizar y probar los conjuntos de preguntas, y una API REST que permite operar sobre los datos almacenados en el servidor tanto a la herramienta de autor como a la aplicación móvil.

Por último, la aplicación móvil es un sistema de aprendizaje mediante sumas sencillas adaptable que tiene como objetivo ayudar con uno de los grandes problemas de la educación a personas con TEA, que es que dados los *déficits* que presentan muchas de estas personas en sus capacidades ejecutivas, es decir, de seguir instrucciones, en muchas ocasiones es complicado que aprendan conceptos que para una persona sin TEA son más fáciles, como por ejemplo las sumas sencillas. Para abordar este problema es necesario modelar los intereses e inquietudes de los usuarios, de forma que puedan estar concentrados y seguir las sesiones de aprendizaje más fácilmente, aprovechando además el conocimiento experto de los docentes y el poder que se les da al proporcionarles la capacidad de decidir ellos mismos cómo y bajo qué condiciones se adapta la sesión a través de la generación de cuestionarios.

6.2. Trabajo futuro

Este trabajo, aunque ha sido pensado como un producto que pueda ser usable por colegios de educación especial, más que un producto completo ha llegado a ser una base modular y extensible sobre la que realizar desarrollos posteriores, que pueden ir orientados en dos caminos distintos.

6.2.1. Mejoras en el trabajo

Existen numerosas mejoras que realizar, dado que el alcance de este trabajo ha sido limitado. La principal mejora es la validación y mejoras que puedan surgir de la realización de pruebas con usuarios. En este trabajo, dadas las circunstancias sobrevenidas, no se han podido realizar pruebas con usuarios, y es algo muy importante para poder determinar el nivel de usabilidad real del sistema. Existen cuestionarios que se pueden realizar a los participantes en las pruebas, como por ejemplo el *System Usability Scale (SUS)*, que permite, por medio de diez preguntas determinar la usabilidad del sistema [44].

Sin embargo, este cuestionario sólo serviría para el caso de la herramienta de autor, de la interfaz del servidor y de la interfaz restringida de la aplicación móvil.

En cualquier caso, de realizar pruebas con personas con TEA se debe tener en cuenta que es

necesario utilizar un método de evaluación basado en la evidencia científica [45].

Aparte de pruebas de usuario, el sistema actual ofrece posibilidades de mejora en otros ámbitos, como por ejemplo en las adaptaciones de la aplicación. Dado el sistema implementado, es muy sencillo agregar nuevas adaptaciones, que no requieren de una actualización de la herramienta de autor, simplemente es necesario informar a los autores del nombre del nuevo objetivo de forma que puedan agregarlo al conjunto de preguntas deseado.

En caso de querer mejorar la herramienta de autor, es posible establecer un sistema de actualizaciones automáticas gracias al soporte que ofrece Eclipse a través de la tecnología Equinox p2 [46]. Una mejora evidente que se podría agregar es la de permitir agregar lógica *booleana* a los cuestionarios, de forma que haya más flexibilidad a la hora de activar y desactivar objetivos, aunque actualmente se puede si tratamos cada diagrama como una *Máquina de Moore* en la que las preguntas sean estados y las respuestas transiciones entre estados. Sin embargo, este enfoque puede complicar bastante la creación de diagramas, y podría evitarse implementando elementos específicos para soportar lógica *booleana*.

También es posible mejorar la implementación del sistema de aprendizaje en sí, transformándolo de un sistema adaptable a un sistema adaptativo (sería relativamente sencillo, dado que la implementación actual ya soporta adaptatividad), para lo que habría que definir las reglas de adaptatividad con la ayuda de expertos, y siempre teniendo en cuenta que los cambios que se hagan durante la ejecución no deberían ser muy radicales para evitar provocar estrés a la persona con TEA que esté interactuando en ese momento [47].

6.2.2. Nuevos desarrollos que hagan uso de este trabajo

Aparte de mejorar el trabajo, pueden surgir nuevos trabajos a partir de este. El sistema de preguntas, tal y como se comentó en el capítulo de diseño, está diseñado de forma que los conceptos que se usan sean aplicables a cualquier ámbito. Por ejemplo, podría utilizarse como un sistema de modelado de usuario para un despacho de abogados, o en el ámbito sanitario, el sistema permitiría que el especialista no tuviera que entrevistar previamente al usuario para conocer su estado. Además, como el sistema soporta múltiples cuestionarios, es posible también adaptar qué cuestionario se muestra al usuario en función de información previa, que puede venir dada de otro cuestionario o de una fuente externa.

La integración con sistemas ya existentes es viable dado el alto desacoplamiento que existe entre software servidor y herramienta de autor, y además si el servidor en el que se quiere integrar ya usa Django, la integración es trivial al estar el software servidor encapsulado como una *aplicación Django*.

Con respecto a la aplicación móvil, toda la lógica de las preguntas está separada de la lógica del juego, siendo posible coger los módulos de una u otra y reutilizarlos en otro proyecto.

BIBLIOGRAFÍA

- [1] B. García-Riaza and A. Iglesias-Rodríguez, "Mobile devices as learning tools: present and prospective use by future professionals of education," in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM '16, (Salamanca, Spain), pp. 637–641, Association for Computing Machinery, Nov. 2016.
- [2] G. Alley-Young, "Technology Tools for Students with Autism: Innovations that Enhance Independence and Learning," *Canadian Journal of Communication*, vol. 41, July 2016.
- [3] J. Llanos and R. M. Carro, "The squares: A multi-touch adaptive game for children integration," in *2015 International Symposium on Computers in Education (SIIE)*, pp. 137–140, Nov. 2015.
- [4] M. Elahi, F. Ricci, and N. Rubens, "Active Learning in Collaborative Filtering Recommender Systems," in *E-Commerce and Web Technologies* (M. Hepp and Y. Hoffner, eds.), Lecture Notes in Business Information Processing, (Cham), pp. 113–124, Springer International Publishing, 2014.
- [5] B. Kitchenham, "Procedures for performing systematic reviews," vol. 33, p. 33, 2004.
- [6] A. Liberati, D. G. Altman, J. Tetzlaff, C. Mulrow, P. C. Gøtzsche, J. P. A. Ioannidis, M. Clarke, P. J. Devereaux, J. Kleijnen, and D. Moher, "The PRISMA statement for reporting systematic reviews and meta-analyses of studies that evaluate health care interventions: Explanation and elaboration," vol. 6, no. 7, 2009.
- [7] J. Martínez, "CORRECTED - UPDATE 2-more mobile phone makers back google's android," 2008.
- [8] D. Cihak, E. Moore, R. Wright, D. McMahon, M. Gibbons, and C. Smith, "Evaluating augmented reality to complete a chain task for elementary students with autism," vol. 31, no. 2, pp. 99–108, 2016.
- [9] A. Díaz-Escudero, J. C. Torrado, J. Gomez, and G. Montoro, "Technology: aided authoring tools for ASD coaching support," in *Proceedings of the XIX International Conference on Human Computer Interaction - Interacción 2018*, pp. 1–6, ACM Press, 2018.
- [10] C. Fage, L. Pommereau, C. Consel, Balland, and H. Sauzéron, "Tablet-based activity schedule for children with autism in mainstream environment," in *Proceedings of the 16th International ACM SIGACCESS Conference on Computers & Accessibility*, ASSETS '14, pp. 145–152, ACM, 2014. event-place: Rochester, New York, USA.
- [11] M. Hentschel, C. Lange-Kuttner, and B. B. Averbek, "Sequence learning with stochastic feedback in a cross-cultural sample of boys in the autistic spectrum," vol. 51, no. 2, pp. 179–194, 2016.
- [12] M. Gotsis, J. Piggot, D. Hughes, and W. Stone, "SMART-games: a video game intervention for children with autism spectrum disorders," in *Proceedings of the 9th International Conference on Interaction Design and Children - IDC '10*, p. 194, ACM Press, 2010.
- [13] B. Li, A. Atyabi, M. Kim, E. Barney, A. Y. Ahn, Y. Luo, M. Aubertine, S. Corrigan, T. St. John, Q. Wang, M. Mademtzi, M. Best, and F. Shic, "Social influences on executive functioning in autism:

- Design of a mobile gaming platform,” in *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pp. 443:1–443:13, ACM, 2018. event-place: Montreal QC, Canada.
- [14] C. Ardito, M. F. Costabile, G. Desolda, M. Manca, M. Matera, F. Paternò, and C. Santoro, “Improving tools that allow end users to configure smart environments,” in *End-User Development* (A. Malizia, S. Valtolina, A. Morch, A. Serrano, and A. Stratton, eds.), vol. 11553, pp. 244–248, Springer International Publishing, 2019.
- [15] A. Balderas, I. Ruiz-Rube, J. M. Mota, J. M. Doderó, and M. Palomo-Duarte, “A development environment to customize assessment through students interaction with multimodal applications,” in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM '16, pp. 1043–1048, ACM, 2016. event-place: Salamanca, Spain.
- [16] G. Desolda, C. Ardito, and M. Matera, “Empowering end users to customize their smart environments: Model, composition paradigms, and domain-specific tools,” vol. 24, no. 2, pp. 12:1–12:52, 2017.
- [17] V. Paramasivam, J. Huang, S. Elliott, and M. Cakmak, “Computer science outreach with end-user robot-programming tools,” in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education - SIGCSE '17*, pp. 447–452, ACM Press, 2017.
- [18] A. Leonardou and M. Rigou, “An adaptive mobile casual game for practicing multiplication,” in *Proceedings of the 20th Pan-Hellenic Conference on Informatics*, PCI '16, pp. 29:1–29:6, ACM, 2016. event-place: Patras, Greece.
- [19] N. S. Raj and V. G. Renumol, “Architecture of an adaptive personalized learning environment (APLE) for content recommendation,” in *Proceedings of the 2Nd International Conference on Digital Technology in Education*, ICDTE 2018, pp. 17–22, ACM, 2018. event-place: Bangkok, Thailand.
- [20] C. Duarte, L. Carriço, T. Guerreiro, C. Almeida, S. Nobre, and A. M. Campos, “Supporting autism therapists: co-designing interventions,” in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pp. 1213–1218, Association for Computing Machinery, 2014.
- [21] M. Gelsomini, “An affordable virtual reality learning framework for children with neurodevelopmental disorder,” in *Proceedings of the 18th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '16, pp. 343–344, Association for Computing Machinery, 2016.
- [22] M. Gelsomini, M. Spitale, E. Beccaluva, L. Viola, and F. Garzotto, “Reflex: adaptive learning beyond the screen,” in *Proceedings of the 24th International Conference on Intelligent User Interfaces: Companion*, IUI '19, pp. 29–30, Association for Computing Machinery, 2019.
- [23] M. F. Msiska and L. van Zijl, “From visual scripting to lua,” in *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, SAICSIT '12, pp. 94–99, Association for Computing Machinery, 2012.
- [24] H. Duncan and J. Tan, “A visual task manager application for individuals with autism,” vol. 27, no. 6, pp. 49–57, 2012.

- [25] G. R. Hayes and S. W. Hosaflook, “HygieneHelper: promoting awareness and teaching life skills to youth with autism spectrum disorder,” in *Proceedings of the 12th International Conference on Interaction Design and Children*, IDC '13, pp. 539–542, Association for Computing Machinery, 2013.
- [26] C. Fage, “an emotion regulation app for school inclusion of children with ASD: design principles and preliminary results for its evaluation,” no. 112, pp. 8–15, 2015.
- [27] N. Harrold, C. T. Tan, D. Rosser, and T. W. Leong, “CopyMe: an emotional development game for children,” in *CHI '14 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '14, pp. 503–506, Association for Computing Machinery, 2014.
- [28] H. Hong, S. Yarosh, J. G. Kim, G. D. Abowd, and R. I. Arriaga, “Investigating the use of circles in social networks to support independence of individuals with autism,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pp. 3207–3216, Association for Computing Machinery, 2013.
- [29] A. Ishaq, M. Shoaib, H. Raza, M. Zaman, M. Shahid, and A. Khaliq, “iTrack application for autism: Supporting and improving safety skills of people with autism spectrum disorder,” in *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers*, UbiComp '18, pp. 70–73, Association for Computing Machinery, 2018.
- [30] I. Lozano-Osorio, D. Roldán-Álvarez, A. Bacelo, and E. Martín, “BlueThinking, a programming tool for the development of executive functions at childhood,” in *Proceedings of the XX International Conference on Human Computer Interaction*, Interacci3n '19, pp. 1–4, Association for Computing Machinery, 2019.
- [31] A. M. Piper, N. Weibel, and J. Hollan, “TAP & PLAY: an end-user toolkit for authoring interactive pen and paper language activities,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12, pp. 149–158, Association for Computing Machinery, 2012.
- [32] M. Morgenlander, A. D'Eugenio, and D. Witan, “uChoose by InteractAble: learning social skills via game play,” in *Proceedings of the 14th International Conference on Interaction Design and Children*, IDC '15, pp. 403–405, Association for Computing Machinery, 2015.
- [33] R. Munoz-Soto, C. Becerra, R. Noël, T. Barcelos, R. Villarroel, S. Kreisel, and M. Cambior, “Project@ matemáticas: A learning object for supporting the practitioners in autism spectrum disorders,” in *2016 XI Latin American Conference on Learning Objects and Technology (LACLO)*, pp. 1–6, 2016. ISSN: null.
- [34] C. Wilson, M. Brereton, B. Ploderer, L. Sitbon, and B. Saggars, “Digital strategies for supporting strengths- and interests-based learning with children with autism,” in *Proceedings of the 19th International ACM SIGACCESS Conference on Computers and Accessibility*, ASSETS '17, pp. 52–61, Association for Computing Machinery, 2017.
- [35] U. Qidwai and O. Connor, “A general purpose game module for children with autism spectrum disorder,” in *2018 IEEE-EMBS Conference on Biomedical Engineering and Sciences (IECBES)*,

- pp. 426–431, 2018. ISSN: null.
- [36] A. Tartaro, J. Cassell, C. Ratz, J. Lira, and V. Nanclares-Nogués, “Accessing peer social interaction: Using authorable virtual peer technology as a component of a group social skills intervention program,” vol. 6, no. 1, pp. 2:1–2:29, 2014.
- [37] R. Gronback, “Eclipse Modeling Project | The Eclipse Foundation.” Library Catalog: www.eclipse.org.
- [38] J. D. Poole, “Model-Driven Architecture: Vision, Standards And Emerging Technologies,” *ECOOP 2001*, p. 15, Apr. 2001.
- [39] “Epsilon.”
- [40] “EuGENia.”
- [41] “The Web framework for perfectionists with deadlines | Django.”
- [42] “Apache Cordova.”
- [43] D. J. Mayhew, *The Usability Engineering Lifecycle: A Practitioner’s Handbook for User Interface Design*. San Francisco, Calif: Morgan Kaufmann, Mar. 1999.
- [44] A. S. f. P. Affairs, “System Usability Scale (SUS),” Sept. 2013. Library Catalog: www.usability.gov
Publisher: Department of Health and Human Services.
- [45] B. Reichow, F. Volkmar, and D. Cicchetti, “Development of the Evaluative Method for Evaluating and Determining Evidence-Based Practices in Autism,” *Journal of autism and developmental disorders*, vol. 38, pp. 1311–9, Aug. 2008.
- [46] E. committers, “Equinox p2 | The Eclipse Foundation.” Library Catalog: www.eclipse.org.
- [47] A. Mejía-Figueroa, M. d. I. Q. Cisnero, and J. R. Juárez-Ramírez, “Developing Usable Software Applications for Users with Autism: User Analysis, User Interface Design Patterns and Interface Components,” in *2016 4th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pp. 195–203, Apr. 2016.

APÉNDICES

UAM

UNIVERSIDAD AUTONOMA
DE MADRID