

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**ROBDA: PLATAFORMA PARA
APRENDIZAJE COMUNITARIO
DE BASES DE DATOS CON
MONITORIZACIÓN**

AUTOR/A: RUBÉN PÉREZ MERCADO

Puerto Real, junio de 2023

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA INFORMÁTICA

**ROBDA: PLATAFORMA PARA
APRENDIZAJE COMUNITARIO
DE BASES DE DATOS CON
MONITORIZACIÓN**

DIRECTOR/A: MANUEL PALOMO DUARTE
CODIRECTOR/A: ANTONIO BALDERAS ALBERICO
AUTOR/A: RUBÉN PÉREZ MERCADO

Puerto Real, junio de 2023

Declaración personal de autoría

Rubén Pérez Mercado con DNI 52418353L, estudiante del Grado en Ingeniería Informática en la Escuela Superior de Ingeniería de la Universidad de Cádiz, como autor de este documento académico titulado ROBDA: Plataforma para aprendizaje comunitario de bases de datos con monitorización y presentado como Trabajo Final de Grado

DECLARO QUE

Es un trabajo original, que no copio ni utilizo parte de obra alguna sin mencionar de forma clara y precisa su origen tanto en el cuerpo del texto como en su bibliografía y que no empleo datos de terceros sin la debida autorización, de acuerdo con la legislación vigente. Asimismo, declaro que soy plenamente consciente de que no respetar esta obligación podrá implicar la aplicación de sanciones académicas, sin perjuicio de otras actuaciones que pudieran iniciarse. En Puerto Real, a 27 de junio de 2023

Fdo: Rubén Pérez Mercado

Agradecimientos

A mis padres, hermana y abuelos, por ayudarme cada día.

A mis tutores, por confiar en mí.

Licencia

Copyright © 2023 Rubén Pérez Mercado. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

Índice general

Índice de figuras	XI
Índice de tablas	XIII
I Prolegómeno	1
1. Introducción	2
1.1. Motivación	2
1.2. Alcance y Objetivos	3
1.3. Glosario de términos	3
1.4. Organización del documento	4
2. Estado del arte	7
2.1. Guess it!	7
2.2. VirUCA	7
2.3. SMCP-Training	8
2.4. StudentQuiz	8
2.5. Anki	8
2.6. Comparativas	9
3. Plan de gestión de proyecto	11
3.1. Metodología de desarrollo	11
3.2. Planificación del proyecto	11
3.2.1. Iteración 0: Fase inicial	11
3.2.2. Iteración 1: Análisis inicial	11
3.2.3. Iteración 2: Aprendizaje de tecnologías	11
3.2.4. Iteración 3: Creación de la API	12
3.2.5. Iteración 4: Desarrollo de la web	12
3.2.6. Iteración 5: Correcciones en la API	12
3.2.7. Iteración 6: Despliegue del proyecto	12
3.2.8. Iteración 7: Creación de la PWA	12
3.2.9. Redacción de la memoria	12
3.3. Diagrama de Gantt	13
3.4. Costes	13
3.4.1. Costes humanos	13
3.4.2. Costes Materiales	13
II Desarrollo	18
4. Requisitos del Sistema	19
4.1. Objetivos de Negocio	19

4.2.	Objetivos del Sistema	19
4.2.1.	Visión general del proyecto	19
4.3.	Requisitos funcionales	21
4.4.	Requisitos no funcionales	22
5.	Análisis del Sistema	25
5.1.	Modelo Conceptual de Datos	25
5.2.	Modelo de Casos de Uso	26
5.2.1.	Diagrama de casos de uso	26
5.2.2.	Descripción de caso de uso: Inicio de sesión	26
5.2.3.	Descripción de caso de uso: Registro	27
5.2.4.	Descripción de caso de uso: Gestión Temas	28
5.2.4.1.	Caso de uso: Creación de temas	28
5.2.4.2.	Caso de uso: Modificación de tema	28
5.2.5.	Descripción de caso de uso: Gestión Imágenes	29
5.2.5.1.	Caso de uso: Creación de imágenes	29
5.2.5.2.	Caso de uso: Modificación de imagen	30
5.2.6.	Descripción de caso de uso: Gestión Preguntas	31
5.2.6.1.	Caso de uso: Creación de preguntas	31
5.2.6.2.	Caso de uso: Modificación de pregunta	32
5.2.6.3.	Caso de uso: Decidir sobre reporte	32
5.2.7.	Descripción de caso de uso: Gestión Competiciones	33
5.2.7.1.	Caso de uso: Creación de competiciones	33
5.2.7.2.	Caso de uso: Modificación de competición	34
5.2.7.3.	Caso de uso: Terminar competición	35
5.2.8.	Descripción de caso de uso: Ver estadísticas	35
5.2.9.	Descripción de caso de uso: Reportar pregunta	35
5.2.10.	Descripción de caso de uso: Repaso	36
5.2.11.	Descripción de caso de uso: Participación en competición	37
5.2.12.	Descripción de caso de uso: Duelo	38
5.2.12.1.	Caso de uso: Crear duelo	38
5.2.12.2.	Caso de uso: Decidir duelo	39
5.2.12.3.	Caso de uso: Jugar duelo	39
5.2.12.4.	Caso de uso: Ver informe del duelo	41
6.	Diseño del Sistema	43
6.1.	Arquitectura del Sistema	43
6.2.	Diseño Físico de Datos	44
6.2.1.	Tabla User	44
6.2.2.	Tabla Tema	45
6.2.3.	Tabla Imagen	46
6.2.4.	Tabla Pregunta	46
6.2.5.	Tabla Opción	46
6.2.6.	Tabla Evento	47
6.2.7.	Tabla Partida	47
6.2.8.	Tabla AnswerLogs	47
6.2.9.	Tabla Repaso	48
6.2.10.	Tabla UserComp	48

6.2.11. Tabla Duelo	49
6.2.12. Tabla Report	49
6.2.13. UsuarioPregunta	49
6.2.14. Aclaraciones sobre los campos de las tablas	50
6.3. Diseño de la Interfaz de Usuario	51
7. Construcción del Sistema	55
7.1. Entorno de Construcción	55
7.1.1. Python	55
7.1.2. JavaScript	55
7.1.3. MySQL	56
7.1.4. Postman	56
7.1.5. Git y GitHub	57
7.1.6. L ^A T _E X	57
7.1.7. Draw.io Diagrams	57
7.1.8. Apache HTTP Server	58
7.2. Código Fuente	58
7.2.1. API	58
7.2.2. Web	60
7.3. Características destacadas del software	61
7.3.1. Elección personalizada de las preguntas	61
7.3.2. Autenticación en la API desde el cliente	62
7.3.3. Ampliación de imágenes	63
7.3.4. PWA	63
8. Pruebas del Sistema	67
8.1. Entorno de pruebas	67
8.2. Errores encontrados en las pruebas	67
8.3. Pruebas de aceptación	68
8.4. Resultados	69
III Epílogo	73
9. Conclusiones	74
9.1. Objetivos alcanzados	74
9.2. Lecciones aprendidas	75
9.3. Trabajo futuro	75
9.3.1. Mejora de la personalización de las preguntas en el modo Repaso	76
9.3.2. Preguntas con código HTML incrustado	76
9.3.3. Estadísticas	77
GNU Free Documentation License	79
1. APPLICABILITY AND DEFINITIONS	79
2. VERBATIM COPYING	81
3. COPYING IN QUANTITY	81
4. MODIFICATIONS	82
5. COMBINING DOCUMENTS	83

6. COLLECTIONS OF DOCUMENTS	84
7. AGGREGATION WITH INDEPENDENT WORKS	84
8. TRANSLATION	84
9. TERMINATION	85
10. FUTURE REVISIONS OF THIS LICENSE	85
11. RELICENSING	85
ADDENDUM: How to use this License for your documents	86
Bibliografía	89
IV Anexos	93
A. Manual del desarrollador	94
A.1. Introducción	94
A.2. Preparación del entorno de trabajo	94
A.2.1. Configuración de la API	95
A.2.2. Configuración de la web	96
B. Manual de usuario	99
B.1. Introducción	99
B.2. Instalación	99
B.3. Uso del sistema	101
B.3.1. Acceso al sistema	101
B.3.2. Perfil	101
B.3.3. Acerca de	101
B.3.4. Alumno	103
B.3.4.1. Modo Repaso	104
B.3.4.2. Modo Competición	105
B.3.4.3. Modo Duelo	108
B.3.5. Profesor	112
B.3.5.1. Administración	112
B.3.5.2. Competiciones	116
B.3.5.3. Estadísticas	118

Índice de figuras

3.1. Diagrama de Gantt I	15
3.2. Diagrama de Gantt II	16
3.3. Diagrama de Gantt III	17
5.1. Diagrama Entidad-Relación Extendido	26
5.2. Diagrama de casos de uso de alto nivel	27
6.1. Arquitectura de la plataforma.	43
6.2. Esquema de la base de datos de <i>ROBDA</i>	45
6.3. Boceto de la plataforma. Versión del profesor para web.	51
6.4. Boceto de la plataforma. Versión del alumno para el móvil.	52
6.5. Primera versión de la plataforma. Versión del profesor para web.	53
6.6. Primera versión de la plataforma. Acceso de alumno desde el navegador del móvil.	54
7.1. Logos de Python, Django y Django Rest Framework	55
7.2. Logos de JavaScript, React y NextJS	56
7.3. Logo de MySQL	56
7.4. Logo de Postman	56
7.5. Logo de GitHub	57
7.6. Logo de \LaTeX	57
7.7. Logo de Draw.io	57
7.8. Logo de Apache HTTP Server	58
7.9. Ampliación de imágenes desde el teléfono móvil.	64
B.1. Instalación de la PWA en Windows 11. Icono de instalación	99
B.2. Instalación de la PWA en Windows 11. Diálogo de confirmación	100
B.3. Instalación de la PWA en Windows 11. Icono y aplicación en el escritorio	100
B.4. Instalación de la PWA en Android	101
B.5. Pantalla de inicio de sesión	102
B.6. Pantalla de registro	102
B.7. Pantalla de edición del perfil	103
B.8. Pantalla de información del proyecto	103
B.9. Menú principal del alumno	104
B.10. Formulario de inicio de repaso	105
B.11. Cuestionario de repaso	106
B.12. Acierto en una pregunta de repaso	107
B.13. Fallo en una pregunta de repaso	108
B.14. Formulario de reporte	109
B.15. Detalle de la competición	109
B.16. Formulario de creación de preguntas de competiciones	110
B.17. Detalle de la competición tras crear la pregunta	110
B.18. Fase de cuestionario de la competición	111

B.19.Inicio de cuestionario en competición	111
B.20.Informe de la partida de la competición	112
B.21.Valoración de la competición	112
B.22.Detalle de la competición finalizada	113
B.23.Listado duelos	113
B.24.Fase inicial duelo	114
B.25.Cuestionario de duelos	114
B.26.Duelo pendiente	115
B.27.Listado de duelos con duelo pendiente de decisión	115
B.28.Duelo finalizado	116
B.29.Menú principal de los profesores	116
B.30.Listado de preguntas	117
B.31.Detalle de la pregunta	117
B.32.Apartado de estadísticas y listado de reportes de una pregunta	118
B.33.Listado de imágenes	118
B.34.Listado de temas	119
B.35.Detalle de competición (profesores)	119
B.36.Competición terminada (profesor)	120
B.37.Estadísticas I	120
B.38.Estadísticas II	121

Índice de tablas

2.1. Comparativa de características de las plataformas estudiadas I	9
2.2. Comparativa de características de las plataformas estudiadas II	10
8.1. Comparativa de las estadísticas de ROBDA frente a webs similares	69

Parte I
Prolegómeno

1. Introducción

1.1. Motivación

En la actualidad, el acceso a Internet y el uso de dispositivos móviles se han convertido en herramientas indispensables para la comunicación y la interacción entre las personas. Esto ha generado cambios significativos en diversos ámbitos de la sociedad, incluyendo la educación. En los últimos años, se ha intentado hacer el aprendizaje más atractivo mediante la gamificación, es decir, mediante técnicas y elementos de los videojuegos para motivar y comprometer a los usuarios. Otra práctica que ha tomado fuerza en la educación en línea es el *peer-assessment*. Esta técnica consiste en que los estudiantes evalúan el trabajo de sus compañeros, lo que les permite desarrollar habilidades críticas y de análisis, además de fomentar la colaboración y el aprendizaje entre iguales. Los estudiantes no solo aprenden realizando las actividades propias, sino que también lo hacen viendo el trabajo de otros.

En la asignatura de Bases de Datos de la Universidad de Cádiz, han de verse ciertos contenidos teóricos que en ocasiones son difíciles de asimilar para los estudiantes. La misión de este proyecto no es otra que motivar a los estudiantes a aprender estos conceptos de forma amena y entretenida, usando las técnicas descritas anteriormente. Partimos del trabajo realizado años atrás con *Guess it!* [4], aunque esta vez nos lo llevamos al terreno de las bases de datos. Queremos que el alumno se divierta mientras aprende, pero también queremos que cuando vaya a prepararse para los exámenes de la asignatura pueda contar con una herramienta útil, que pueda utilizar a su ritmo.

En este contexto, surge *ROBDA*, una aplicación en la que los alumnos deberán responder a preguntas cortas que habrán creado sus compañeros sobre conceptos teóricos de bases de datos como el modelo entidad-relación, SQL o normalización. Se fomentará la participación activa de los usuarios para que ellos mismos identifiquen preguntas erróneas, y así mantener un banco de preguntas extenso y fiable. Los profesores podrán monitorizar toda la plataforma, que contará con varios modos de juego para que los alumnos puedan tener libertad de elegir cómo aprender.

Este proyecto tiene sus raíces en otros proyectos similares que Manuel Palomo y Antonio Balderas habían ideado. Tras trabajar en el curso 2021/22 con ellos en el desarrollo de un proyecto web relacionado con el aprendizaje de SQL para la misma asignatura, que resultó en un trabajo publicado en [23], me ofrecieron crear esta plataforma para el curso 2022/23. El hecho de crear una plataforma que pudiese servir tanto a profesores como alumnos durante los próximos cursos de una asignatura que ya conocía, además de la motivación a seguir aprendiendo programación web fue lo que me llevó a aceptar la propuesta.

1.2. Alcance y Objetivos

ROBDA pretenderá satisfacer las necesidades de estudio de una materia cualquiera. Pese a que la aplicación se centra en la asignatura de Bases de Datos, debería ser perfectamente válida para otras asignaturas en las que el alumno tenga que aprender conceptos teóricos básicos, sean relacionados o no con la informática. Lo único necesario es que se puedan idear preguntas de tipo test con opción simple sobre esos conceptos. La plataforma tendrá que motivar a los estudiantes a participar en la asignatura, y permitir que los profesores puedan ver sus partidas y toda la batería de preguntas que los alumnos creen.

El objetivo principal del proyecto será desarrollar una plataforma de aprendizaje basada en preguntas y respuestas con las siguientes características:

- Deberá tener dos roles distintos: El alumno, y el profesor.
- Contará con una base de datos de preguntas dinámicas, que el profesor pueda gestionar fácilmente, y en la que los alumnos también puedan contribuir (siempre bajo la supervisión del profesor).
- Tendrá diferentes modos de juego gestionables por el profesor.
- Ofrecerá a los profesores diversas estadísticas de uso e informes de las partidas de alumnos de todos los modos de juego.

1.3. Glosario de términos

- **API (Application Programming Interface):** Conjunto de reglas y protocolos que permiten a diferentes aplicaciones comunicarse entre sí. Proporciona una interfaz que define los métodos de interacción y los formatos de datos utilizados para intercambiar información [9].
- **REST (Representational State Transfer):** Estilo de arquitectura *software* para diseñar servicios web que se basa en el protocolo HTTP. Permite la creación, lectura, actualización y eliminación de recursos a través de las operaciones estándar de HTTP (e.g. GET, POST, PUT y DELETE). REST enfatiza la escalabilidad, la interoperabilidad y la simplicidad en el desarrollo de APIs [17].
- **JSON (JavaScript Object Notation):** Formato de intercambio de datos ligero y legible por humanos. Se utiliza ampliamente en aplicaciones web y APIs para representar datos estructurados. Se basa en una sintaxis de pares clave-valor y es fácilmente procesable por una amplia variedad de lenguajes de programación [12].
- **JWT (JSON Web Token):** Estándar abierto basado en JSON para la creación de tokens de acceso seguros y autenticados. Se utiliza para autenticar y autorizar solicitudes en aplicaciones web y APIs. Consta de tres partes: encabezado, carga útil y firma, y se puede verificar y decodificar para obtener información sobre el usuario y los permisos asociados [13].

- **React:** Biblioteca de JavaScript de código abierto creada para construir interfaces de usuario interactivas. Permite crear componentes reutilizables que se actualizan eficientemente cuando cambia el estado de la aplicación [16].
- **Next.js:** Framework de React que se enfoca en la creación de aplicaciones web rápidas y escalables. Proporciona características como renderizado del lado del servidor (SSR), generación de sitios estáticos, enrutamiento simple y soporte para API integradas [14].
- **Python:** Lenguaje de programación de alto nivel, interpretado y de propósito general. Es conocido por su legibilidad y su sintaxis clara y concisa. Es ampliamente utilizado en diversos campos como desarrollo web, análisis de datos, inteligencia artificial, automatización de tareas y scripting. Es conocido por su amplia biblioteca estándar y por su gran comunidad [15].
- **Django:** Framework de desarrollo web de alto nivel y de código abierto, escrito en Python. Proporciona una estructura y conjunto de herramientas que facilitan la creación de aplicaciones web robustas y escalables [11].
- **DRF (Django Rest Framework):** Biblioteca creada por terceros para Django que facilita la construcción de APIs web potentes y flexibles. Proporciona herramientas para la serialización de datos, la autenticación, la autorización, la validación de datos y la gestión de solicitudes HTTP. DRF se basa en los principios de REST y facilita la creación de APIs RESTful [7].
- **PWA (Progressive Web Application):** Aplicación web que simula a una aplicación nativa. Las PWAs son diseñadas para ser rápidas, confiables y funcionar sin conexión. Pueden utilizar características como el almacenamiento en caché, notificaciones push y acceso a hardware del dispositivo para proporcionar una experiencia fluida y optimizada en diferentes dispositivos y plataformas. Las PWAs se ejecutan en el navegador web y se pueden instalar en el escritorio o pantalla de inicio del dispositivo del usuario [10].

1.4. Organización del documento

El presente documento está estructurado en una serie de capítulos, que describirán cada una de las etapas del proyecto. A continuación, detallamos cada uno de los capítulos:

- **Capítulo 1: Introducción.** En el capítulo actual, se ha querido realizar una introducción al resto de documento, exponiendo los orígenes y los objetivos del proyecto. Además, contiene un glosario con los términos relevantes que aparecen a lo largo del documento.
- **Capítulo 2: Estado del arte.** En este capítulo expondremos el contexto en el cual nace nuestra plataforma, así como las diferentes aplicaciones que han ido surgiendo en los últimos años y que han servido de inspiración para la nuestra.
- **Capítulo 3: Plan de gestión de proyecto.** Contiene todo lo relacionado con la planificación del proyecto, principalmente las fases y los costes del mismo.

- **Capítulo 4: Requisitos del sistema.** Se presentarán todos los requisitos de nuestro proyecto.
- **Capítulo 5: Análisis del sistema.** Contiene el modelo conceptual de la plataforma, y los casos de uso de la misma.
- **Capítulo 6: Diseño del sistema.** En este capítulo, se detallan todos los aspectos relacionados con la arquitectura del sistema.
- **Capítulo 7: Construcción del sistema.** Expondremos los detalles de implementación del sistema, mencionando las herramientas empleadas para su desarrollo.
- **Capítulo 8: Pruebas del sistema.** Se describirán las pruebas realizadas a nuestro proyecto.
- **Capítulo 9: Conclusiones.** En este apartado se presentarán los resultados obtenidos con el proyecto, además de algunas reflexiones sobre la plataforma en sí, los objetivos cumplidos y el trabajo futuro.
- **Anexo A: Manual del desarrollador:** Guía detallada para la puesta a punto del entorno de trabajo y de algunos detalles del entorno de producción.
- **Anexo B: Manual de usuario:** Manual de referencia para los profesores y alumnos de la plataforma.

2. Estado del arte

Actualmente existen diversas plataformas en el mercado que ofrecen la posibilidad de estudiar contenidos teóricos de una materia concreta mediante la utilización de cuestionarios, tarjetas de vocabulario, diagramas, entre otros. Estas plataformas se han vuelto cada vez más populares gracias a la facilidad de uso que brindan y a la flexibilidad que ofrecen en términos de acceso a los materiales de aprendizaje. Para el desarrollo de este proyecto, se han tenido en cuenta tanto casos de éxito en universidades con plataformas de aprendizaje como las herramientas populares que más se asemejan a nuestra idea:

2.1. Guess it!

Guess it! es un juego para dispositivos Android desarrollado por la Universidad de Cádiz. En él, los usuarios deben intentar adivinar el término que falta en las frases incompletas que le aparecen. A partir de un cierto número de términos jugados, se le pedía al usuario introducir una frase incompleta propia, que posteriormente podrán contestar sus compañeros. De esta manera, se fomentaba la participación del alumnado en la creación de ejercicios. Esta aplicación se usó para ver el impacto que tenía en el aprendizaje de vocabulario en alumnos de Filología Alemana. Se consiguieron muchas partidas por parte de los alumnos, y se observó una mejora considerable en su vocabulario gracias a la personalización del orden de las preguntas a cada usuario mediante la generación automática de tablas estáticas en las que se calculaba un valor para cada usuario y pregunta [22, 21, 4]. La gran ventaja de este proyecto es que los alumnos podían añadir nuevos ejercicios a la plataforma sin la intervención de un profesor, el cual solo tenía que intervenir si alguna era valorada de forma negativa por otros alumnos. No obstante, era una aplicación enfocada única y exclusivamente al aprendizaje de idiomas, y aunque se fomentaba la creación de ejercicios, no había ninguna recompensa para aquellos usuarios que creasen preguntas más elaboradas, por lo que muchas de ellas eran de poca calidad.

2.2. VirUCA

VirUCA es un juego de preguntas similar a *Trivial*, desarrollado por la Universidad de Cádiz. Se trata de una aplicación web en la que los estudiantes juegan por equipos, teniendo que responder a preguntas que previamente habían creado otros estudiantes. Permitía disponer de varias asignaturas dentro de la plataforma. *VirUCA* fue probado con alumnos de la asignatura de Virología del Grado en Biotecnología de la Universidad de Cádiz, en la que se observó un incremento en la motivación del alumnado [6]. Uno de los principales problemas de esta aplicación era que todas las preguntas creadas por los alumnos debía ser validadas por el profesor de forma manual. Además,

los estudiantes únicamente podían jugar desde clase. Por último, algunos alumnos hubiesen preferido que el juego no fuese por turnos, sino que participasen todos al mismo tiempo para que el desarrollo de la partida fuese más dinámico.

2.3. SMCP-Training

SMCP-Training es una herramienta virtual para el aprendizaje de las frases normalizadas para las comunicaciones marítimas (en inglés Standard Marine Communication Phrases o SMCP) creada por la Universidad de Cádiz para dos asignaturas impartidas en la Escuela de Ingenierías Marina, Náutica y Radioelectrónica [3]. Esta plataforma ofrecía una serie de preguntas de distintas categorías gestionables por el profesor (e.g. mensajes de aviso, informes o establecimiento de un diálogo), que a su vez podían ser de distinto tipo (e.g. de opción múltiple o de escribir la respuesta, entre otros). La plataforma constaba de varios modos de entrenamiento con los que el alumno podía progresar en su aprendizaje, y de un modo examen que, utilizando un algoritmo de repetición espaciada (SRA), permitía gestionar la repetición de los exámenes de forma automática para adaptarse a las necesidades del alumno. Los profesores por su parte podían ver todos los entrenamientos de los alumnos. Los resultados de esta plataforma fueron satisfactorios, ya que el rendimiento de los alumnos mejoró, y aunque no pudo afirmarse que esa mejora fue producida exclusivamente por la plataforma, la mayoría de los alumnos contestó afirmativamente a la pregunta de si pensaban que la herramienta influyó en las calificaciones que obtuvieron.

2.4. StudentQuiz

StudentQuiz¹ es un plugin de Moodle creado por la Universidad de Ciencias Aplicadas Rapperswil [20]. Gracias a este plugin, se pueden crear en un curso actividades en las que profesores y estudiantes creen un banco de preguntas conjunto, que posteriormente podrán usar para realizar cuestionarios. Permite que las preguntas sean de diversos tipos (preguntas de opción múltiple, de verdadero/falso, de relacionar elementos, etc.), y es capaz de personalizar el cuestionario para cada usuario, según la dificultad con la que perciba las preguntas. La plataforma permite a los profesores evaluar el rendimiento de los usuarios y proporciona retroalimentación en tiempo real. Es una herramienta *open source* y fácil de usar, pero no está diseñada para ser usada desde dispositivos móviles.

2.5. Anki

Anki² es una plataforma de estudio basada en tarjetas (*flashcards*) que utiliza un algoritmo de repetición espaciada (SRA) con el que programa el estudio de las

¹StudentQuiz - https://moodle.org/plugins/mod_studentquiz

²Anki - <https://apps.ankiweb.net/>

preguntas según la dificultad percibida por el usuario. Es una herramienta *open source*, de la que existe tanto versión de escritorio como aplicación para dispositivos móviles. Permite varios tipos de preguntas. Varios estudios han concluido en que el uso de esta aplicación es capaz de potenciar la retención de la información en estudiantes [24, 19]. No obstante, es una aplicación pensada para el estudio individual, y si bien es cierto que pueden importarse y exportarse los bancos de preguntas para compartirlos, tampoco existe la figura del profesor para monitorizar el aprendizaje del alumno.

2.6. Comparativas

El objetivo de este proyecto es tomar las ventajas de cada uno de los proyectos anteriores, e intentar mejorar o proponer una alternativa en los puntos débiles de cada uno, además de aportar alguna otra característica innovadora y que destaque nuestro proyecto frente al resto. En las Tablas 2.1 y 2.2 se recopilan las características que deseamos implementar en nuestro proyecto y su presencia en los proyectos previamente analizados.

Tabla 2.1

Comparativa de características de las plataformas estudiadas I

Plataforma	¿Soporta preguntas de diversas materias?	¿Permite a los alumnos crear preguntas directamente en el banco de preguntas?	¿Es capaz de personalizar el orden de las preguntas a cada usuario de forma automática?	¿Permite el estudio con un banco de preguntas compartido?
Guess it!	Sí ³	Sí	Sí ⁴	Sí
VirUCA	Sí	No	No	Sí
SMCP-Training	Sí	No	Sí	Sí
StudentQuiz	Sí	Sí	No ⁵	Sí
Anki	Sí	Sí	Sí	No ⁶

³Guess It! permitía ejercicios de temas predefinidos, el profesor no podía crear los temas

⁴Guess It! no personaliza el orden en tiempo real, lo hace actualizando los valores de las tablas estáticas diariamente.

⁵StudentQuiz personaliza el orden basándose en la valoración que el usuario o el resto del grupo haya otorgado a cada pregunta.

⁶Anki no permite de forma natural tener bancos de preguntas compartidos. No obstante, existen módulos como [1], que permite simular esto utilizando archivos creados con Google Docs.

⁷Guess It! está disponible para PC y para dispositivos Android.

⁸Anki no permite valorar preguntas ya que el usuario tiene pleno control sobre las preguntas que estudia. No obstante, puede valorar mazos de preguntas creados por otras personas en [2]

Tabla 2.2

Comparativa de características de las plataformas estudiadas II

Plataforma	¿Los usuarios pueden valorar las preguntas creadas por otros usuarios?	¿Funciona en todos los dispositivos?	¿Existe la figura del profesor para gestionar y monitorizar el contenido de la plataforma?
Guess it!	Sí	No ⁷	Sí
VirUCA	No	Sí	Sí
SMCP-Training	No	Sí	Sí
StudentQuiz	Sí	No	Sí
Anki	No ⁸	Sí	No

3. Plan de gestión de proyecto

3.1. Metodología de desarrollo

Para este proyecto hemos decidido usar una metodología de desarrollo iterativa incremental. Esta metodología se basa en dividir el proyecto en pequeñas iteraciones o ciclos, que constan de una serie de tareas que incluyen el análisis, diseño, implementación, pruebas y entrega de una funcionalidad específica.

Se han realizado diversas iteraciones para crear las distintas funcionalidades del sistema, pero las podríamos clasificar en dos grupos distintos: Las iteraciones relacionadas con el “backend” de la plataforma, y las relacionadas con el “frontend” (que hacían uso de lo que habíamos desarrollado en las iteraciones del primer grupo).

3.2. Planificación del proyecto

El desarrollo de este proyecto se puede dividir en una serie de fases, que describimos a continuación:

3.2.1. Iteración 0: Fase inicial

En esta fase, se tuvo una primera toma de contacto con la idea original del proyecto. También se seleccionaron las tecnologías y lenguajes a usar en el mismo.

3.2.2. Iteración 1: Análisis inicial

Se recabó información acerca de las plataformas ya existentes, y analizando los pros y contras de todas ellas se seleccionaron las mejores características para nuestro proyecto. Se establecieron los requisitos de la plataforma.

3.2.3. Iteración 2: Aprendizaje de tecnologías

Para llevar a cabo un proyecto de calidad, se seleccionaron tecnologías y herramientas en función de los requisitos del proyecto, y no de los conocimientos del desarrollador. Además, los conocimientos previos que se tenían de programación web no eran lo suficientemente amplios como para llevar un proyecto de esta envergadura. Por ese mismo motivo, era necesaria una fase en la que se aprendiesen los conceptos básicos de los frameworks, lenguajes, y software utilizados.

3.2.4. Iteración 3: Creación de la API

En esta fase se ha desarrollado una primera versión de la API de Django que nos permitiese comenzar el desarrollo de la web. Se realizó un prototipo completo de la API y se creó la base de datos de la plataforma. También se realizaron las pruebas pertinentes para acreditar que las funcionalidades de la plataforma realizaban las acciones esperadas.

3.2.5. Iteración 4: Desarrollo de la web

Una vez desarrollado el prototipo de la API, comenzamos a desarrollar la web en NextJS. Se diseñó la interfaz gráfica de la web, se seleccionaron los componentes y elementos necesarios, y se creó la comunicación web-API. Esta fase incluye las pruebas de la interfaz gráfica y de navegación de la web.

3.2.6. Iteración 5: Correcciones en la API

A medida que se iba desarrollando la web, se detectaron varios problemas de diseño y de lógica en la API. Esto obligó a realizar otra iteración sobre la API, llevada en paralelo con el desarrollo de la web para tratar de solucionar estos fallos. Al igual que en las iteraciones anteriores, se incluyen aquí las nuevas pruebas realizadas a la API.

3.2.7. Iteración 6: Despliegue del proyecto

La plataforma fue desplegada en un servidor de la propia Universidad de Cádiz. Se realizó una vez se dio por terminado el desarrollo de la web.

3.2.8. Iteración 7: Creación de la PWA

Una vez el código del proyecto estaba terminado, se realizaron los cambios necesarios para crear una PWA a partir de la web.

3.2.9. Redacción de la memoria

Como su propio nombre indica, en esta fase hemos redactado la documentación de este trabajo, es decir, este mismo documento. Esta tarea se ha realizado en paralelo a las demás.

3.3. Diagrama de Gantt

El diagrama de Gantt del proyecto se muestra en las figuras 3.1, 3.2 y 3.3

3.4. Costes

Para la estimación del coste total del proyecto debemos estimar los costes humanos y los materiales.

3.4.1. Costes humanos

Para este proyecto, se han dedicado aproximadamente 25 horas semanales de media. Como el proyecto se ha realizado en 30 semanas, se han dedicado unas 750 horas en total. Solamente ha trabajado en el proyecto un único desarrollador, que se ha encargado de todas las tareas descritas anteriormente.

Los primeros cuatro meses del proyecto (17 semanas), se realizaron a través de una beca de la Universidad de Cádiz, en la que el desarrollador tenía una remuneración bruta de 246.21 € mensuales. Por otro lado, también se ha obtenido la Beca de Colaboración del Ministerio de Educación y Formación Profesional, el cual vincula al desarrollador durante siete meses con este proyecto, es decir, durante las 30 semanas de duración del mismo. La cantidad percibida por dicha beca asciende a 2000 €.

$$C_{humano} = 246,21 \text{ €} * 4 \text{ meses} + 2000 \text{ €} = 2984,84 \text{ €}$$

3.4.2. Costes Materiales

Este proyecto ha sido desarrollado en un portátil HP Pavilion del 2018, que fue comprado a finales de 2019 por un precio de 819 €. Suponiendo que el portátil llega a tener una vida útil de cinco años, el coste amortizado es de 94 €.

$$Amortiz_{laptop} = \frac{819 \text{ €}}{5 \text{ años} * 365 \text{ días}} * (30 \text{ semanas} * 7 \text{ días}) = 94,24 \text{ €}$$

A esto hay que añadir 29.90 € mensuales por la fibra óptica de internet. Todo el software utilizado es gratuito, y se asume que el precio de alojar la plataforma es gratuito, pues se alojará en un servidor de la Universidad de Cádiz.

$$C_{material} = Amortiz_{laptop} + C_{fibra} = 94,24 \text{ €} + 29,90 \text{ €} * 7 \text{ meses} = 303,54 \text{ €}$$

Por tanto, teniendo en cuenta los costes anteriores, el coste total del proyecto ha sido 3288.38 €.

$$C_{total} = C_{humano} + C_{material} = 2984,84 \text{ €} + 303,54 \text{ €} = 3288,38 \text{ €}$$

	SEMANA 1	SEMANA 2	SEMANA 3	SEMANA 4	SEMANA 5	SEMANA 6	SEMANA 7	SEMANA 8	SEMANA 9	SEMANA 10
Iteración 0 - Selección de los recursos necesarios para el proyecto										
Iteración 1 - Viabilidad del proyecto y estudio del mercado										
Iteración 1 - Recopilación de requisitos										
Iteración 2 - Aprendizaje Django & Django REST Framework										
Iteración 2 - Aprendizaje NextJS y ReactJS										
Iteración 2 - Aprendizaje PWA										
Iteración 3 - Creación de la API										
Iteración 4 - Desarrollo de la web										
Iteración 5 - Correcciones en la API										
Iteración 6 - Despliegue										
Iteración 7 - Creación PWA										
Documentación - Redacción de la memoria										

Figura 3.1: Diagrama de Gantt I

	SEMANA 11	SEMANA 12	SEMANA 13	SEMANA 14	SEMANA 15	SEMANA 16	SEMANA 17	SEMANA 18	SEMANA 19	SEMANA 20
Iteración 0 - Selección de los recursos necesarios para el proyecto										
Iteración 1 - Viabilidad del proyecto y estudio del mercado										
Iteración 1 - Recopilación de requisitos										
Iteración 2 - Aprendizaje Django & Django REST Framework										
Iteración 2 - Aprendizaje NextJS y ReactJS										
Iteración 2 - Aprendizaje PWA										
Iteración 3 - Creación de la API										
Iteración 4 - Desarrollo de la web										
Iteración 5 - Correcciones en la API										
Iteración 6 - Despliegue										
Iteración 7 - Creación PWA										
Documentación - Redacción de la memoria										

Figura 3.2: Diagrama de Gantt II

	SEMANA 21	SEMANA 22	SEMANA 23	SEMANA 24	SEMANA 25	SEMANA 26	SEMANA 27	SEMANA 28	SEMANA 29	SEMANA 30
Iteración 0 - Selección de los recursos necesarios para el proyecto										
Iteración 1 - Viabilidad del proyecto y estudio del mercado										
Iteración 1 - Recopilación de requisitos										
Iteración 2 - Aprendizaje Django & Django REST Framework										
Iteración 2 - Aprendizaje NextJS y ReactJS										
Iteración 2 - Aprendizaje PWA										
Iteración 3 - Creación de la API										
Iteración 4 - Desarrollo de la web										
Iteración 5 - Correcciones en la API										
Iteración 6 - Despliegue										
Iteración 7 - Creación PWA										
Documentación - Redacción de la memoria										

Figura 3.3: Diagrama de Gantt III

Parte II

Desarrollo

4. Requisitos del Sistema

Antes de la creación de la plataforma, debemos tener una idea de cómo queremos que se comporte. Por ello, necesitamos plasmar las ideas del cliente (en este caso, los propios profesores, sin olvidarnos de pensar también en las necesidades del alumno) en unos requisitos que representen fielmente lo que queremos conseguir. En las siguientes secciones, se detallan estos requisitos.

4.1. Objetivos de Negocio

El proyecto que se va a desarrollar tiene como objetivo principal mejorar la calidad de la educación y el aprendizaje de conceptos teóricos de los alumnos. En este caso concreto, se va a comprobar si cumplimos este objetivo con la parte teórica de la asignatura de Bases de Datos del Grado de Ingeniería Informática de la Universidad de Cádiz, pero la misma aplicación podría enfocarse en cualquier otra materia, puesto que el tema de las preguntas será totalmente configurable por el profesor.

También se buscará que el alumno no solo aprenda, sino que también se divierta aprendiendo. Para ello, se ofrecerá una experiencia de aprendizaje innovadora y atractiva que aumente la satisfacción de los estudiantes. Lo que se pretende es que, cuando el alumno quiera prepararse de cara a los exámenes de la asignatura, considere que *ROBDA* es una de las herramientas más útiles que pueda tener a su alcance.

Asimismo, se pretende ofrecer un sistema que delegue responsabilidades en el alumnado, para que ellos mismos puedan participar también en la creación y gestión las preguntas, todo ello bajo la supervisión del profesor, que gracias a esto podrá centrarse más en las tareas de monitorización.

4.2. Objetivos del Sistema

Vamos a plantear un sistema que satisfaga las necesidades de los profesores y de los alumnos, aprendiendo de los puntos fuertes y débiles de las plataformas analizadas en la Sección 2.

4.2.1. Visión general del proyecto

ROBDA dispondrá de varios modos de juego, pero en todos el desarrollo de la partida es el mismo: responder preguntas de tipo test relacionadas con una materia en concreto. Para que la plataforma sea compatible con la modalidad de docencia bilingüe, podremos asignar un idioma a las preguntas creadas, para que a la hora de realizar los

cuestionarios el alumno elija realizar el cuestionario en el idioma que desee. Los modos de juego que crearemos en la plataforma serán los siguientes:

- Repaso: El modo de juego más parecido a *Guess It!*. Consiste en un cuestionario infinito sobre preguntas de un tema e idioma elegidos previamente por el usuario. Las preguntas que le aparecen al usuario tendrán una probabilidad de aparición para personalizar cada repaso a las necesidades de cada alumno. Esta probabilidad de aparición se calculará en tiempo real, y vendrá determinada por una serie de factores como la dificultad de la pregunta para el usuario o la distancia respecto a la última aparición de la pregunta. Los alumnos deberán valorar la dificultad de cada pregunta que respondan, y tendrán la posibilidad de reportar preguntas que consideren erróneas.

Este modo de juego busca generar el mayor número de preguntas respondidas por los alumnos, ofreciéndoles una jugabilidad pensada para repasar de cara a los exámenes de la asignatura.

- Competiciones: Los estudiantes participan en competiciones creadas por el profesor sobre un tema e idioma concretos, en las que tendrán que crear una pregunta a modo de llave de acceso. Cuando termine la fase de creación de preguntas, tendrán que responder un cuestionario con límite de tiempo (establecido en 15 minutos) con preguntas creadas en la misma competición por otros usuarios (se seleccionan 10 preguntas al azar entre las creadas. Si no hay suficientes, se toman preguntas disponibles en el resto de modos de juego). Una vez realizado, los alumnos podrán reportar las preguntas que consideren erróneas, y valorar el propio evento. Se consiguen 50 puntos si la pregunta creada es correcta (es decir, si nadie ha reportado ningún error en la pregunta que haya sido validado por el profesor), 10 puntos por cada acierto del cuestionario, 15 por reportar preguntas erróneas y 20 por valorar la competición. Finalmente, se mostrará una clasificación con los mejores jugadores de la competición. Las fases del juego comienzan y terminan a las horas indicadas por el profesor en el proceso de creación de la competición.

Este modo de juego de juego tiene varios objetivos: Por un lado, se pretende motivar a los usuarios a crear preguntas que se hayan trabajado, ya que crear una pregunta errónea no le generará puntos, y crear una pregunta muy simple hará que los compañeros consigan puntos más fácilmente. De esta forma, intentamos que los alumnos creen preguntas de calidad para la plataforma, que posteriormente podamos usar en el resto de modos de juego. Además, también se recompensan los reportes de las preguntas, aspecto muy importante ya que las preguntas a responder estarán recién creadas. Así, podríamos considerar que en este modo de juego se realiza una criba de las preguntas erróneas realizada por los propios alumnos, y necesitando únicamente la intervención del profesor para decidir si los reportes son válidos o no. Además de quitar carga de trabajo al profesor, también le permite tener un mayor control sobre las preguntas creadas por los alumnos, pues será él quien determine cuándo pueden los alumnos crear preguntas, sobre qué tema, y en qué idioma, según la configuración de la competición.

- Duelos: En este modo de juego, un usuario podrá retar a un compañero a realizar el mismo cuestionario, sobre un tema e idioma indicados por el retador. El

cuestionario tendrá 10 preguntas, y 15 minutos de duración. Gana el duelo quien consiga más aciertos en el cuestionario.

Con este modo de juego ofrecemos una experiencia entretenida para usar en las mismas sesiones de la asignatura, en las que podremos retar a los compañeros que tengamos al lado. Al igual que en el resto de modos de juego, el alumno también podrá reportar las preguntas.

Para realizar un reporte, el alumno deberá seleccionar el motivo entre los que se ofrecen en el formulario y una breve descripción. Una vez enviado el reporte, el profesor podrá ver una notificación en el panel de preguntas, indicando que la pregunta se ha reportado. Accediendo al detalle de la pregunta, el profesor decidirá si esos reportes proceden o no, y se podrá modificar o eliminar la pregunta si lo considera necesario.

Por su parte, el profesor también podrá gestionar el resto del contenido de la plataforma, así como ver estadísticas de las partidas por modos de juego y de las preguntas. También podrá ver todas las partidas de los alumnos en detalle.

4.3. Requisitos funcionales

Los principales requisitos funcionales de la plataforma son los siguientes:

1. RF1: Inicio de sesión.
2. RF2: Registro.
3. RF3: Cierre de sesión desde cualquiera de los menús.
4. RF4: Ver créditos e información del proyecto.
5. RF5: Edición de los datos del perfil.
6. RF6: Navegación entre pantallas intuitiva mediante una barra lateral
7. RF7: Creación de dos roles principales (profesor y alumno).
8. RF8: Gestión de los temas (profesor).
9. RF9: Gestión de las imágenes (profesor).
10. RF10: Gestión de las preguntas y de los reportes (profesor).
11. RF11: Gestión de las competiciones (profesor).
12. RF12: Visualización de las partidas y de estadísticas (profesor).
13. RF13: Jugar a “Repaso” (alumno).
14. RF14: Jugar a “Competición” (alumno).
15. RF15: Jugar a “Duelo” (alumno).

Como veremos más adelante, estos requisitos representan las acciones que los usuarios pueden realizar de forma abstracta. Muchas de ellas (principalmente las que tienen que ver con jugar partidas) pueden descomponerse en otras tareas.

4.4. Requisitos no funcionales

La plataforma debe cumplir con unos requisitos que garanticen la calidad de nuestro software:

- RNF1: La plataforma debe ser compatible con cualquier dispositivo que cuente con alguno de los sistemas operativos más populares en 2023:
 - Windows 10 o superior.
 - Android 13 o superior.
 - Linux Ubuntu LTS 20.04 o superior.
 - iOS 16 o superior.
 - macOS Ventura o superior.
- RNF2: La plataforma será compatible con los principales navegadores gráficos actualizados a versiones lanzadas en 2023 o años posteriores.
 - Google Chrome 109 o superior.
 - Mozilla Firefox 109 o superior.
 - Microsoft Edge 109 o superior.
 - Safari 16.3 o superior.
- RNF3: El sistema debe garantizar que los datos recopilados no puedan ser accedidos por un tercero, por lo que se deberán cifrar los datos sensibles. Además, se debe procurar que los cuestionarios sean fiables, es decir, que los alumnos no puedan hacer algún truco con el que modificar los resultados, o manipularlos a su favor. Asimismo, se implementarán mecanismos de seguridad esenciales para los ataques más comunes.
- RNF4: El sistema debe ser lo suficientemente eficiente y escalable como para que el usuario no se vea perjudicado, sobre todo en medio de un cuestionario con cuenta atrás. En este aspecto, se pretende que el rendimiento de la plataforma sea similar al de otras plataformas educativas de la universidad, y que sea capaz de manejar una carga de al menos 100 usuarios de forma concurrente.
- RNF5: El servicio debe estar disponible las veinticuatro horas del día para los usuarios, con una disponibilidad mínima del 99 %.

5. Análisis del Sistema

5.1. Modelo Conceptual de Datos

Para el diseño de los datos, tenemos que tener en cuenta qué datos debemos guardar y la manera en la que se guardan, de forma que los usuarios puedan acceder a ellos de forma rápida.

Tras la obtención de los requisitos de la plataforma, creamos un diagrama EER inicial con el que tener una idea de la estructura de la base de datos (Figura 5.1). Este diagrama contiene los elementos más relevantes de la plataforma, así como la mayoría de relaciones que existirán entre ellos.

Cabe destacar algunas características modeladas con este diagrama, junto con algunas restricciones:

- Cualquier usuario, independientemente de su rol puede crear preguntas. No obstante, los alumnos solo podrán crearlas participando en las competiciones, que determinarán el tema y el idioma de dicha pregunta.
- Como se puede apreciar en el diagrama, las preguntas pueden tener opcionalmente una imagen, pero no es necesario que la tenga, pero una pregunta debe tener siempre un tema asignado.
- Se han modelado en dos entidades distintas la pregunta y sus opciones. Esto se ha hecho por motivos de escalabilidad: Ahora mismo, se ha limitado por simplicidad el número de opciones a 4, y solo una de ellas debe ser correcta. Modelándolo así, permitimos en un futuro poder ampliar a un número de opciones variable (e.g. entre 2 y 7) y un número variable de respuestas correctas (para crear preguntas de múltiple opción correcta).
- La relación entre las partidas y las preguntas deberá contener información acerca de la respuesta del usuario.
- Para la personalización del cuestionario a los alumnos, deberá guardarse en la base de datos información para determinar cómo de recomendable es una pregunta para un usuario concreto.

Por motivos de claridad, se han obviado los campos de cada entidad y las relaciones que abarcan la gestión de las preguntas, competiciones, temas, imágenes y reportes por parte del profesor; así como la creación de reportes por parte del alumno. Añadiendo esta información y las cuestiones a tener en consideración citadas anteriormente, se obtiene el diseño final de la base de datos (Véase la Sección 6.2).

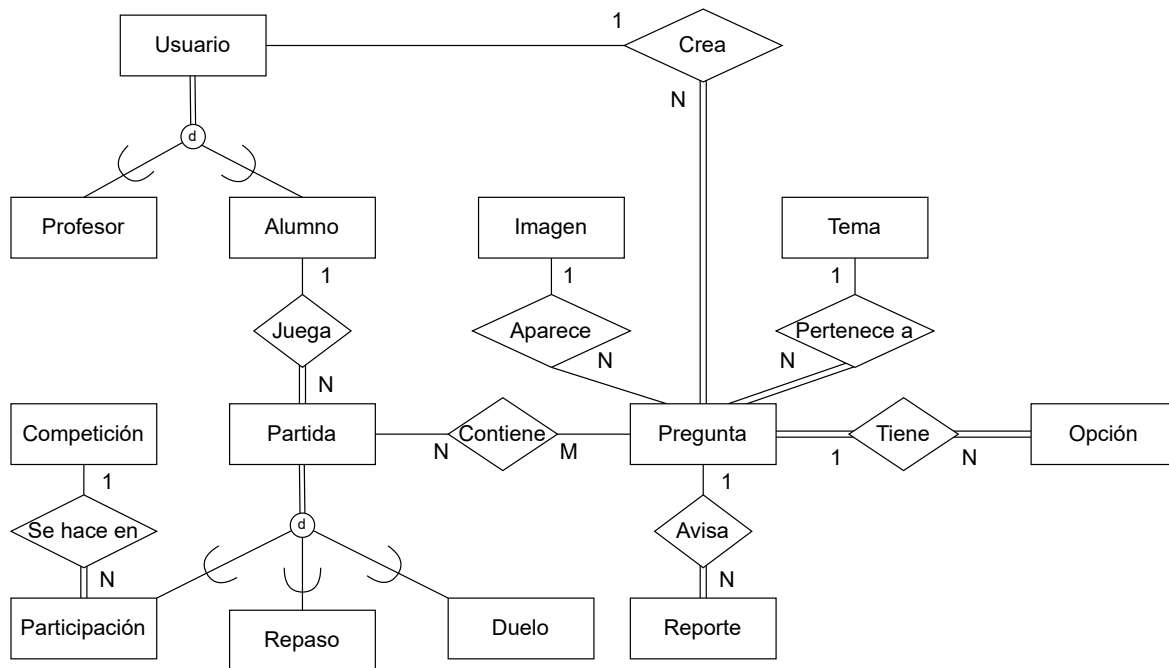


Figura 5.1: Diagrama Entidad-Relación Extendido

5.2. Modelo de Casos de Uso

5.2.1. Diagrama de casos de uso

En la plataforma podemos clasificar a los usuarios en dos roles distintos: Por un lado, los profesores, que se encargarán de la gestión y monitorización, y por otro los alumnos, que principalmente jugarán partidas. Ambos acceden a la plataforma de la misma manera. En la figura 5.2 podemos ver un diagrama de casos de uso con las funcionalidades del sistema, sin entrar en detalle en las distintas acciones que componen cada una. En las próximas subsecciones, veremos en profundidad todas ellas.

5.2.2. Descripción de caso de uso: Inicio de sesión

- **Descripción:** El usuario se identifica correctamente en el sistema.
- **Actores:** Usuario.
- **Precondiciones:** Ninguna.
- **Postcondiciones:** El usuario accede al menú principal de la plataforma.
- **Escenario principal:**
 1. El sistema muestra la pantalla de inicio de sesión.
 2. El usuario introduce su nombre de usuario y su contraseña.
 3. El sistema valida los datos.

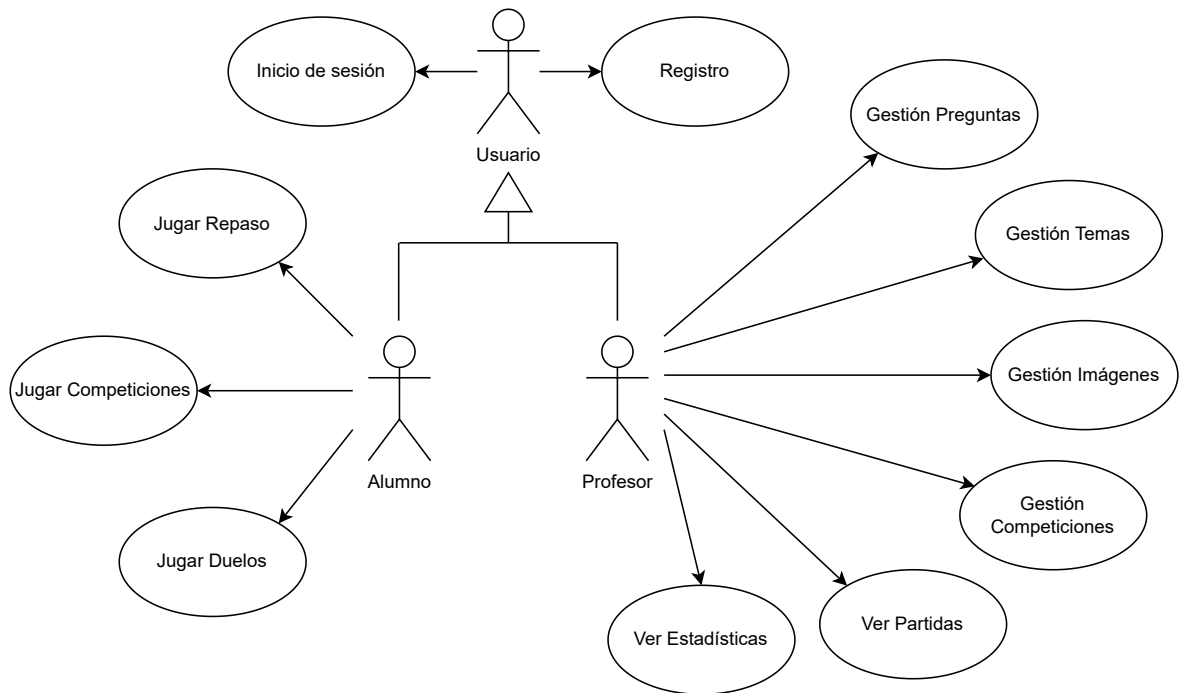


Figura 5.2: Diagrama de casos de uso de alto nivel

4. El usuario es redirigido al menú principal.

■ **Extensiones - Flujo alternativo:**

- 3a. El sistema comprueba que no existe un usuario registrado con ese nombre y contraseña, y pide al usuario reintroducir los datos.

5.2.3. Descripción de caso de uso: Registro

■ **Descripción:** El usuario crea una cuenta en el sistema.

■ **Actores:** Usuario.

■ **Precondiciones:** Ninguna.

■ **Postcondiciones:** El usuario se registra exitosamente y es redirigido a la pantalla de inicio de sesión.

■ **Escenario principal:**

1. El sistema muestra la pantalla de registro.
2. El usuario introduce su nombre, email y contraseña.
3. El sistema valida los datos ingresados.
4. El sistema crea una cuenta para el usuario.
5. El usuario es redirigido a la pantalla de inicio de sesión

■ **Extensiones - Flujo alternativo:**

- 3a. El sistema detecta que el email ingresado o el nombre de usuario ya está asociado a otra cuenta existente y muestra un mensaje de error al usuario.
- 3b. El sistema detecta que la contraseña no cumple con los requisitos mínimos y muestra un mensaje de error al usuario.

5.2.4. Descripción de caso de uso: Gestión Temas

La gestión de las temas consiste en la creación y edición de los temas.

5.2.4.1. Caso de uso: Creación de temas

- **Descripción:** El profesor crea un tema en el sistema.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema.
- **Postcondiciones:** El tema se crea exitosamente en el sistema.
- **Escenario principal:**
 1. El profesor accede a la pantalla de creación de temas.
 2. El sistema muestra el formulario de creación de temas.
 3. El profesor ingresa el nombre del tema.
 4. El profesor pulsa el botón *Crear*, y el sistema valida el tema.
 5. El sistema reinicia el formulario y muestra un mensaje informativo por pantalla.
- **Extensiones - Flujo alternativo:**
 - 3a. El profesor no ha ingresado un nombre válido y el sistema muestra un mensaje de error.
 - 2-3a. El profesor cancela la creación del tema y el sistema le redirige a la pantalla principal de temas.

5.2.4.2. Caso de uso: Modificación de tema

- **Descripción:** El profesor modifica un tema en el sistema.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema y existe al menos un tema creado.
- **Postcondiciones:** El tema se modifica exitosamente en el sistema.
- **Escenario principal:**

1. El profesor accede al detalle de un tema.
2. El sistema muestra los detalles del tema seleccionado en un formulario.
3. El profesor realiza las modificaciones necesarias en el nombre del tema.
4. El profesor pulsa el botón *Guardar*, y el sistema valida los cambios realizados y muestra un mensaje informativo por pantalla.

■ **Extensiones - Flujo alternativo:**

- 3a. El profesor decide no modificar el tema y regresa a la pantalla principal de temas sin realizar cambios pulsando el botón *Volver a temas*.
- 3b. El profesor pulsa el botón *Eliminar*.
 1. El sistema muestra un diálogo de confirmación por pantalla.
 2. a) El profesor confirma la acción pulsando en *Eliminar*, y el sistema realiza una eliminación en cascada de todo lo que depende del tema indicado.
 - b) El profesor pulsa *Cancelar* o en un punto fuera del diálogo, y se cancela la acción de eliminar.
- 4a. El profesor ha realizado cambios inválidos en el tema (nombre nulo o ya existente) y el sistema muestra un mensaje de error.

5.2.5. Descripción de caso de uso: Gestión Imágenes

La gestión de imágenes consiste en la creación y edición de imágenes.

5.2.5.1. Caso de uso: Creación de imágenes

- **Descripción:** El profesor sube una imagen al sistema.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema.
- **Postcondiciones:** La imagen se crea exitosamente en el sistema.
- **Escenario principal:**
 1. El profesor accede a la pantalla de creación de imágenes.
 2. El sistema muestra el formulario de creación de imágenes.
 3. El profesor ingresa el nombre de la imagen, el tema que representa y sube la propia imagen.
 4. El profesor pulsa el botón *Crear*, y el sistema valida la imagen.
 5. El sistema reinicia el formulario y muestra un mensaje informativo por pantalla.

■ **Extensiones - Flujo alternativo:**

- 3a. El profesor no ha ingresado un nombre válido y el sistema muestra un mensaje de error.
- 3b. El profesor no ha incluido una imagen válida y el sistema muestra un mensaje de error.
- 2-3a. El profesor cancela la creación de la imagen y el sistema le redirige a la pantalla principal de imágenes.

5.2.5.2. Caso de uso: Modificación de imagen

■ **Descripción:** El profesor modifica los datos de una imagen en el sistema.

■ **Actores:** Profesor.

■ **Precondiciones:** El profesor ha iniciado sesión en el sistema y existe al menos una imagen creada.

■ **Postcondiciones:** La imagen se modifica exitosamente en el sistema.

■ **Escenario principal:**

1. El profesor accede al detalle de una imagen.
2. El sistema muestra los detalles de la imagen seleccionada en un formulario.
3. El profesor realiza las modificaciones necesarias en los datos de la imagen.
4. El profesor pulsa el botón *Guardar*, y el sistema valida los cambios realizados y muestra un mensaje informativo por pantalla.

■ **Extensiones - Flujo alternativo:**

- 3a. El profesor decide no modificar la imagen y regresa a la pantalla principal de imágenes sin realizar cambios pulsando el botón *Volver a imágenes*.
- 3b. El profesor pulsa el botón *Eliminar*.
 1. El sistema muestra un diálogo de confirmación por pantalla.
 2. a) El profesor confirma la acción pulsando en *Eliminar*, y el sistema realiza una eliminación en cascada de todo lo que depende de la imagen indicada.
 - b) El profesor pulsa *Cancelar* o en un punto fuera del diálogo, y se cancela la acción de eliminar.
- 4a. El profesor ha realizado cambios inválidos en la imagen (nombre nulo o ya existente) y el sistema muestra un mensaje de error.

5.2.6. Descripción de caso de uso: Gestión Preguntas

La gestión de las preguntas se puede dividir en tres casos de usos principalmente: El relacionado con la creación de las preguntas, el relativo a la edición de una pregunta y la decisión sobre un reporte de una determinada pregunta.

5.2.6.1. Caso de uso: Creación de preguntas

- **Descripción:** El profesor crea una pregunta en el sistema.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema.
- **Postcondiciones:** La pregunta se crea exitosamente en el sistema.
- **Escenario principal:**
 1. El profesor accede a la pantalla de creación de preguntas.
 2. El sistema muestra el formulario de creación de preguntas.
 3. El profesor ingresa el enunciado de la pregunta.
 4. El profesor ingresa cuatro opciones de respuesta, la primera de ellas la única correcta.
 5. El profesor elige un tema para la pregunta.
 6. El profesor selecciona un idioma para la pregunta.
 7. El profesor pulsa el botón *Crear*, y el sistema valida la pregunta.
 8. El sistema reinicia el formulario y muestra un mensaje informativo por pantalla.
- **Extensiones - Flujo alternativo:**
 - 3a. El profesor no ha ingresado un enunciado válido y el sistema muestra un mensaje de error.
 - 4a. El profesor no ha ingresado las cuatro opciones de respuesta y el sistema muestra un mensaje de error.
 - 6a. El profesor desea adjuntar una imagen a la pregunta.
 1. El profesor selecciona la opción para adjuntar una imagen.
 2. El sistema muestra las imágenes relacionadas con el tema previamente marcado.
 3. El profesor elige una de las imágenes.
 4. El sistema muestra una previsualización de la imagen, y bloquea el campo del tema para que concuerde con el de la imagen seleccionada.

- 6b. El profesor decide no adjuntar una imagen a la pregunta y continúa sin realizar ninguna acción adicional.
- 2-6a. El profesor cancela la creación de la pregunta y el sistema le redirige a la pantalla principal de preguntas.

5.2.6.2. Caso de uso: Modificación de pregunta

- **Descripción:** El profesor modifica una pregunta existente en el sistema.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema y existe al menos una pregunta creada. No deben existir reportes activos para dicha pregunta.
- **Postcondiciones:** La pregunta se modifica exitosamente en el sistema.
- **Escenario principal:**
 1. El profesor accede al detalle de una pregunta y pulsa en el botón *Editar*.
 2. El sistema muestra los detalles de la pregunta seleccionada en un formulario.
 3. El profesor realiza las modificaciones necesarias en el enunciado, opciones de respuesta, tema, idioma o imagen.
 4. El profesor pulsa el botón *Guardar*, y el sistema valida los cambios realizados y muestra un mensaje informativo por pantalla.
- **Extensiones - Flujo alternativo:**
 - 3a. El profesor decide no modificar la pregunta y regresa al detalle de la pregunta sin realizar cambios pulsando el botón *Cancelar*.
 - 3b. El profesor pulsa el botón *Eliminar*.
 1. El sistema muestra un diálogo de confirmación por pantalla.
 2. a) El profesor confirma la acción pulsando en *Eliminar*, y el sistema realiza un *soft delete* de la pregunta.
 - b) El profesor pulsa *Cancelar* o en un punto fuera del diálogo, y se cancela la acción de eliminar.
 - 4a. El profesor ha realizado cambios inválidos en la pregunta (por ejemplo, enunciado vacío, opciones de respuesta incompletas, tema no válido, etc.) y el sistema muestra un mensaje de error.

5.2.6.3. Caso de uso: Decidir sobre reporte

- **Descripción:** El profesor decide si un reporte es correcto o no.
- **Actores:** Profesor.

- **Precondiciones:** El profesor ha iniciado sesión en el sistema y existe al menos una pregunta creada.
- **Postcondiciones:** El reporte cambia su estado en el sistema.
- **Escenario principal:**
 1. El profesor accede al detalle de una pregunta.
 2. El sistema muestra los detalles de la pregunta seleccionada junto con una lista de los reportes activos de la misma.
 3. El profesor pulsa en el botón *Aceptar reporte* para validar el reporte.
 4. El sistema muestra la lista actualizada de reportes, y un mensaje informativo.
- **Extensiones - Flujo alternativo:**
 - 2a. El sistema muestra los detalles de la pregunta seleccionada junto con una lista vacía de reportes. El caso de uso termina.
 - 3a. El profesor pulsa el botón *Rechazar reporte* para eliminar el reporte de la lista.

5.2.7. Descripción de caso de uso: Gestión Competiciones

La gestión de las preguntas está compuesta principalmente por tres casos de uso distintos: la creación de competencias, la modificación de competencias, y la finalización de competencias.

5.2.7.1. Caso de uso: Creación de competencias

- **Descripción:** El profesor crea una competición en el sistema.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema y existe al menos un tema en el sistema.
- **Postcondiciones:** La competición se crea exitosamente en el sistema.
- **Escenario principal:**
 1. El profesor accede a la pantalla de creación de competencias.
 2. El sistema muestra el formulario de creación de competencias.
 3. El profesor ingresa el nombre de la competición, el tema y el idioma.
 4. El profesor selecciona la fechas de inicio de la competición, y de finalización de las fases de la misma.
 5. El profesor pulsa el botón *Crear*, y el sistema valida la competición.

6. El sistema reinicia el formulario y muestra un mensaje informativo por pantalla.

■ **Extensiones - Flujo alternativo:**

- 3a. El profesor no ha ingresado un nombre válido y el sistema muestra un mensaje de error.
- 4a. El profesor no ha ingresado fechas válidas y el sistema muestra un mensaje de error.
- 2-4a. El profesor cancela la creación de la pregunta y el sistema le redirige a la pantalla principal de preguntas.

5.2.7.2. Caso de uso: Modificación de competición

■ **Descripción:** El profesor modifica una competición existente en el sistema.

■ **Actores:** Profesor.

■ **Precondiciones:** El profesor ha iniciado sesión en el sistema y existe al menos una competición creada. La competición no puede estar finalizada.

■ **Postcondiciones:** La competición se modifica exitosamente en el sistema.

■ **Escenario principal:**

1. El profesor accede al detalle de una competición y pulsa en el botón *Editar*.
2. El sistema muestra los detalles de la pregunta seleccionada en un formulario.
3. El profesor realiza las modificaciones necesarias en el nombre, tema, idioma y fechas.
4. El profesor pulsa el botón *Guardar*, y el sistema valida los cambios realizados y muestra un mensaje informativo por pantalla.

■ **Extensiones - Flujo alternativo:**

- 3a. El profesor decide no modificar la competición y regresa al detalle de la misma sin realizar cambios pulsando el botón *Cancelar*.
- 3b. El profesor pulsa el botón *Eliminar*.
 1. El sistema muestra un diálogo de confirmación por pantalla.
 2. a) El profesor confirma la acción pulsando en *Eliminar*, y el sistema elimina la competición.
b) El profesor pulsa *Cancelar* o en un punto fuera del diálogo, y se cancela la acción de eliminar.
- 3a. El profesor ha realizado cambios inválidos en la competición (por ejemplo, enunciado vacío, opciones de respuesta incompletas, tema no válido, etc.) y el sistema muestra un mensaje de error.

5.2.7.3. Caso de uso: Terminar competición

- **Descripción:** El profesor finaliza una de las competiciones.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema, no existen reportes asociados a preguntas creadas en la competición y la competición se encuentra en la fase *Esperando corrección del profesor*.
- **Postcondiciones:** La competición se finaliza.
- **Escenario principal:**
 1. El profesor pulsa el botón *Terminar* en la pantalla de detalle de competición.
 2. El sistema actualiza la competición, muestra un mensaje informativo y deja de permitir la edición de la competición.

5.2.8. Descripción de caso de uso: Ver estadísticas

- **Descripción:** El profesor ve las estadísticas generales de la plataforma.
- **Actores:** Profesor.
- **Precondiciones:** El profesor ha iniciado sesión en el sistema.
- **Postcondiciones:** El profesor ve distintas estadísticas de las preguntas, las partidas y los reportes de la plataforma.
- **Escenario principal:**
 1. El sistema muestra las estadísticas de la plataforma por pantalla.
- **Extensiones - Flujo alternativo:**
 - 1a. El profesor selecciona un tema o un idioma en una de las estadísticas.
 1. El sistema actualiza la estadística mostrada según el tema y el idioma indicados.
 - 1a. El profesor selecciona otro evento en una de las estadísticas
 1. El sistema actualiza la estadística mostrada según el evento indicado.

5.2.9. Descripción de caso de uso: Reportar pregunta

- **Descripción:** El alumno reporta una pregunta.
- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema, y existe al menos una pregunta.

- **Postcondiciones:** El sistema guarda un reporte asociado a la pregunta en cuestión.
- **Escenario principal:**
 1. El alumno pulsa el botón de *Reportar* en cualquiera de los modos de juego.
 2. El sistema muestra el formulario de reporte en un modal.
 3. El alumno indica el motivo y una breve justificación del reporte.
 4. El alumno confirma la acción pulsando en *Reportar*, y el sistema guarda el reporte y muestra un mensaje informativo por pantalla.
- **Extensiones - Flujo alternativo:**
 - 1-3a. El alumno pulsa *Cancelar* o en un punto fuera del modal, se cancela la acción de reportar y se muestra de nuevo el punto en el que se encontraba el alumno antes de comenzar el caso de uso.

5.2.10. Descripción de caso de uso: Repaso

- **Descripción:** El alumno juega una partida al modo de juego *Repaso*.
- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema.
- **Postcondiciones:** El alumno juega una partida de *Repaso* con un número indeterminado de preguntas de un tema e idioma elegidos.
- **Escenario principal:**
 1. El sistema muestra la pantalla inicial de *Repaso*, con un pequeño formulario.
 2. El alumno indica un idioma y un tema, y pulsa el botón *Jugar*.
 3. El sistema crea una partida.
 4. El sistema obtiene los valores de relevancia de cada pregunta y muestra al alumno la pregunta elegida mediante esos valores.
 5. El alumno selecciona una opción y pulsa el botón *Comprobar*.
 6. El sistema guarda la respuesta marcada, actualiza el valor de relevancia de la pregunta, revela la solución por pantalla y muestra las acciones de valorar pregunta y de reportar pregunta.
 7. El alumno valora la pregunta y pulsa el botón de *Enviar valoración*.
 8. El sistema muestra los botones de *Terminar repaso* y de *Continuar*.
 9. El alumno pulsa el botón *Terminar repaso*, finalizando así la partida, y el sistema le redirige a la pantalla principal de *Repaso*.
- **Extensiones - Flujo alternativo:**

- 3a. El sistema comprueba que no existen suficientes preguntas para la combinación de tema e idioma seleccionada, y muestra un error por pantalla.
- 7-8a. El alumno pulsa el botón de *Reportar* (*Extends: 5.2.9 Reportar pregunta*).
- 9a. El alumno pulsa el botón de *Continuar*, volviendo así al paso 4.

5.2.11. Descripción de caso de uso: Participación en competición

- **Descripción:** El alumno participa en una competición.
- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema, y existe al menos una competición creada por un profesor.
- **Postcondiciones:** El alumno visualiza la información del evento.
- **Escenario principal:**
 1. El sistema muestra la información de una competición.
- **Extensiones - Flujo alternativo:**
 - 1a. La competición está en la fase de creación de preguntas
 - El alumno no ha creado una pregunta.
 1. El sistema muestra un formulario para la creación de una pregunta (*Extends: 5.2.6.1 Crear pregunta*).
 - El alumno ya ha creado una pregunta.
 1. El sistema muestra la pregunta creada al alumno por pantalla.
 - 1b. La competición se encuentra en la fase de cuestionario.
 - El usuario ha creado la pregunta en la primera fase y no ha realizado el cuestionario aún.
 1. El sistema muestra un botón en la pantalla para iniciar el cuestionario.
 2. El alumno pulsa el botón e inicia la partida.
 3. El sistema selecciona preguntas del tema e idioma de la competición, priorizando las que otros alumnos hayan creado en la misma competición y muestra por pantalla la primera pregunta del cuestionario junto con una cuenta atrás.
 4. El alumno selecciona una opción y pulsa el botón *Siguiente*.

5. El sistema guarda la respuesta marcada y muestra por pantalla la siguiente pregunta, volviendo al paso 4 si la siguiente pregunta no es la última del cuestionario.
 6. El alumno selecciona una opción para la última pregunta y pulsa el botón *Finalizar*.
 7. El sistema guarda las respuestas marcadas y los tiempos de forma definitiva, y redirige al usuario a la pantalla de la participación, donde podrá ver su partida.
- El usuario ha creado la pregunta en la primera fase y ha realizado el cuestionario.
 1. El sistema muestra por pantalla la partida del alumno, junto con una petición de valoración del duelo.
 2. El alumno ve los detalles de su cuestionario.
 - ◇ El alumno selecciona una de las preguntas para reportarla (*Extends: 5.2.9 Reportar pregunta*).
 - ◇ El alumno otorga una puntuación para valorar el evento y el sistema registra la puntuación.
 - El alumno ha participado como mínimo en la fase de creación de preguntas, y la competición se encuentra esperando la corrección del profesor.
 1. El sistema muestra por pantalla la partida del jugador, sin posibilidad de realizar ninguna acción adicional.
 - El alumno ha participado como mínimo en la fase de creación de preguntas, y la competición se encuentra finalizada.
 1. El sistema muestra por pantalla a los mejores cinco jugadores de la competición junto con sus puntuaciones, así como el resumen de la partida del propio alumno y el desglose de su puntuación final.
 - 1c. El usuario no ha participado en la competición y la misma se encuentra en una fase posterior a la de creación de preguntas.
 1. El sistema muestra por pantalla un mensaje informativo para el usuario, que no podrá participar en la competición si no ha participado en las fases previas.
 - 1d. El alumno pulsa el botón de *Volver a competiciones* y regresa al listado de las mismas.

5.2.12. Descripción de caso de uso: Duelo

5.2.12.1. Caso de uso: Crear duelo

- **Descripción:** El alumno crea un duelo.

- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema.
- **Postcondiciones:** El alumno crea un duelo contra un rival elegido, del tema e idioma seleccionados.
- **Escenario principal:**
 1. El sistema muestra la pantalla inicial de *Repaso*, con un pequeño formulario.
 2. El alumno indica el nombre de usuario del rival a retar, un idioma y un tema, y pulsa el botón *Crear*.
 3. El sistema crea una partida, y redirige al usuario a la pantalla del duelo.
- **Extensiones - Flujo alternativo:**
 - 3a. El sistema comprueba que el usuario seleccionado no existe, o es un profesor, y muestra un error por pantalla.
 - 3b. El sistema comprueba que no existen suficientes preguntas para la combinación de tema e idioma seleccionada, y muestra un error por pantalla.
 - 1-2a. El alumno pulsa el botón de *Volver a duelos* y cancela la operación.

5.2.12.2. Caso de uso: Decidir duelo

- **Descripción:** El alumno decide si aceptar un duelo o no.
- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema, y ha sido retado por otro alumno a un duelo.
- **Postcondiciones:** El alumno cambia el estado del duelo a *Aceptado* o *Rechazado*, según acepte o no el duelo.
- **Escenario principal:**
 1. El sistema muestra el menú de *Duelo* con el listado de duelos, en el que aparece el duelo que debe aceptar.
 2. El alumno pulsa el botón de *Aceptar duelo*, y es redirigido a la pantalla del duelo aceptado.
- **Extensiones - Flujo alternativo:**
 - 2a. El usuario pulsa el botón *Rechazar duelo* y el duelo deja de aparecer en el listado.

5.2.12.3. Caso de uso: Jugar duelo

- **Descripción:** El alumno juega un cuestionario en el modo de juego *Duelo*.

- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema, y existe un duelo creado relacionado con el alumno. El alumno debe ser el retador y el duelo debe estar en fase de creación, o ser el retado y estar el duelo aceptado.
- **Postcondiciones:** El alumno juega una partida de *Duelo* con preguntas de un tema e idioma establecidos cuando se creó el duelo.
- **Escenario principal:**
 1. El sistema muestra la pantalla inicial del duelo con información acerca de éste.
 2. El alumno pulsa el botón de *Jugar*.
 3. El sistema selecciona las preguntas (aleatoriamente en caso del retador, seleccionando las mismas que las del retador en caso del retado), crea los registros de las preguntas para la partida del alumno y muestra por pantalla la primera pregunta del cuestionario junto con una cuenta atrás.
 4. El alumno selecciona una opción y pulsa el botón *Siguiente*.
 5. El sistema guarda la respuesta marcada y muestra la siguiente pregunta, volviendo al paso 4 si la siguiente pregunta no es la última del cuestionario.
 6. El alumno selecciona una opción para la última pregunta y pulsa el botón *Finalizar*.
 7. El sistema guarda las respuestas marcadas y los tiempos permanentemente, y redirige al usuario a la pantalla principal del duelo, donde podrá ver su puntuación.
- **Extensiones - Flujo alternativo:**
 - 2a. El alumno pulsa el botón *Volver a duelos*, cancelándose el comienzo de la partida.
 - 7-8a. El alumno abandona la partida (cierra pestaña o ventana, decide volver atrás o recarga la página).
 1. El sistema muestra un diálogo de confirmación por pantalla.
 2. a) El alumno confirma la acción pulsando en *Abandonar la página*, y el sistema no registra las respuestas (el alumno obtiene 0 respuestas correctas).
 - b) El alumno pulsa *Permanecer en la página*, y se cancela la acción de salir.
 - 7-8b. El alumno no consigue responder las preguntas en el tiempo establecido, y el sistema registra las respuestas que tenga hasta ese momento y redirige al usuario a la pantalla principal del duelo.

5.2.12.4. Caso de uso: Ver informe del duelo

- **Descripción:** El alumno ve los resultados del duelo.
- **Actores:** Alumno.
- **Precondiciones:** El alumno ha iniciado sesión en el sistema, y existe un duelo finalizado que el alumno ha jugado.
- **Postcondiciones:** El alumno ve los datos relacionados con el duelo.
- **Escenario principal:**
 1. El sistema muestra la pantalla inicial del duelo con información acerca de éste.
 2. El alumno ve los detalles del duelo.
- **Extensiones - Flujo alternativo:**
 - 2a. El alumno pulsa en la pestaña de uno de los dos jugadores.
 1. El sistema muestra un resumen con las preguntas del cuestionario, así como el detalle de la primera pregunta del mismo.
 2. a) El alumno pulsa sobre alguno de los botones con el número de la pregunta, y ve el detalle de la respuesta.
 - 2b. El alumno, estando marcada la pestaña de su resumen de partida y una pregunta cualquiera, reporta la pregunta.
 1. El alumno pulsa el botón de *Reportar* (*Extends*: [5.2.9](#) Reportar pregunta).
 - 2c. El alumno abandona la pantalla, terminando el caso de uso.

6. Diseño del Sistema

6.1. Arquitectura del Sistema

Como hemos visto en las secciones anteriores, necesitamos crear un sistema que sea capaz de centralizar toda la información de la plataforma, y que permita interactuar con ella de forma sencilla, con fluidez y eficiencia. Además, queremos que nuestro sistema sea escalable. Por este motivo, se van a realizar dos subproyectos: Una *API RESTful*, que contenga toda la información de la plataforma, y el proyecto web, que posteriormente transformaremos en una *Progressive Web Application (PWA)*. En un primer momento, se pensó en desarrollar una aplicación Android, pero finalmente se descartó, ya que con una PWA podremos reducir los costes del desarrollo del proyecto y el tiempo empleado en la fase de desarrollo. Además, permitirá a los usuarios utilizar la aplicación desde los dispositivos indicados en los requisitos, y los usuarios no necesitarán actualizar la aplicación con cada nueva versión de la plataforma. No obstante, el rendimiento de una PWA, pese a no ser malo (mejora considerablemente al de una Web App convencional), será siempre peor que el de una aplicación nativa. En cualquier caso, si en un futuro se plantease el desarrollo de una aplicación Android o iOS complementaria a la PWA, bastaría con realizar las peticiones oportunas a nuestra API desde esa nueva aplicación.

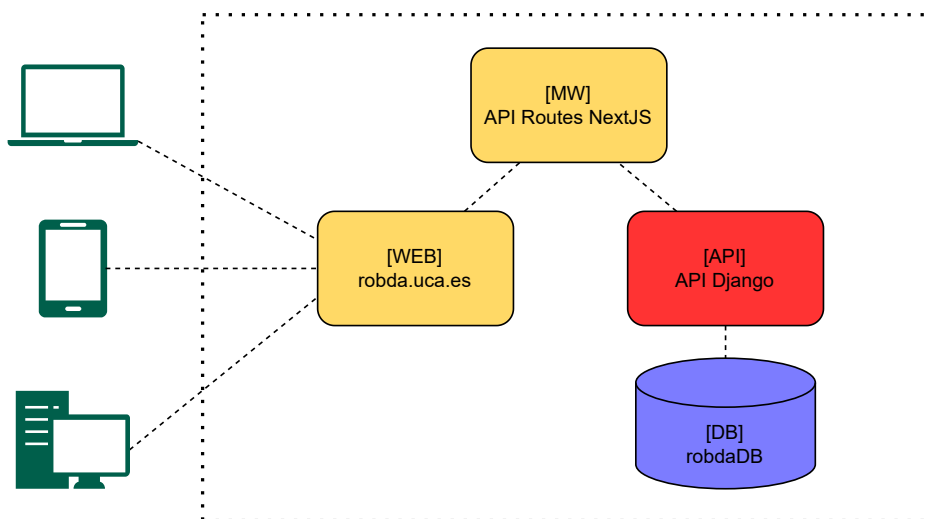


Figura 6.1: Arquitectura de la plataforma.

Una vez tenemos la idea de la estructura de nuestro proyecto, es necesario elegir unas tecnologías adecuadas.

En el caso de API se ha elegido Django Rest Framework, una biblioteca que permite, junto con el framework de Python Django, crear APIs REST de forma muy sencilla. Se seleccionó como lenguaje Python porque ya se había trabajado bastante con él anteriormente, por lo que no iba a suponer ninguna impedimento en el desarrollo. Lo

único necesario era aprender el funcionamiento del framework. Por su parte, Django Rest Framework es una de las bibliotecas más usadas para crear este tipo de APIs.

Para la web, se optó por elegir una tecnología moderna y cuyos conceptos básicos fuesen fáciles de aprender. Al ser esta la primera experiencia real del autor trabajando en el diseño de una web, se intentó facilitar el trabajo para la adaptabilidad y vistosidad de la web. En este sentido, se eligió NextJS, un framework muy popular de React, que nos facilitará la tarea de crear la web. Además, NextJS nos permite convertir la web desarrollada en una PWA en pocos minutos.

Para la comunicación entre la web y la API, es necesario algún tipo de middleware con el que tratar las peticiones. Esa función la ejercerán las API Routes del mismo proyecto de NextJS. En la sección 7.2 lo explicaremos más en detalle.

La API de Django estará conectada directamente a una base de datos MySQL. La decisión de tener una base de datos relacional en vez de una no relacional se debe a la necesidad de ofrecer los datos almacenados de diversas maneras, en las que haya que relacionar datos de entidades distintas con frecuencia. Podemos ver el esquema general de la arquitectura en la figura 6.1

6.2. Diseño Físico de Datos

Partiendo de los requisitos establecidos en la sección 5.1, diseñamos una base de datos que estructure de forma correcta la información necesaria en la plataforma. Esa base de datos se presenta en la figura 6.2, donde aparecen todos los campos que componen las tablas y sus respectivos tipos. Como podemos observar, todas las tablas contienen un campo *id*, que representa la clave primaria subrogada de la tabla.

Cabe destacar que la base de datos real que tendremos en nuestra plataforma contendrá una serie de tablas extra, relacionadas con el framework usado (en este caso, Django), pero no han sido incluidas en la figura por simplicidad. Además, debido a las características de este framework, nos permitirá crear en los modelos una serie de propiedades o datos derivados de los ya almacenados, que nos serán útiles para obtener más información de esta base de datos (e.g. las puntuaciones de los duelos a partir de las partidas de ambos jugadores).

A continuación describiremos las tablas que aparecen en la figura anteriormente mencionada y sus campos:

6.2.1. Tabla User

Esta tabla registra los datos de los usuarios de la plataforma.

- *id*: Clave primaria subrogada.
- *email*: Email del usuario.
- *username*: Nombre de usuario.

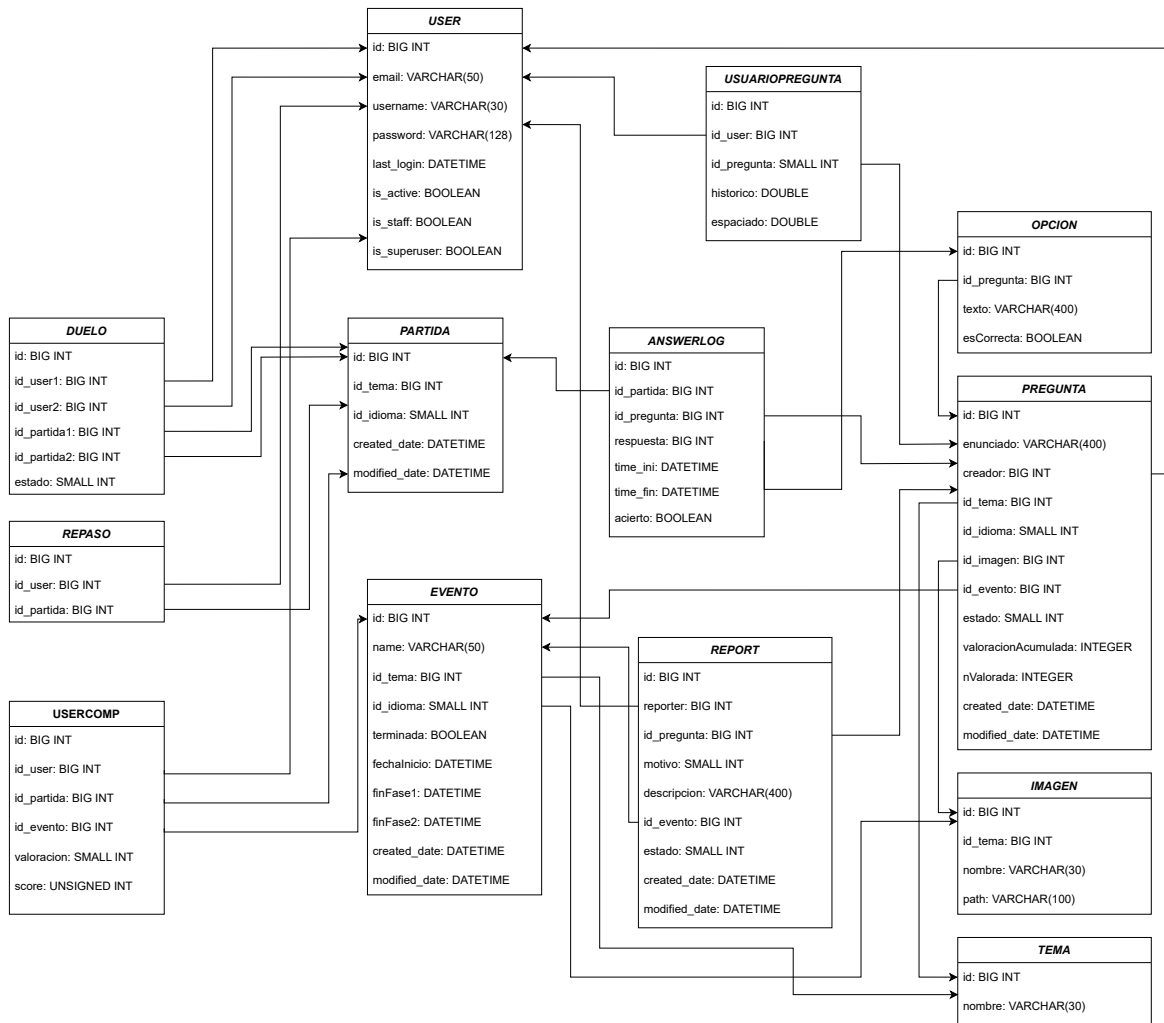


Figura 6.2: Esquema de la base de datos de *ROBDA*

- *password*: Contraseña cifrada.
- *last_login*: Último inicio de sesión (gestionado automáticamente por Django).
- *is_active*: Marcador para saber si una cuenta está activa (verdadero) o si se ha eliminado (falso).
- *is_staff*: Marcador para saber si el usuario es profesor (verdadero) o alumno (falso).
- *is_superuser*: Marcador para saber si el usuario es el administrador de la plataforma (verdadero) u otro usuario (falso).

6.2.2. Tabla Tema

Esta tabla guarda la información de los temas disponibles en la plataforma.

- *id*: Clave primaria subrogada.
- *nombre*: Nombre del tema.

6.2.3. Tabla Imagen

Esta tabla guarda la información de las imágenes disponibles en la plataforma.

- *id*: Clave primaria subrogada.
- *id_tema*: Clave foránea que apunta al tema donde puede usarse la imagen.
- *nombre*: Nombre de la imagen.
- *path*: Dirección a la imagen.

6.2.4. Tabla Pregunta

Esta tabla guarda la información relacionada con las preguntas de la plataforma.

- *id*: Clave primaria subrogada.
- *enunciado*: Enunciado de la pregunta.
- *creador*: Clave foránea que apunta al usuario que creó la pregunta.
- *id_tema*: Clave foránea que apunta al tema relacionado con la pregunta.
- *id_idioma*: Identificador el idioma.
- *id_imagen*: Clave foránea que apunta a la imagen que debe mostrar la pregunta. Puede ser nula en caso de no haber seleccionado una imagen al crear la pregunta.
- *id_evento*: Clave foránea que apunta al evento donde fue creada la pregunta. Tendrá valor nulo si el creador es un profesor.
- *estado*: Identificador del estado de la pregunta.
- *valoracionAcumulada*: Suma total de las valoraciones de los usuarios a la pregunta. Por defecto su valor inicial es cero.
- *nValorada*: Número de veces que la pregunta ha sido valorada. Por defecto su valor inicial es cero.
- *created_date*: Fecha y hora de creación de la pregunta.
- *modified_date*: Fecha y hora de la última modificación realizada a la pregunta.

6.2.5. Tabla Opción

Esta tabla guarda la información relacionada con las opciones de cada pregunta. Se ha separado en una tabla distinta por si en un futuro se permiten preguntas con un número de opciones variable, o con varias opciones correctas.

- *id*: Clave primaria subrogada.
- *id_pregunta*: Clave foránea que apunta a la pregunta en la que la opción aparece.
- *texto*: Texto de la opción.

- *esCorrecta*: Valor que indica si la opción es la respuesta a la pregunta (verdadero) o es incorrecta (falso).

6.2.6. Tabla Evento

Esta tabla contiene la información de las competiciones creadas por los profesores.

- *id*: Clave primaria subrogada.
- *name*: Nombre de la competición.
- *id_tema*: Clave foránea que apunta al tema relacionado con la pregunta.
- *id_idioma*: Identificador el idioma.
- *terminada*: Valor que indica si la competición ha finalizado (verdadero) o si sigue activa (falso).
- *fechaInicio*: Fecha y hora del comienzo de la competición.
- *fnFase1*: Fecha y hora del fin de la fase de creación de preguntas.
- *fnFase2*: Fecha y hora del fin de la fase de cuestionario.
- *created_date*: Fecha y hora de creación de la pregunta.
- *modified_date*: Fecha y hora de la última modificación realizada a la pregunta.

6.2.7. Tabla Partida

Aquí registramos todas las partidas que se crean en la plataforma.

- *id*: Clave primaria subrogada.
- *id_tema*: Clave foránea que apunta al tema relacionado con la pregunta.
- *id_idioma*: Identificador el idioma.
- *created_date*: Fecha y hora de creación de la pregunta.
- *modified_date*: Fecha y hora de la última modificación realizada a la pregunta.

6.2.8. Tabla AnswerLogs

Guardamos en esta tabla los registros de cada pregunta de cada partida.

- *id*: Clave primaria subrogada.
- *id_partida*: Clave foránea que apunta a la partida a la que pertenece el registro.
- *id_pregunta*: Identificador de la partida relacionada con ese registro.

- *respuesta*: Clave foránea que apunta a la opción que ha seleccionado el usuario en al responder la pregunta. Como el registro se crea al pedir la pregunta al servidor, inicialmente su valor es nulo.
- *timeIni*: Fecha y hora del momento en el que el usuario ve la pregunta. En los modos de juego Competición y Repaso, puede darse el caso en que un registro se genere pero no le haya dado tiempo a llegar a leerla. Por tanto, inicialmente el valor de este campo será nulo.
- *timeIni*: Fecha y hora del momento en el que el usuario ve la pregunta. En todos los modos de juego el usuario puede haber leído una pregunta y no haberla contestado. Inicialmente el valor tendrá valor nulo, y si sucede la situación anterior se asignará valor a *timeIni*, mientras que este seguirá siendo nulo.
- *acierto*: Valor que determina si el usuario acertó la pregunta en el momento de contestarla. Necesario en caso de modificación de las opciones de la pregunta.

6.2.9. Tabla Repaso

En esta partida, guardamos los campos adicionales de las partidas jugadas al modo de juego Repaso.

- *id*: Clave primaria subrogada.
- *id_user*: Clave foránea que apunta al usuario que jugó la partida.
- *id_partida*: Clave foránea que apunta al registro con el resto de información de la partida.

6.2.10. Tabla UserComp

Esta tabla registra los datos de las participaciones en competiciones de los alumnos.

- *id*: Clave primaria subrogada.
- *id_user*: Clave foránea que apunta al usuario participa en la competición.
- *id_partida*: Clave foránea que apunta a la partida del usuario. Como el registro de la participación se crea en la fase de creación de preguntas, el valor inicial de este campo será nulo.
- *id_evento*: Clave foránea que apunta al evento en el que se realiza la participación.
- *valoracion*: Valoración del usuario en escala 1-5 que puede ofrecer el usuario en la última fase. Inicialmente su valor es nulo.
- *score*: Puntuación conseguida por el usuario en el momento de finalizar la competición. Por tanto, antes de finalizar el valor de este campo también será nulo.

6.2.11. Tabla Duelo

Esta tabla contiene los datos de los duelos jugados.

- *id*: Clave primaria subrogada.
- *id_user1*: Clave foránea que apunta al usuario que crea el duelo.
- *id_user2*: Clave foránea que apunta al usuario que recibe el duelo.
- *id_partida1*: Clave foránea que apunta a la partida del retador. Como el duelo se crea antes de jugar la primera partida, el valor inicial es nulo.
- *id_partida2*: Clave foránea que apunta a la partida del usuario retado. Hasta que este usuario no comience su partida, no se establecerá un valor en este campo, siendo inicialmente nulo.
- *estado*: Identificador del estado del duelo.

6.2.12. Tabla Report

Aquí se guardan los registros de los reportes creados por alumnos.

- *id*: Clave primaria subrogada.
- *reporter*: Clave foránea que apunta al usuario que crea el reporte.
- *id_pregunta*: Clave foránea que apunta a la pregunta reportada.
- *motivo*: Identificador del motivo de reporte.
- *descripcion*: Información adicional que el alumno proporciona para justificar el reporte.
- *id_evento*: Clave foránea que apunta al evento donde se ha reportado. Si el reporte no ha sido generado en el modo Competición, su valor será nulo.
- *estado*: Identificador del estado del reporte.
- *created_date*: Fecha y hora de creación de la pregunta.
- *modified_date*: Fecha y hora de la última modificación realizada a la pregunta.

6.2.13. UsuarioPregunta

Tabla que contiene información generada por el sistema para personalizar el orden de las preguntas en el modo Repaso.

- *id*: Clave primaria subrogada.
- *id_usuario*: Clave foránea que apunta al usuario.
- *id_pregunta*: Clave foránea que apunta a la pregunta.

- *historico*: Valor en el rango $(0, 1)$ que indica cómo de mal se le da la pregunta al usuario. Por defecto cuando se crea el registro tiene valor 0.5.
- *espaciado*: Valor en el rango $(0, 1]$ que representa la última aparición de la pregunta al usuario de forma decimal. Por defecto tiene valor inicial 1. Para más información sobre estos parámetros, véase [7.3.1](#).

6.2.14. Aclaraciones sobre los campos de las tablas

Como se puede observar, existen diversos atributos que están relacionados con el estado de un modelo. Dicho de otra manera, algunas de las tablas contienen campos numéricos que representan un estado o valor concreto (como un enumerado). Estos atributos son los siguientes:

- **Preguntas - Estado:**

1. En evento (la pregunta ha sido creada en una competición, y no ha finalizado aún).
2. Activa (la pregunta ha sido creada por profesores, o ha concluido la competición en la que se creó. La pregunta no tiene reportes pendientes).
3. Eliminada (la pregunta está eliminada y no es posible verla desde la plataforma, aunque todavía permanece en la base de datos por si posteriormente se decide analizar este tipo de preguntas).
4. Reportada (la pregunta tiene reportes pendientes. Por motivos de consistencia, las preguntas reportadas que aparezcan a los profesores serán únicamente las de competiciones pendientes de corrección del profesor, y las reportadas fuera de competición).

- **Duelos - Estado:**

1. En creación (el duelo ha sido registrado en el sistema, pero el retador no ha jugado su partida aún).
2. Pendiente (el retador ha terminado su partida y el duelo está a la espera de la respuesta del segundo jugador).
3. Finalizado (los dos jugadores han terminado sendas partidas).
4. Rechazado (el segundo jugador ha rechazado jugar el duelo).
5. Aceptado (el segundo jugador ha aceptado el duelo, pero no ha jugado su partida aún).

- **Reportes - Estado:**

1. Pendiente (el reporte está activo y el profesor no ha tomado una decisión).
2. Validado (el reporte ha sido aceptado por el profesor).
3. Invalidado (el reporte ha sido rechazado por el profesor).

- **Reportes - Motivo:**

1. Incorrecto (la pregunta tiene algún fallo).
2. Contenido ofensivo (la pregunta no cumple las reglas básicas de conducta).
3. Otro (cualquier otro motivo, que estará especificado en la descripción).

Las equivalencias entre el estado y su ID estarán establecidas a nivel de aplicación (ya que el framework usado se presta a hacerlo de esta manera). Asimismo, si nos fijamos en la figura anteriormente mencionada, tenemos en varias tablas el atributo `id_idioma`, pero no una tabla `IDIOMA`. Aquí ocurre lo mismo que antes, aunque en este caso `Idioma` sí es un modelo independiente. En nuestra plataforma, aunque podrían añadirse más idiomas sin que suponga mucho esfuerzo, los idiomas registrados actualmente son:

1. Español
2. Inglés

6.3. Diseño de la Interfaz de Usuario

Durante las primeras semanas, se realizó un boceto de la interfaz de la plataforma. Las figuras 6.3 y 6.4 representan los bocetos iniciales para la plataforma. El primero de ellos representa el menú principal del profesor desde una pantalla de ordenador, mientras que la segunda representa el menú principal del alumno en la versión de móvil. En ambos casos, se usan colores claros, y componentes vistosos. La tipografía debe ser legible, y los iconos que se utilicen en la plataforma deben sugerir de forma clara las acciones o elementos que representan. En caso de acceder con un dispositivo con pantalla pequeña, la página debe adaptarse, de forma que el usuario pueda ver una barra lateral para navegar de forma sencilla por la plataforma.

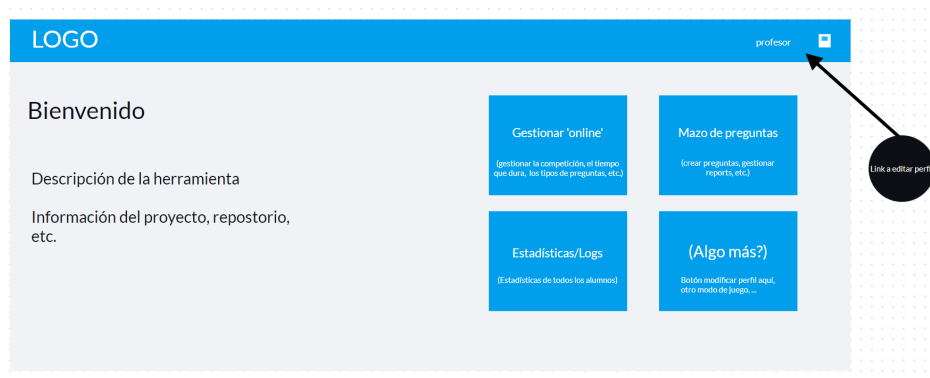


Figura 6.3: Boceto de la plataforma. Versión del profesor para web.

Como vemos en el primer prototipo mostrado en las figuras 6.6 y 6.5, y como podremos ver en el resultado final, esta idea original del diseño de la interfaz se mantiene, si bien es cierto que se modificarán varias cosas. Entre lo más destacado, cambian los colores de la plataforma, de forma que la barra lateral será de un color azul oscuro, la barra superior será de color blanco - transparente, y se añadirá un color morado para los botones (color más visible otras pantallas de la plataforma, véase el anexo B).

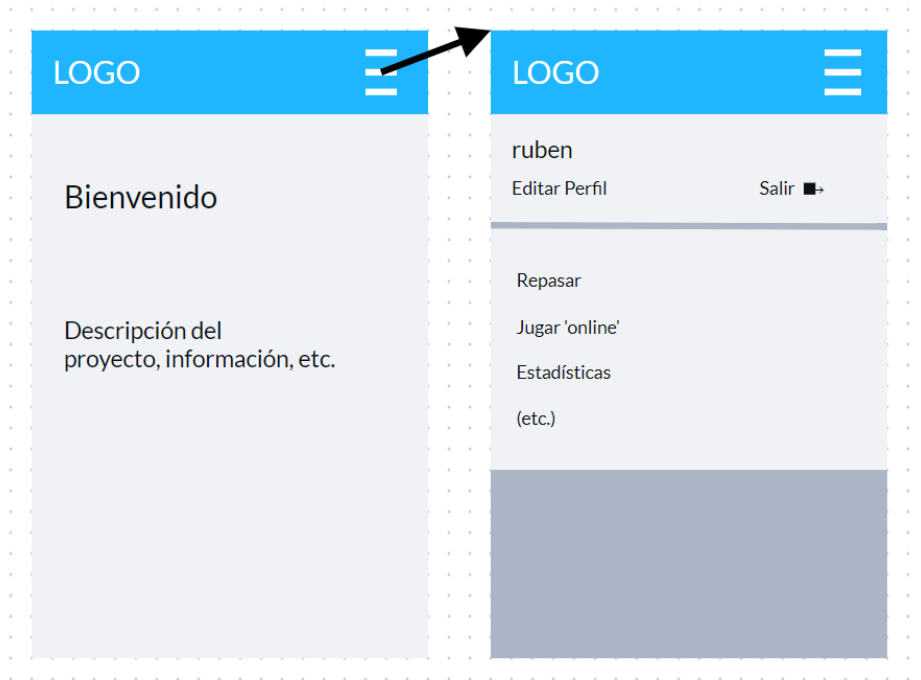


Figura 6.4: Boceto de la plataforma. Versión del alumno para el móvil.

Estos cambios se deben principalmente al uso de una plantilla *open source* llamada Material Kit React [5] para facilitar el desarrollo de la interfaz de la plataforma. Más específicamente, se tuvo problemas para mantener los datos del usuario en la web, ya que al refrescar la página se borraban y había que pedirlos de nuevo a la API. Esta plantilla nos permitió avanzar en el proyecto sin tener que configurar desde cero el contexto de autenticación, además de proporcionarnos componentes ya prefabricados que tomar como base para los nuestros.

La plantilla usa componentes de la biblioteca MaterialUI¹. Los iconos son de Heroicons² y de Material Icons (propios de la biblioteca MaterialUI), mientras que las tipografías más usadas son Plus Jakarta Sans (principalmente para títulos y encabezados) y Roboto Mono (para párrafos y demás texto).

¹MaterialUI - <https://mui.com/>

²Heroicons - <https://heroicons.com/>

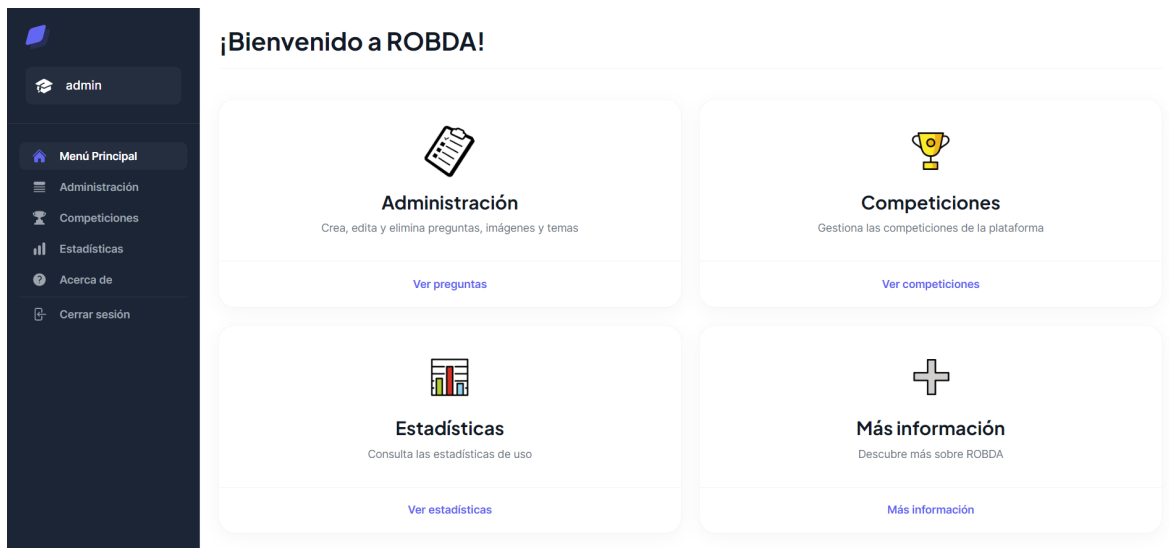


Figura 6.5: Primera versión de la plataforma. Versión del profesor para web.

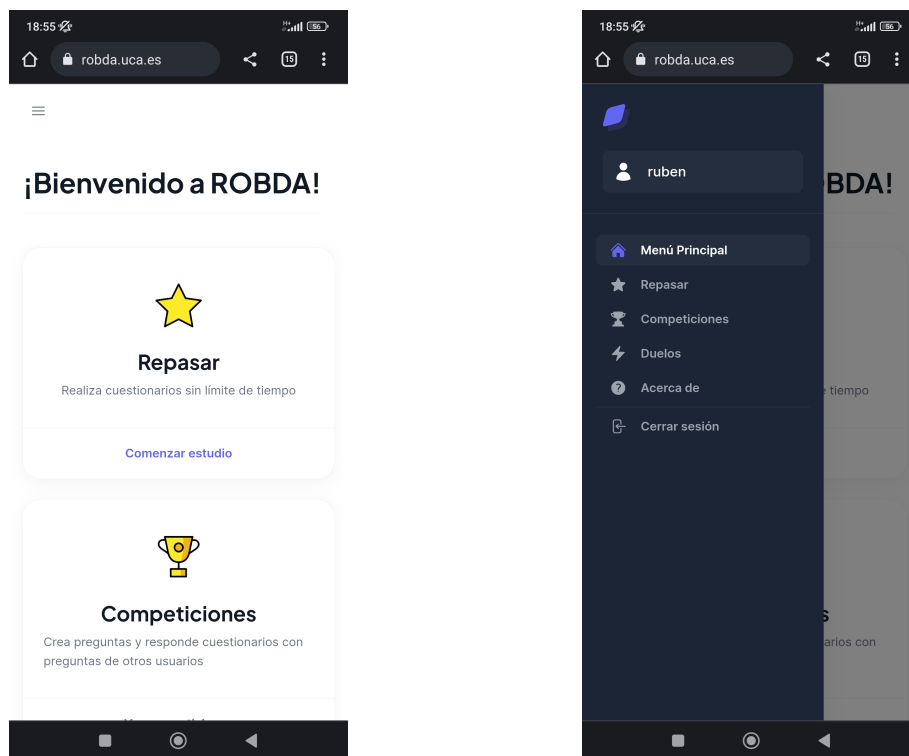


Figura 6.6: Primera versión de la plataforma. Acceso de alumno desde el navegador del móvil.

7. Construcción del Sistema

7.1. Entorno de Construcción

Para la realización de este proyecto, se han utilizado varios lenguajes de programación y herramientas:

7.1.1. Python

Python¹ es un lenguaje de programación interpretado y de alto nivel, conocido por su simplicidad, legibilidad y fácil aprendizaje. Es muy utilizado en el desarrollo de aplicaciones web, científicas, de inteligencia artificial y en la creación de scripts. Tiene una gran cantidad de bibliotecas y frameworks disponibles que permiten acelerar el desarrollo de aplicaciones web y otras aplicaciones. En nuestro proyecto, hemos implementado nuestra API usando **Django**² y **Django Rest Framework**³ debido a su gran popularidad para el desarrollo de Web APIs. Además, el hecho de utilizar Python nos permitirá utilizar en nuestro código bibliotecas como **NumPy**⁴, que en nuestro proyecto hemos utilizado para seleccionar las preguntas más idóneas para cuando el usuario esté jugando (véase el apartado 7.3.1).



Figura 7.1: Logos de Python, Django y Django Rest Framework

7.1.2. JavaScript

JavaScript es un lenguaje de programación interpretado y de alto nivel, ampliamente utilizado en el desarrollo web para crear interactividad en la página y proporcionar una experiencia de usuario mejorada. Es el lenguaje elegido para crear la web desde la que accederá el cliente. Para hacer nuestra página web dinámica, y fácil de crear, hemos utilizado **React**⁵ para la construcción de interfaces de usuario interactivas usando componentes reutilizables, y **NextJS**⁶, un framework de React que proporciona herramientas para el manejo de rutas, renderizado del lado del servidor, pre-renderizado estático y dinámico, y otros elementos importantes para la creación de

¹Python - <https://www.python.org/>

²Django - <https://www.djangoproject.com/>

³Django Rest Framework - <https://www.django-rest-framework.org/>

⁴NumPy - <https://numpy.org/>

⁵React - <https://es.reactjs.org/>

⁶NextJS - <https://nextjs.org/>

aplicaciones web modernas. Además, el proceso para transformar a PWA una web en la que se ha utilizado React y NextJS es más sencillo que con otras alternativas.

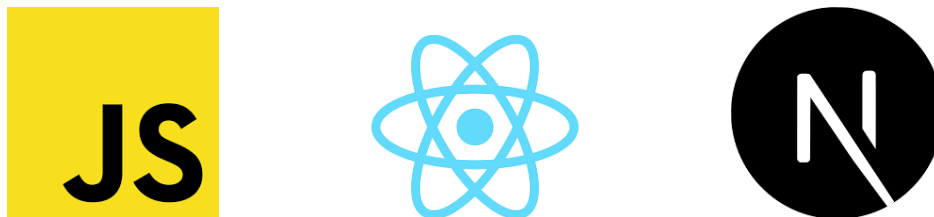


Figura 7.2: Logos de JavaScript, React y NextJS

7.1.3. MySQL

Para almacenar los datos en el servidor, se ha utilizado **MySQL**⁷, un sistema de gestión de bases de datos relacional de código abierto. Es uno de los sistemas de gestión de bases de datos más populares y ampliamente utilizados. Se ha elegido MySQL por su potencia y sencillez, además de haberse usado también en proyectos anteriores.



Figura 7.3: Logo de MySQL

7.1.4. Postman

Postman⁸ es una herramienta con la que podemos probar nuestra API de forma rápida y sencilla mediante una interfaz gráfica. Se ha utilizado en este proyecto para probar los posibles resultados que pueda devolver la API en función de la entrada y del rol del usuario, entre otros factores.



Figura 7.4: Logo de Postman

⁷MySQL - <https://www.mysql.com/>

⁸Postman - <https://www.postman.com/>

7.1.5. Git y GitHub

Todo el código del proyecto se ha alojado en un repositorio de **GitHub** [8], por lo que se ha usado **Git** como sistema de control de versiones.



Figura 7.5: Logo de GitHub

7.1.6. L^AT_EX

Para la realización de esta memoria, se ha usado **L^AT_EX** debido a su gran utilidad para crear documentos científicos y técnicos, permitiendo crear figuras, tablas y referencias de forma fácil y rápida.



Figura 7.6: Logo de L^AT_EX

7.1.7. Draw.io Diagrams

Draw.io⁹ es una herramienta que permite crear diversos tipos de diagramas. Para crear todos los esquemas que aparecen en esta memoria, se ha usado la aplicación de escritorio de esta plataforma.



Figura 7.7: Logo de Draw.io

⁹Draw.io - <https://app.diagrams.net/>

7.1.8. Apache HTTP Server

Apache HTTP Server¹⁰ es un servidor web de código abierto ampliamente utilizado en la industria. Fue creado por la Apache Software Foundation y es una de las soluciones más populares para el alojamiento de sitios web en Internet. Su propósito principal de Apache Server es servir páginas web. En el servidor en el que hemos desplegado la plataforma se manejan las solicitudes con Apache, que las redirige a la API o a la web según corresponda.



Figura 7.8: Logo de Apache HTTP Server

7.2. Código Fuente

Como venimos diciendo anteriormente, el proyecto está compuesto por dos sub-proyectos distintos: una API y una web (que incluye a su vez la PWA). En el directorio raíz del repositorio del proyecto [8], se pueden ver dos carpetas distintas: una carpeta llamada `/api` y otra llamada `/web`, que albergan el código de la API y el código de la web respectivamente. También encontramos el archivo con la licencia del proyecto, así como el `README.md` con una descripción breve sobre la plataforma.

7.2.1. API

Dentro del directorio `/api`, podemos encontrar varios subdirectorios y un archivo, los cuales se comentan a continuación:

- `manage.py`: Es el archivo principal de Django para la gestión de la API. Se utiliza para realizar tareas como la migración de la base de datos, la creación de usuarios de administrador, y otras tareas de administración de la API.
- `/api`: Contiene la configuración principal de la API de Django, incluyendo las URLs principales de la API, la configuración de la base de datos, los archivos de configuración de Django y otros archivos esenciales para la ejecución de la API.

¹⁰Apache - <https://httpd.apache.org/>

- `/apps` : Contiene las aplicaciones de Django que se utilizan en la API. En Django, una aplicación está formada por un conjunto de modelos, serializadores (*serializers*) y vistas (*viewsets*) necesarios para obtener la información solicitada, o para insertar, actualizar o eliminar información.
 - Los modelos son una representación de la estructura de datos que se va a almacenar en la base de datos. Gracias a estos modelos, podemos manejar los datos de la base de datos de forma sencilla, e incluso declarar propiedades (datos que no se guardan en la base de datos pero que se pueden obtener derivados de otros que sí se guardan). Además, podremos modificar el esquema de nuestra base de datos mediante migraciones. Cuando hemos añadido, modificado o eliminado algún campo, tendremos que ejecutar el comando `python manage.py makemigrations` desde la terminal, que generará un archivo con el conjunto de cambios a realizar en la base de datos (se guardan en una carpeta `/migrations` dentro del directorio de la aplicación donde está el modelo declarado). Para aplicar los cambios, bastará con ejecutar el comando `python manage.py migrate`.
 - Los *viewsets* definen el conjunto de operaciones permitidas en un modelo. En esta plataforma las acciones permitidas dependen del modelo, pero en general se permiten operaciones de obtención de información bien en forma de un listado, bien de un elemento concreto (Petición de tipo `GET`), y creación (`POST`), modificación (`PUT`) y eliminación (`DELETE`) de un elemento. Podría decirse que los *viewsets* representan “¿Qué acción puedo realizar?” y “¿Cómo debo realizar la acción?” dentro de cada aplicación de la API.
 - Los serializadores, por su lado, se utilizan para manejar la entrada y salida de datos de la API, permitiéndonos configurar qué datos se envían en la respuesta o se esperan recibir en la petición. También sirven para validar los datos entrantes.

Por simplicidad, hemos tomado la decisión de crear una aplicación llamada `/base`, que contendrá componentes personalizados comunes en toda la plataforma, como son la paginación o los permisos.

El resto de aplicaciones, no obstante, contendrán todos los modelos dentro de cada app en un fichero llamado `models.py`, así como los *viewsets* y serializadores de cada una de las aplicaciones creadas en ficheros cuyo nombre contendrán la palabra *viewsets* o *serializers* respectivamente. Cada aplicación se encarga de una parte concreta de la plataforma:

- `/users` : Gestión de usuarios.
- `/eventos` : Gestión de las competiciones por parte del profesorado.
- `/preguntas` : Gestión de las preguntas, de las imágenes que las pueden acompañar y de los temas.
- `/partidas` : Gestión de las partidas de repaso, de las participaciones en competición de los alumnos y de los duelos.
- `/stats` : Aplicación encargada de obtener las estadísticas de las preguntas

y de las partidas.

7.2.2. Web

La web está contenida en el directorio `/web`. Esta web deriva de una plantilla *open source* que usa Material UI en React [5]. Dentro de este directorio, encontraremos algunos archivos importantes:

- `/public`: Contiene principalmente las imágenes propias de la web, así como las distintas versiones del icono de la plataforma.
- `/src`: Es la carpeta que contiene el código de nuestra plataforma.
- `next.config.js`: Es el archivo de configuración de NextJS. En él hemos añadido las configuraciones necesarias para la creación de la PWA.

Dentro del directorio `/src`, podemos encontrar más carpetas interesantes:

- `/components`: Contiene los componentes comunes a varias secciones de la plataforma. Por lo general, son elementos pequeños que han sido reutilizados en varias de las pantallas de la plataforma.
- `/contexts`: En esta carpeta tenemos creado el contexto de la información del usuario, que nos permite usarlo para saber si un usuario ha iniciado sesión o saber cuál es su rol.
- `/hocs`: Aquí tenemos los componentes de alto nivel, que nos permiten añadir alguna funcionalidad adicional a los componentes. En concreto, disponemos de dos: `with-auth-guard`, para verificar que el usuario ha que realiza la acción ha iniciado sesión, y `with-authorization`, que nos permite verificar el rol del usuario para ver si puede llevar a cabo las acciones que pretende realizar.
- `/layouts`: Contiene el diseño básico de cada uno de los apartados de la plataforma. Según la pantalla en la que nos encontremos, aparecerá uno u otro, que mostrarán u ocultarán la barra lateral, añadirán botones especiales, o tendrán algunas características especiales que justifiquen su existencia.
- `/pages`: Podría considerarse el directorio más importante de este nivel, pues es el que determinará el contenido de las páginas de nuestra plataforma. Las rutas de la misma vendrán dadas por la estructura de esta carpeta. Además, hay que destacar un subdirectorio importante, `/api`, pues contendrá todas las API Routes de la plataforma. La comunicación entre la web y la API de Django se realiza a través de estas API Routes.
- `/sections`: Contiene la mayor parte del contenido de las páginas de pages. Hemos aprovechado para crear componentes específicos de cada sección, para mantener el código de las páginas lo más breve posible. Dentro de cada uno de los archivos de este directorio, podremos encontrar las tablas, filtros, formularios y demás partes que componen nuestra página.

- `/utils`: Si bien es cierto que este directorio venía con la plantilla, hemos incluido un archivo extremadamente importante: `axiosAuth`. Aquí se ha declarado una instancia de axios con la que realizar las peticiones desde el cliente a las API Routes. En la sección 7.3.2 se puede encontrar una explicación más detallada de su relevancia.

7.3. Características destacadas del software

A continuación se van a listar una serie de detalles de la implementación que requieren una mención especial:

7.3.1. Elección personalizada de las preguntas

En el modo de juego *Repaso*, la elección de las preguntas se realiza de forma personalizada para cada usuario. Para ello, disponemos de una tabla en nuestra base de datos llamada `UsuarioPregunta`, la cual contiene los registros asociados a un par de usuario y pregunta concretos, junto con dos valores que representan el histórico y el espaciado de la pregunta. A continuación se definen los dos términos:

- **Histórico**: Un valor en el rango $(0, 1)$ que representa lo bien o lo mal que se le da esa pregunta al alumno. Dicho en otras palabras, podríamos decir que se trata del porcentaje de fallo que el sistema predice para el par de usuario y pregunta. Si un alumno acierta la pregunta en cuestión, el valor decrecerá de forma directamente proporcional al valor actual. Si el alumno falla la pregunta, el valor crecerá de forma inversamente proporcional al valor actual. La fórmula del valor de este factor es la siguiente:

$$hist(u, p)_n = \begin{cases} 0,5 & \text{si } n = 0 \\ hist(u, p)_{n-1} + K * (haFallado - hist(u, p)_{n-1}) & \text{e.o.c} \end{cases}$$

Donde n es el número de veces que le ha salido la pregunta p al usuario u en *Repaso*, $haFallado \in \{0, 1\}$ representa si ha acertado (0) o fallado (1) la pregunta en el último intento, y $K = 0,3$.

- **Espaciado**: Un valor en el rango $[0, 1]$ que representa la prioridad que tiene una pregunta según el tiempo que haya pasado desde la última vez que le salió al alumno. Un valor pequeño representa que la pregunta hace mucho tiempo que no le ha salido al usuario. La siguiente fórmula representa el valor de esta variable para una pregunta p , un usuario u , después de n veces en la que no le haya salido.

$$esp(u, p)_n = S^n$$

Donde $S = 0,99$. Se presupone que en una pregunta nueva (es decir, una pregunta que no aparezca aún en los registros de *UsuarioPregunta* para el usuario que juega

la partida) $n = \infty$, y por tanto $esp(u, p)_n = 0$. Esto hace que las preguntas nuevas tengan prioridad sobre preguntas más recientes, pero se siga manteniendo el orden frente a preguntas que se han preguntado hace mucho.

El valor esta fórmula será prácticamente insignificante cuando el número de preguntas candidatas del repaso sea demasiado pequeño, pues todas las preguntas tendrán un valor de espaciado parecido.

Con estos dos valores, calcularemos los pesos finales de las preguntas. Estos pesos se calculan de la siguiente manera:

$$peso(u, p) = hist(u, p) * (1 - esp(u, p))$$

Tras el cálculo de esos pesos, se calculará la probabilidad que debería tener cada pregunta de salir, y se escoge una de ellas es base a esta probabilidad. Esto se hace gracias a la función `random.choice()` de la biblioteca Numpy. De esta manera, las preguntas con un mayor peso tendrán más posibilidades de salir (preguntas difíciles para el usuario, que no han salido nunca o que han salido hace mucho), mientras que las que tengan un valor pequeño saldrán con muy poca frecuencia (preguntas que han salido recientemente, o que el alumno siempre acierta).

En la fórmula anterior, tanto el histórico como el espaciado están guardados en la tabla anteriormente mencionada. Para tener ese cálculo actualizado siempre, se deben hacer una serie de operaciones:

- Cuando se ha elegido una pregunta que no se le ha preguntado al alumno aún, se debe crear un registro en la tabla `UsuarioPregunta`, con el valor por defecto del histórico (0.5). Independientemente de si estaba o no el registro, se establece el factor de espaciado a 1 (es decir, S^0).
- Antes de enviar la pregunta, se actualiza el valor de espaciado de todas las preguntas candidatas no elegidas multiplicando su valor por S .
- Cuando se recibe la respuesta por parte del usuario, se actualiza el valor del histórico correspondiente a esa pregunta.

También se podrían haber tenido en cuenta otros factores (como una la dificultad de la pregunta para todo el conjunto de alumnos), pero por motivos de eficiencia no se han incluido más. En la sección 9.3 se puede ver una alternativa eficiente para intentar mejorar la fórmula propuesta.

7.3.2. Autenticación en la API desde el cliente

La autenticación del usuario desde el proyecto de NextJS a la API de Django se ha realizado mediante el uso de JSON Web Tokens. En este caso, se han utilizado dos tokens: uno de acceso, con el cual se hacen las peticiones generales a la API, y otro de refresco, que nos permite actualizar el token de acceso. Esto se hace principalmente para aumentar la seguridad de la plataforma.

¿Cómo hemos guardado esos tokens? Los hemos guardado como cookies de tipo `HttpOnly`, con el atributo `Secure` configurado y con el atributo `SameSite=strict`. De esta manera, los protegemos frente a los ataques más comunes.

¿Cómo hemos realizado ese refresco? Gracias a la declaración de un interceptor en una instancia de axios. Esta instancia ha sido la que se ha utilizado para realizar todas las peticiones desde el cliente hasta las API Routes de NextJS, y desde estas rutas se hacían las peticiones a la API. Cuando realizamos una petición con nuestro token de acceso, y antes de realizarla nuestro token había expirado (o simplemente no existía el token), la API devolverá un error de tipo `401 Unauthorized`, que pasará a esa instancia de axios tras haberlo devuelto la API Route. En el interceptor, hemos creado un fragmento de código para que, en caso de producirse un error 401, realice una petición a la API para refrescar el token. Si esa petición funciona, se guardan los nuevos tokens (no solo se actualiza el de acceso, el de refresco también, y el anterior pasa a una lista negra por motivos de seguridad). En caso de producirse de nuevo otro error 401, significará que el token de refresco no es válido (bien por haber expirado, bien por no existir), y redirigiremos al usuario al login para que inicie sesión.

Además, el hecho de tener una instancia de axios nos permite tener preestablecida la URL base a la que vamos a hacer las peticiones y los headers que vamos a utilizar sin necesidad de repetirlos en cada petición.

¿Por qué las peticiones se hacen mediante las API Routes de NextJS y no se hacen directamente desde el cliente? Por añadir una capa más de seguridad a nuestro proyecto. Haciéndolo de esta manera, podemos controlar qué cookies enviamos a la API, y también tratar la petición y la respuesta antes de que le llegue al cliente de una forma sencilla.

7.3.3. Ampliación de imágenes

Las imágenes que aparecen en la plataforma, por lo general, representan tablas o diagramas en los que hay que leer texto. Es de vital importancia que ese texto pueda verse perfectamente desde cualquier dispositivo. En dispositivos con pantallas más pequeñas como teléfonos móviles, puede darse el caso en que la imagen no se vea lo suficientemente grande como para poder ver ese texto con claridad.

Por ese motivo, las imágenes se muestran usando un módulo llamado Yet Another React Lightbox¹¹. Gracias a uno de sus plugins, podemos añadir *zoom* a las imágenes, de forma que se pueda agrandar el tamaño de las mismas para poder ver el texto que aparece en ellas con claridad. El resultado de añadir esta característica lo podemos ver en la figura 7.9, donde se muestra la imagen del cuestionario ampliada. El propio usuario será capaz de configurar el nivel de zoom de la imagen, pudiendo apreciar así incluso los detalles más pequeños.

7.3.4. PWA

Como vimos anteriormente, uno de los requisitos no funcionales que se pretendía satisfacer era la compatibilidad de la plataforma con cualquier dispositivo que disponga de navegador web. Si bien este requisito lo podíamos satisfacer con la propia web,

¹¹<https://yet-another-react-lightbox.com/>

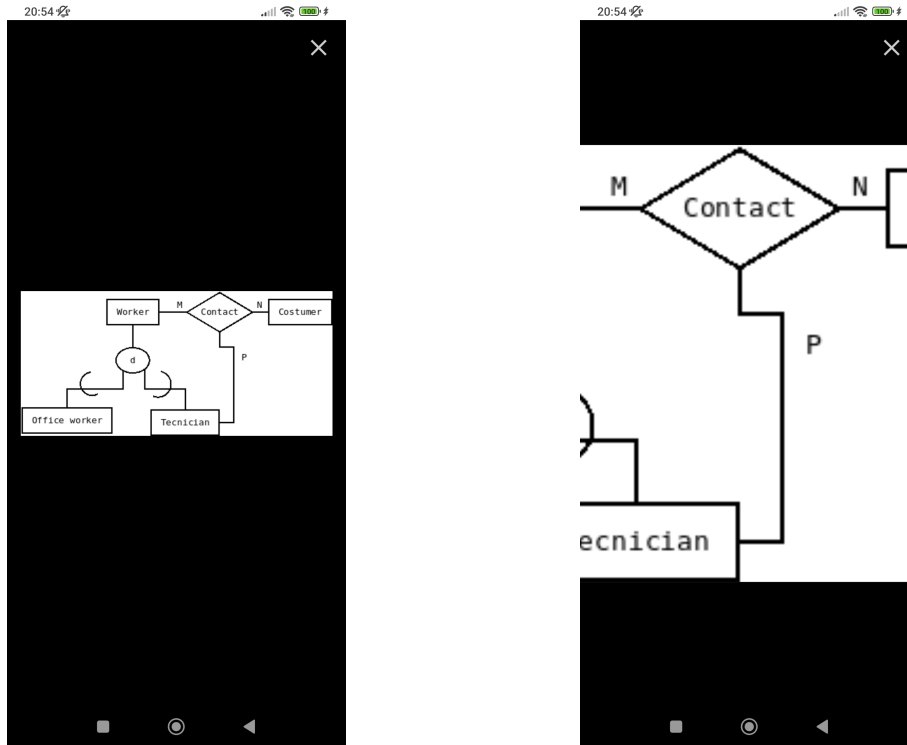


Figura 7.9: Ampliación de imágenes desde el teléfono móvil.

hemos querido dar un paso más y crear una PWA. Podemos instalarla desde el propio navegador de forma rápida y sencilla, y así tendremos acceso a la plataforma como si de una aplicación nativa se tratase. Esta PWA hará las veces de aplicación Android, aplicación iOS y aplicación de escritorio.

La creación de una PWA, a partir de una web en NextJS, es bastante sencilla. Lo único que se necesita es modificar la información del fichero `/public/manifest.json` para incluir información acerca de la plataforma y sobre los iconos, cambiar el archivo `/src/pages/_document.js` para incluir el archivo anterior con una etiqueta `meta`, y modificar el archivo `next.config.js` para añadir la configuración que queramos a nuestra PWA. Para que todo funcione correctamente, se deberá instalar el módulo `next-pwa`. La PWA solo puede funcionar en entorno de producción.

8. Pruebas del Sistema

8.1. Entorno de pruebas

Las pruebas del sistema se han realizado de forma manual cada vez que se desarrollaba alguna característica nueva, y cada vez que se tenía que cambiar alguna antigua. Al estar la plataforma dividida de forma conveniente, los modos de juego de los alumnos y la gestión de los profesores de cada elemento era completamente independientes. Gracias a estas pruebas, se han ido descubriendo errores que se habían cometido, y se han ideado nuevas estrategias para desarrollar una plataforma lo más profesional y robusta posible.

Para el proyecto de Django, se ha usado Postman (véase el apartado 7.1.4). Esta herramienta nos ha permitido crear todas las peticiones HTTP, probándolas así con distintos datos de forma que se comprobase que nuestra API funcionaba cuando la petición era correcta, y mostraba un error cuando los datos de la petición no fuesen los esperados.

Respecto a la web, se han tenido que comprobar que las peticiones de las API Routes funcionasen correctamente, ya que si no funcionan, la web no podría obtener los datos de la plataforma. Además, y respecto al diseño de la web, se ha comprobado que las pantallas se viesan bien desde dispositivos con pantallas de diversos tamaños (mediante la consola de desarrollador de Firefox, la cual permite visualizar cualquier página simulando la pantalla de un tamaño dado). El desarrollo de cada pantalla no concluía hasta que se comprobase que los botones, figuras y tablas de la pantalla se vieran correctamente en móviles, tablets o dispositivos con pantallas grandes.

En esta parte, se han tenido bastantes problemas para corregir algunos errores, pues en ocasiones, cuando la API fallaba, el error no se mostraba al cliente. Debíamos ver el error desde la parte del servidor de la web, comprobar dónde estaba el fallo (si en la petición o en la API en sí), y corregirlo.

Tras terminar el desarrollo de la plataforma al completo, se realizó una prueba de todas las funcionalidades del sistema para comprobar que no nos dejábamos nada atrás. Se comprobó así que la plataforma estaba lista para usarse.

8.2. Errores encontrados en las pruebas

Gracias a las pruebas se han encontrado varios errores, algunos más importantes que otros. A continuación se detallan las funcionalidades a las que más tiempo se ha dedicado, junto con los motivos de los fallos.

- En la pantalla de creación de imágenes, se debe obtener la imagen desde el cliente, subirla a la web, mantenerla para que el profesor vea la previsualización de la

misma, y cuando envíe el formulario pasar la imagen desde la web hasta la API.

En primer lugar, se comprobó que la imagen tal y como la recogía la web no permitía previsualizarla correctamente. Esto se pudo solucionar tras tratar la imagen recibida, y mostrar el archivo en binario. Pese a ello, el principal problema que tuvimos que afrontar aquí era el envío de la imagen desde la web hasta la API, pues se observó que la API no conseguía reconocer el fichero enviado como imagen. Esta funcionalidad no fue problemática a la hora de probarla con Postman, pues funcionaba correctamente. También se podían crear imágenes sin problemas haciendo una petición con curl a la API, así que el problema estaba en el servidor web. Al final, el problema era que la API esperaba recibir un *form-data* con la imagen, y lo que estaba reconociendo era texto plano. Tras cambiar el cuerpo de esa petición a la API por un *form-data*, se consiguió arreglar este fallo. Debemos destacar aquí que NodeJS 18 permite crear un objeto `FormData()` sin necesidad de añadir dependencias externas, pero en versiones anteriores no.

- Cuando se comenzó a desarrollar la web, lo primero que se desarrolló fue lo relacionado con la autenticación. Tras crear la pantalla de registro, la de inicio de sesión y la gestión de los datos del usuario, se observó durante las pruebas del *silent refresh* (la renovación de los tokens JWT) que no funcionaba como era de esperar. Concretamente, si se había iniciado sesión previamente y el usuario refrescaba la página, se debía de hacer una petición a la API para recuperar los datos del usuario. Esta petición se hacía correctamente, pero la web no esperaba a recibirlos para renderizar los componentes. Esto hacía que el usuario viera la página sin sus datos, después un “pantallazo” que le enviaba a la pantalla de inicio de sesión, para inmediatamente después devolver al usuario al punto donde se encontraba.

Se trató de corregir este fallo durante aproximadamente un mes con dos intentos. El primero, usando la librería *Redux*¹ para tratar de guardar el estado del usuario. Tras fracasar, se intentó crear un contexto (similar a la función de Redux, pero nativa de React), aunque el resultado fue el mismo. Al ser la primera experiencia que el desarrollador del trabajo tenía con React y no se quería tanto tiempo en algo que no era relevante en el trabajo, se optó por usar una plantilla de React que tuviese ya implementada esta parte. Dicha plantilla también usa un contexto para almacenar la información del usuario y recuperarla cuando sea necesario, pero cuidaba ese detalle de esperar a tener los datos para renderizar la página. De esta manera, pudimos centrarnos en las funcionalidades verdaderamente relevantes sin necesidad de complicar el trabajo en aspectos más específicos del framework.

8.3. Pruebas de aceptación

Cuando se terminó el desarrollo de la plataforma, se permitió que los profesores de la asignatura de Bases de Datos accediesen a la plataforma para que probasen las distintas funcionalidades de la misma. No se reportó ningún error en el funcionamiento de la plataforma, dando por satisfactorio el resultado de estas pruebas.

¹<https://redux.js.org/>

8.4. Resultados

Más allá de las pruebas anteriormente descritas, se han llevado a cabo tras terminar el proyecto otra serie de pruebas para garantizar que cumplíamos los requisitos no funcionales propuestos en la sección 4.4. Para ello, hemos utilizado `lightest.app`², que nos permite ver el rendimiento de nuestra plataforma y compararlo con otras. Por tanto, vamos a averiguar si cumplimos con el requisito no funcional RNF3 comparando nuestra plataforma con las que todavía están disponibles en internet de las que hemos visto en secciones anteriores y que pertenecen a la Universidad de Cádiz (i.e. ChatbotSQL, VirUCA y SMCP-Training), además de la web educativa por excelencia, Moodle³. Los resultados del test, extraídos de [18], pueden verse en la tabla 8.1.

Si analizamos los resultados, podemos ver que nuestra plataforma obtiene la segunda mejor puntuación entre todas, solo un punto por detrás de la primera. No obstante, debemos destacar que tanto SMCP-Training como VirUCA usan el protocolo HTTP y no HTTPS, lo que les hace ganar en el apartado de rendimiento a costa de perder en el de buenas prácticas. Por otro lado, podemos ver que la página que menor puntuación obtiene de forma inesperada es Moodle, pero al ser la que más contenido tiene es lógico que tenga una mala puntuación en rendimiento. En cualquier caso, el resultado de nuestra plataforma, que es el que a nosotros nos interesa en especial, es bastante bueno, por lo que podríamos afirmar que cumplimos con los requisitos no funcionales mencionados y que además que la elección de tecnologías a usar en el proyecto ha sido adecuada.

Tabla 8.1

Comparativa de las estadísticas de ROBDA frente a webs similares

Plataforma	Rendimiento	Accesibilidad	Buenas prácticas	SEO	Total
ROBDA	76	90	92	92	350
VirUCA	98	80	83	90	351
ChatbotSQL	55	89	92	92	328
SMCP-Training	68	97	83	92	340
Moodle	61	88	92	84	325

Siguiendo con las pruebas de requisitos no funcionales, también hemos puesto a pruebas las condiciones impuestas en los requisitos RNF1 y RNF2. Las pruebas han consistido en probar la web desde diversos dispositivos, que se listan a continuación:

- PC MSI con Windows 10.
- Portátil HP con Windows 11.
- Portátil Toshiba con Ubuntu LTS 20.04.
- Smartphone Xiaomi Mi 11i con Android 13.
- iPad Pro con iPadOS 16.
- iPhone 13 con iOS 16.

²<https://www.lightest.app/>

³<https://moodle.org/>

- MacBook Air con macOS Ventura.

Asimismo, se han probado los navegadores web gráficos que se habían establecido como requisito mínimo:

- Google Chrome 109.
- Mozilla Firefox 107.
- Microsoft Edge 109.
- Safari 16.3.

Con esto hemos garantizado que la web funciona correctamente con estas especificaciones mínimas, sin perjuicio de que también pueda funcionar para versiones anteriores.

Gracias a estas pruebas, pudimos detectar un error que únicamente sucedía en Microsoft Edge y Google Chrome, navegadores que no disponíamos en el entorno de desarrollo. El problema estaba en el listado de imágenes, ya que algunas desbordaban de su contenedor y provocaba un efecto de solapamiento entre imágenes.

Parte III

Epílogo

9. Conclusiones

9.1. Objetivos alcanzados

El objetivo de este proyecto, tal y como hemos comentado en secciones anteriores, era crear una plataforma educativa para la asignatura de Bases de Datos del grado en Ingeniería Informática de la Universidad de Cádiz basada en cuestionarios.

La plataforma creada tenía que ser útil para los profesores, de forma que pudiesen amenizar el aprendizaje de conceptos teóricos y analizar los datos de los alumnos, pero también tenía que ser útil para los alumnos, que gracias a esta plataforma podrán poner a prueba sus conocimientos y repasar de cara a los exámenes de la asignatura.

Hemos creado una plataforma con varios modos de juego, que unidos todos ellos cumplen estos objetivos iniciales:

- El modo de juego de competiciones ha sido la justificación perfecta para incentivar a los alumnos a crear preguntas lo suficientemente buenas como para que supongan un reto a sus compañeros, además de recompensarles por analizar las preguntas de sus compañeros. Este modo de juego será el que a medio/largo plazo nutra de preguntas a nuestra plataforma.

Podría considerarse este modo de juego como una criba de las preguntas erróneas, pues las preguntas creadas de los usuarios aparecerán a los demás compañeros en sus cuestionarios. De esta manera, separamos estas preguntas, recién creadas y que todavía necesitan ser probadas, de los otros modos de juego en los que aparecen preguntas buenas (o al menos, preguntas que nadie ha reportado). El sistema de gestión de reportes implica que el alumno notifica al profesor de una forma sencilla cuando ve una pregunta errónea, de modo que facilita la labor del profesor a la hora de controlar el contenido.

El hecho de que el profesor sea el que controla las competiciones significa además que el profesor será el que determine qué preguntas crea el alumno, cuándo puede crearlas, el tema que traten y el idioma en el que se escriban, de modo que el alumno estará siempre controlado por el profesor. También podrá obtener información sobre la valoración que los alumnos han otorgado a la competición, pudiendo realizarse un posterior estudio de los temas que más gustan entre los alumnos.

- El modo de repaso, por su parte, es un modo principalmente ideado para ayudar al alumno. Lo que se propone con este modo de juego es una forma más de poder repasar los contenidos de la asignatura, con las preguntas que los profesores y otros compañeros han creado previamente. Además, este modo de juego ofrece al alumno una experiencia totalmente personalizada, de modo que mostrará siempre las preguntas que sean más difíciles para el usuario.

Asimismo, este modo de juego ofrece también información relevante al profesor,

ya que, además de las propias partidas, se recopilará información de la dificultad percibida por el alumno para cada pregunta que le aparezca. Esta información podría ser útil en un posterior análisis.

- Los duelos ofrecen una experiencia más directa a los alumnos para poder jugar con amigos. Este modo de juego les permitirá retarse los unos a los otros, sin olvidar que mientras lo hacen están respondiendo preguntas que podrían caerles en un examen.

Añadido a lo anterior, se ha conseguido crear la PWA que permite a los usuarios acceder a la plataforma desde cualquier dispositivo. Para ello, se ha tenido que desarrollar la web de forma que fuese totalmente adaptable y eficiente.

9.2. Lecciones aprendidas

A lo largo de este proyecto, se han tenido que superar una serie de objetivos y dificultades para conseguir un resultado de calidad.

Las tecnologías empleadas en él han sido escogidas única y exclusivamente por las características del proyecto en sí, y no por la experiencia que tenía en ellas. Por ello, a lo largo del proyecto han habido varias fases de aprendizaje, en las que se ha tenido que aprender cómo crear una API, cómo trabajar con componentes web, y cómo trabajar con los frameworks que se han usado.

Pese a que no era el primer proyecto web en el que trabajaba (ya había trabajado en otro proyecto de la asignatura el año anterior), sí ha sido el primero en el que he tenido que comenzar desde cero. Todo lo relacionado con programación que había aprendido hasta el momento había sido de manera autodidacta, pues no había cursado ninguna asignatura de programación web en la carrera, lo que ha provocado que en ocasiones no encontrase la forma de solucionar un problema y tuviese que buscar alguna alternativa viable con los conocimientos que tenía. Esto ha sucedido sobre todo con el proyecto en NextJS, que aunque se han utilizado componentes predefinidos para no tener que bregar con CSS en la medida de lo posible, en ocasiones han quedado claras las carencias que tenía en desarrollo frontend. También se ha tenido que desplegar el proyecto en un servidor de la Universidad de Cádiz sin apenas experiencia previa, pero se pudo hacer relativamente rápido.

Una vez terminado un proyecto de esta envergadura, creo que mis conocimientos de programación web han aumentado. De hecho, he encontrado en el desarrollo de APIs una buena opción para mi carrera profesional, pues he disfrutado mucho trabajando con Django Rest Framework.

9.3. Trabajo futuro

La plataforma, como hemos repetido en varias ocasiones a lo largo de la memoria, está creada para usarse en la asignatura de Bases de Datos del grado en Ingeniería

Informática de la Universidad de Cádiz. Sin embargo, al haber terminado el desarrollo justo después de que los alumnos hayan realizado el examen, no ha sido posible poner a prueba la plataforma con un grupo de alumnos real. Por ese motivo, uno de los principales pasos futuros será probar la plataforma con los alumnos de la asignatura para el curso 2023/24. En función del feedback que puedan proporcionarnos, podremos obtener un mayor o menor número de mejoras futuras, si bien podemos listar a continuación una serie de características que podrían incluirse en próximas versiones.

9.3.1. Mejora de la personalización de las preguntas en el modo Repaso

En la primera versión de la plataforma, hemos creado una fórmula bastante sencilla para la personalización de las preguntas en el modo Repaso (véase 7.3.1). En esta fórmula, solamente se consideran la dificultad de la pregunta para el usuario basada en el histórico de aciertos y fallos, y la última vez que el usuario vio la pregunta. No obstante, en la idea original teníamos pensado añadir un factor que proporcione información sobre la dificultad general de la pregunta, es decir, de la dificultad de la pregunta para todos los usuarios. Se pensó en utilizar el porcentaje de aciertos, pero se descartó debido a que, por cada pregunta que le salga al usuario, la plataforma tendrá que calcular el porcentaje de acierto de cada pregunta candidata accediendo a los registros. En caso de tener muchas preguntas y muchos registros, temíamos que el tiempo que tardase en hacer los cálculos fuese muy grande. Por ello, se propuso como alternativa analizar la dificultad percibida de la pregunta (la valoración que los usuarios hacen tras responder cada pregunta en el modo Repaso), ya que así no se tendría que consultar la tabla de registros. Para asegurarnos de que los cambios en la fórmula sean positivos, se pretende realizar un estudio antes para comprobar si existe correlación entre esa dificultad percibida y el porcentaje de aciertos. También se podría estudiar la media de los valores de histórico agrupados por pregunta. En cualquier caso, la realidad es que para esta primera versión se ha optado por una fórmula más simple y genérica, que posteriormente iremos optimizando conforme sepamos con mayor seguridad qué métricas pueden usarse.

Asimismo, la idea de personalización del cuestionario que inicialmente hemos aplicado en el modo de juego *Repaso*, también podría aplicarse de forma sencilla en la elección de preguntas de *Duelos*, de forma que se combinen los factores de ambos usuarios para generar un cuestionario que suponga un reto para ambos. En el modo de juego *Competición* no tiene sentido, pues la idea de este juego es que los usuarios se enfrenten a preguntas recién creadas por otros compañeros.

9.3.2. Preguntas con código HTML incrustado

Si bien es cierto que este objetivo no es del todo necesario, a la hora de importar las preguntas existentes en el campus virtual de la asignatura a la plataforma se han tenido problemas con las preguntas que contenían código HTML. El problema está en que en muchas de ellas, ese código representaba una tabla incrustada, o el subrayado de las claves de una tabla.

Esto ha provocado que en la versión inicial de la plataforma solamente se cuenten con alrededor de 100 preguntas de ER, y 30 preguntas de SQL. Por ello se baraja como opción permitir código HTML en las preguntas, si bien habría que limpiar las cadenas desde la API para evitar que inyecten código malicioso al crear preguntas. Otra opción posible sería buscar una notación distinta para las claves en vez de subrayarlas (e.g. marcarlas con un asterisco), y transformar las tablas HTML en imágenes para que los usuarios, que presumiblemente no habrán tocado código HTML antes, puedan reutilizar esas mismas tablas sin tener que aprender este lenguaje de marcado.

9.3.3. Estadísticas

Para esta versión de la plataforma se han incluido muchas gráficas y datos relevantes para los profesores, si bien es cierto que toda la información que se pueda extraer de los registros será bien recibida por ellos. Por este mismo motivo, se pretende llevar a cabo un seguimiento de la plataforma para el próximo curso, de forma que, en caso de que los profesores deseen añadir otras gráficas distintas a las que se tienen actualmente, pueda hacerse sin problemas. Además, se pretende que en el futuro estas estadísticas sean exportables en el formato estándar xAPI para que puedan analizar los datos en detalle.

GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.

[<https://fsf.org/>](https://fsf.org/)

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

Preamble

The purpose of this License is to make a manual, textbook, or other functional and useful document “free” in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of “copyleft”, which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The “**Document**”, below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as “**you**”. You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A “**Modified Version**” of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A “**Secondary Section**” is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document’s overall subject (or to related matters) and contains

nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The “**Invariant Sections**” are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The “**Cover Texts**” are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A “**Transparent**” copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not “Transparent” is called “**Opaque**”.

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The “**Title Page**” means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, “Title Page” means the text near the most prominent appearance of the work’s title, preceding the beginning of the body of the text.

The “**publisher**” means any person or entity that distributes copies of the Document to the public.

A section “**Entitled XYZ**” means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as “**Acknowledgements**”, “**Dedications**”, “**Endorsements**”, or

“**History**”).) To “**Preserve the Title**” of such a section when you modify the Document means that it remains a section “Entitled XYZ” according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document’s license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

- K. For any section Entitled “Acknowledgements” or “Dedications”, Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled “Endorsements”. Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled “Endorsements” or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version’s license notice. These titles must be distinct from any other section titles.

You may add a section Entitled “Endorsements”, provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original

author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled “History” in the various original documents, forming one section Entitled “History”; likewise combine any sections Entitled “Acknowledgements”, and any sections Entitled “Dedications”. You must delete all sections Entitled “Endorsements”.

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an “aggregate” if the copyright resulting from the compilation is not used to limit the legal rights of the compilation’s users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document’s Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and

the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled “Acknowledgements”, “Dedications”, or “History”, the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <https://www.gnu.org/licenses/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License “or any later version” applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy’s public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

“Massive Multiauthor Collaboration Site” (or “MMC Site”) means any World Wide Web server that publishes copyrightable works and also provides prominent fa-

cilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A “Massive Multiauthor Collaboration” (or “MMC”) contained in the site means any set of copyrightable works thus published on the MMC site.

“CC-BY-SA” means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

“Incorporate” means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is “eligible for relicensing” if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

Copyright © YEAR YOUR NAME. Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled “GNU Free Documentation License”.

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the “with ... Texts.” line with this:

with the Invariant Sections being LIST THEIR TITLES, with the Front-Cover Texts being LIST, and with the Back-Cover Texts being LIST.

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Bibliografía

- [1] AnkiWeb. Remote Decks: Anki collaboration using google docs, 2021.
- [2] AnkiWeb. Shared Decks, 2023.
- [3] Antonio Balderas and María-Araceli Losey-León. Virtual learning tool for the training of the standard marine communication phrases. In *2018 International Symposium on Computers in Education (SIIE)*, pages 1–5, 2018.
- [4] Anke Berns, Manuel Palomo-Duarte, Juan Manuel Doderó, and Alberto Cejas. Guess it! Using gamified apps to support students foreign language learning by organic community-driven peer-assessment. In *Open Learning and Teaching in Educational Communities: 9th European Conference on Technology Enhanced Learning, EC-TEL 2014, Graz, Austria, September 16-19, 2014, Proceedings 9*, pages 482–485. Springer, 2014.
- [5] Devias. Material Kit React - GitHub, 2023.
- [6] Ana Belén Díaz, Antonio Balderas, and Sokratis Papaspyrou. Gamification adaptable to the content of the subjects through the VirUCA web platform. In *2018 International Symposium on Computers in Education (SIIE)*, pages 1–4. IEEE, 2018.
- [7] Django REST framework. Django REST framework, 2023.
- [8] GitHub. Repositorio de robda, 2023.
- [9] Wikipedia, la enciclopedia libre. API, 2023.
- [10] Wikipedia, la enciclopedia libre. Aplicación web progresiva, 2023.
- [11] Wikipedia, la enciclopedia libre. Django (framework), 2023.
- [12] Wikipedia, la enciclopedia libre. JSON, 2023.
- [13] Wikipedia, la enciclopedia libre. JSON Web Token, 2023.
- [14] Wikipedia, la enciclopedia libre. Next.js, 2023.
- [15] Wikipedia, la enciclopedia libre. Python, 2023.
- [16] Wikipedia, la enciclopedia libre. React, 2023.
- [17] Wikipedia, la enciclopedia libre. Transferencia de estado representacional, 2023.
- [18] lightest.app. Comparativa de rendimiento de webs, 2023.
- [19] Matthew Lu, John H Farhat, and Gary L Beck Dallaghan. Enhanced learning and retention of medical knowledge using the mobile flash card application anki. *Medical Science Educator*, 31(6):1975–1981, 2021.

- [20] Denis Manente and Simon Schaefer. *Moodle Plugin StudentQuiz*. PhD thesis, HSR Hochschule für Technik Rapperswil, 2017.
- [21] Manuel Palomo-Duarte, Anke Berns, Alberto Cejas, Juan Manuel Dodero, Juan Antonio Caballero, and Iván Ruiz-Rube. Assessing foreign language learning through mobile game-based learning environments. *International Journal of Human Capital and Information Technology Professionals (IJHCITP)*, 7(2):53–67, 2016.
- [22] Manuel Palomo-Duarte, Anke Berns, Juan Manuel Dodero, and Alberto Cejas. Foreign language learning using a gamificated app to support peer-assessment. In *Proceedings of the Second International Conference on Technological Ecosystems for Enhancing Multiculturality*, pages 381–386, 2014.
- [23] Rubén Pérez-Mercado, Antonio Balderas, Andrés Muñoz, Juan Francisco Cabrera, Manuel Palomo-Duarte, and Juan Manuel Dodero. ChatbotSQL: Conversational agent to support relational database query language learning. *SoftwareX*, 22:101346, May 2023.
- [24] Tanvi Rana, Chaiya Laoteppitaks, Guiyun Zhang, Gregory Troutman, and Shruti Chandra. An investigation of Anki Flashcards as a study tool among first year medical students learning anatomy. *The FASEB Journal*, 34(S1):1–1, 2020.

Parte IV

Anexos

A. Manual del desarrollador

A.1. Introducción

La instalación de roBDa es relativamente sencilla para aquel desarrollador que posea ciertas nociones de programación web y de despliegue de aplicaciones.

A.2. Preparación del entorno de trabajo

Para realizar modificaciones en esta plataforma (y para su despliegue), es necesario cumplir una serie de requisitos:

- La aplicación usa una base de datos dentro de MySQL. Se necesita tener MySQL instalado, y haber creado una base de datos vacía con un nombre que posteriormente usaremos.
- Es necesario tener instalado Python (versión $\geq 3,8$) y pip (versión $\geq 20,0$).
- Es necesario tener instalado Node (versión $\geq 18,14$) y npm (versión $\geq 9,3$). Para el servidor donde se aloje la plataforma, también será de gran utilidad tener pm2 instalado.
- Git instalado y configurado.
- Al ser una aplicación web, es necesario tener un servidor web instalado en el servidor en el que se vaya a desplegar la aplicación. En nuestro caso, hemos usado Apache.

Una vez tengamos todo instalado, lo primero que tenemos que hacer será clonar el proyecto en nuestro equipo. Por tanto, debemos ejecutar el siguiente comando:

```
git clone https://github.com/rubenperez/roBDa.git
```

Cuando tengamos el código fuente en nuestro equipo, tendremos que crear un entorno virtual. Para ello, dentro de la carpeta con el código ejecutamos el comando:

```
python -m venv /path/to/new/virtual/environment
```

Donde `/path/to/new/virtual/environment` será la ruta en la que queramos crearlo y el nombre que le pongamos. En nuestro caso, hemos decidido crearlo en el mismo directorio raíz del proyecto (en el que ya estamos situados) y llamarlo `robda-venv`, luego el comando a ejecutar es:

```
python -m venv ./robda-venv
```

Una vez tengamos el entorno virtual, accedemos a él:

```
source robda-venv/bin/activate
```

Hay que recordar que se debe cambiar `robda-venv` por el nombre que hayamos elegido. Tras haber hecho esto, nos aparecerá el nombre del entorno virtual entre paréntesis al comienzo de la línea de comandos.

A.2.1. Configuración de la API

Dentro del entorno virtual, nos situamos en el directorio `/api`, e instalamos las dependencias:

```
pip install -r requirements.txt
```

De esta manera, instalamos todo lo necesario para configurar el resto de cosas. Nos quedaría por realizar las migraciones de los modelos a la base de datos. Primero de todo, debemos crear el fichero `.env` a partir del fichero `.env.example`:

```
mv .env.example .env
```

```
DEBUG=
```

```
SECRET_KEY=
```

```
DB_NAME=
```

```
DB_USERNAME=
```

```
DB_PASSWORD=
```

```
DB_HOST='localhost'
```

```
DB_PORT='3306'
```

Es necesario rellenar la información de la siguiente manera:

- `DEBUG=True` si estamos en el entorno de desarrollo, `False` si estamos en producción.
- A `DB_NAME` le asignamos el nombre de la base de datos vacía que previamente habíamos creado en MySQL.
- `DB_USERNAME` y `DB_PASSWORD` son el nombre de usuario y contraseña con el que accedemos a MySQL. Debe tener permisos suficientes sobre la base de datos previamente creada.
- Para la `SECRET_KEY`, existen páginas web como Djecrety¹, que generan esta clave. Simplemente copiamos la clave que nos devuelva la web y se la asignamos a esta variable.
- Por último, añadimos la palabra `export` delante de todas las variables para que se pueda acceder a ellas desde el resto de ficheros.

Cuando tengamos configurado este fichero, podremos migrar la base de datos:

```
python manage.py makemigrations
```

¹Djecrety - <https://djecrety.ir/>


```
python manage.py migrate
```

Hecho esto, podemos crear un superusuario en la API de la siguiente manera:

```
python manage.py createsuperuser
```

Nos pedirán los datos necesarios para crear el superusuario. Este superusuario tendrá todos los privilegios posibles, y por tanto, podrá acceder a la web como si de un profesor se tratase. No obstante, para mayor seguridad los profesores no tendrán todos los privilegios. Para cambiar el rol de una cuenta de un usuario ya registrada en la plataforma y que tenga privilegios de profesor, se puede utilizar la shell de Django:

```
python manage.py shell
```

```
>> from apps.users.models import User
```

```
>> User.objects.get(username={nombre-usuario}).update(is_staff=True)
```

Hay que sustituir `{nombre-usuario}` por el nombre de usuario al que queramos cambiarle el rol.

Por último, para lanzar el servidor, usamos el comando:

```
python manage.py runserver
```

El servidor se lanzará en `http://localhost:8000/`. Es importante, si accedemos directamente a la API, añadir la barra al final de la URL en cualquier petición.

A.2.2. Configuración de la web

La web, por otra parte, necesita menos configuración que la API. Si nos movemos al directorio `/web`, podemos instalar las dependencias usando el comando:

```
npm install
```

En este mismo directorio, veremos que también existe un fichero `.env.example`. Tendremos que renombrarlo como `.env` (al igual que se hizo en la anterior sección), y configurar las siguientes variables:

```
SECURE=  
NEXT_PUBLIC_API_URL=
```

- `SECURE='False'` si no estamos usando el protocolo HTTPS, (principalmente para entornos de desarrollo). En general, en producción valdría `'True'`
- A `NEXT_PUBLIC_API_URL` le asignaremos la URL de la API. En entorno de desarrollo, esta URL es `http://127.0.0.1:8000`. En nuestro entorno de producción, es `https://robda.uca.es/drf`.

Para lanzar el servidor desde el entorno de desarrollo, nos bastará con ejecutar el comando:

```
npm run dev
```

En el entorno de desarrollo, tendremos que compilar antes la página, posteriormente, podremos lanzar la web:

```
npm run build
```

```
npm start
```

El primero de los comandos anteriores es estrictamente necesario en caso de haber hecho modificaciones en la web.

Para que la página esté siempre activa en el servidor, hacemos uso de *pm2*. Para añadir un servicio:

```
pm2 start npm --name {nombre-aplicacion} -- start
```

En nuestro caso, `{nombre-aplicacion}` es *robda*. Aquí se muestran algunos comandos útiles:

- Parar servicio: `pm2 stop robda`
- Reanudar servicio: `pm2 start robda`
- Reiniciar servicio: `pm2 restart robda`
- Listar servicios disponibles: `pm2 list`
- Ver info del servicio: `pm2 show robda`
- Ver logs del servicio: `pm2 logs robda`

En cualquier caso, la web estará disponible en `http://localhost:3000`

B. Manual de usuario

B.1. Introducción

ROBDA es una plataforma web educativa de la Universidad de Cádiz, centrada en el aprendizaje de teoría de la asignatura de Bases de Datos. Es una plataforma basada en cuestionarios de tipo test con preguntas de distintos temas.

El objetivo de la plataforma es ofrecer una experiencia gamificada al alumno de esta asignatura, al mismo tiempo que el profesor puede supervisar y monitorizar todo lo acontecido en la plataforma.

B.2. Instalación

Para poder usar ROBDA únicamente se necesita un dispositivo con acceso a internet y navegador. La plataforma es accesible desde la dirección <http://robda.uca.es>, si bien es cierto que también se dispone de una aplicación web progresiva (PWA) para facilitar el acceso. Para instalar dicha PWA, tenemos que pulsar el icono de instalación señalado con una flecha naranja en la figura B.1, y posteriormente confirmar la instalación en el diálogo que nos aparecerá tal y como se muestra en la figura B.2. Una vez instalada, nos aparecerá un icono en el escritorio, y podremos acceder a la plataforma como si de una aplicación de escritorio se tratase (véase la figura B.3).



Figura B.1: Instalación de la PWA en Windows 11. Icono de instalación

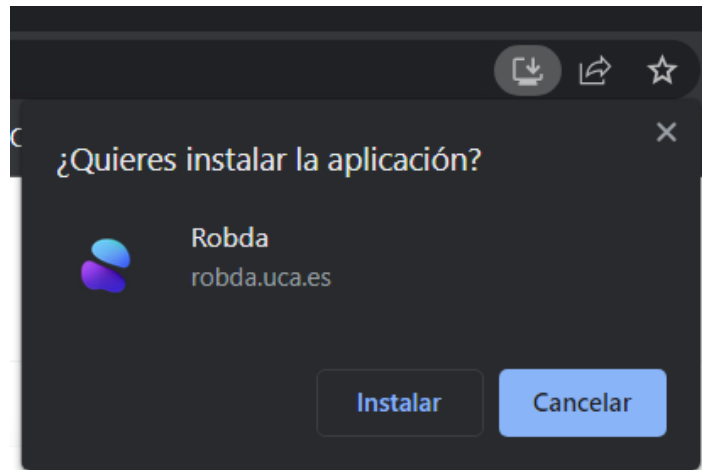


Figura B.2: Instalación de la PWA en Windows 11. Diálogo de confirmación

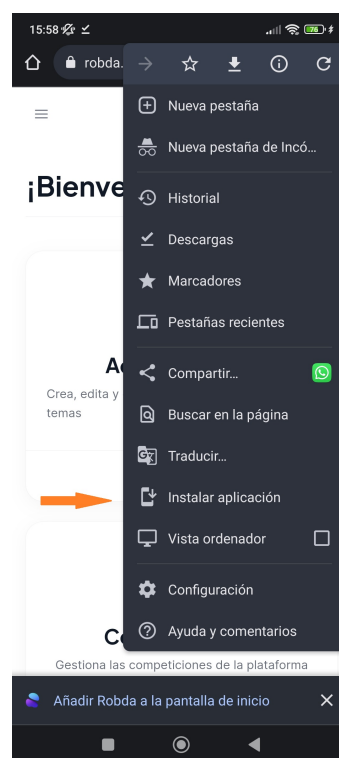


Figura B.3: Instalación de la PWA en Windows 11. Icono y aplicación en el escritorio

Para instalar la PWA desde dispositivos móviles, es posible que nada más acceder a la web nos aparezca un mensaje como el que hemos destacado en la figura B.4a. En caso de que no aparezca ningún mensaje, podemos abrir el menú de opciones del navegador (en nuestro caso, pulsando en los tres puntos), y aparecerá una opción que nos permita instalarla, como mostramos en la figura B.4b.



(a) Mensaje de instalación



(b) Instalación desde opciones

Figura B.4: Instalación de la PWA en Android

B.3. Uso del sistema

B.3.1. Acceso al sistema

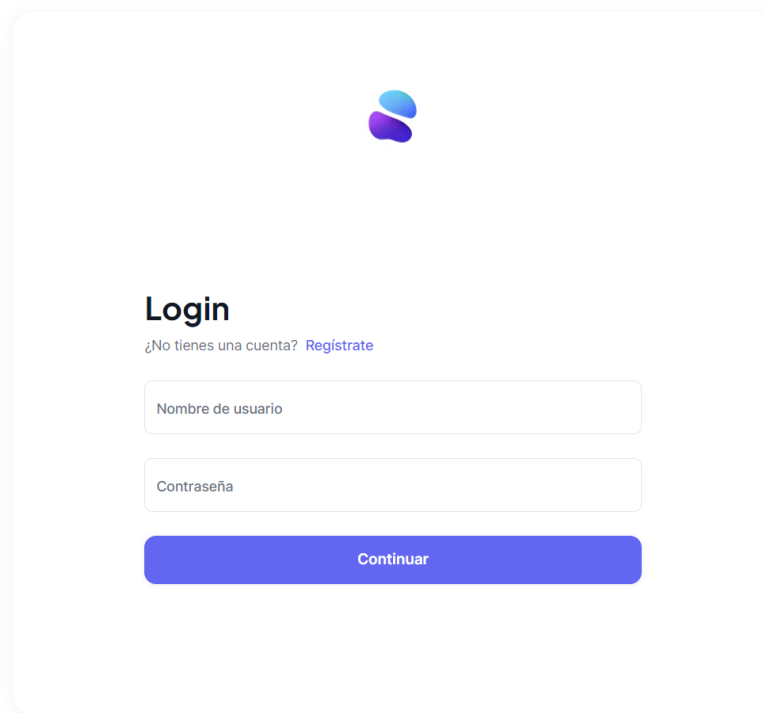
Nada más entrar la plataforma, nos aparecerá la pantalla de inicio de sesión. Tal y como podemos ver en la figura B.5, tendremos que introducir un nombre de usuario y una contraseña válidos. En caso de no disponer de una cuenta en la plataforma, podemos dirigirnos a la pantalla de registro pulsando en el enlace *Regístrate*. Una vez en la pantalla de registro, podremos ver un formulario como el de la figura B.6, en el que tendremos que introducir un nombre usuario, un correo, y la contraseña.

B.3.2. Perfil

Si pulsamos el botón con nuestro nombre en la barra lateral, nos dirigiremos a la pantalla de perfil. Allí podremos modificar los datos de nuestro perfil (figura B.7).

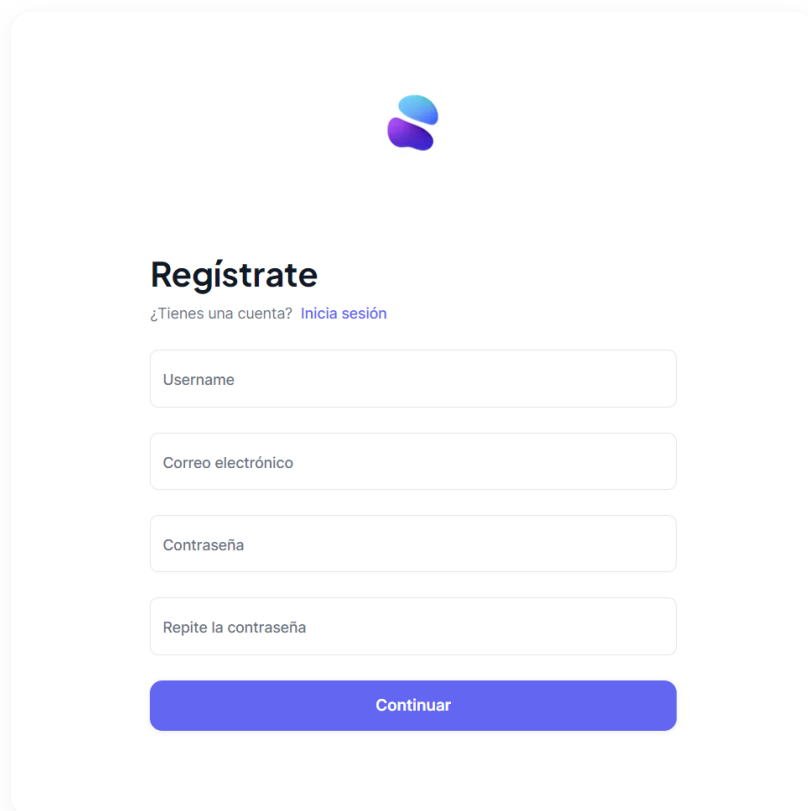
B.3.3. Acerca de

Todos los usuarios de la plataforma tendrán también una página en común con información acerca del proyecto. Se puede ver en esta pantalla una descripción de



The login screen features a logo at the top center consisting of two overlapping circles, one light blue and one purple. Below the logo is the heading "Login" in bold black text. Underneath the heading is a link: "¿No tienes una cuenta? [Regístrate](#)". There are two input fields: "Nombre de usuario" and "Contraseña". At the bottom is a blue button labeled "Continuar".

Figura B.5: Pantalla de inicio de sesión



The registration screen features the same logo at the top center. Below the logo is the heading "Regístrate" in bold black text. Underneath the heading is a link: "¿Tienes una cuenta? [Inicia sesión](#)". There are four input fields: "Username", "Correo electrónico", "Contraseña", and "Repite la contraseña". At the bottom is a blue button labeled "Continuar".

Figura B.6: Pantalla de registro

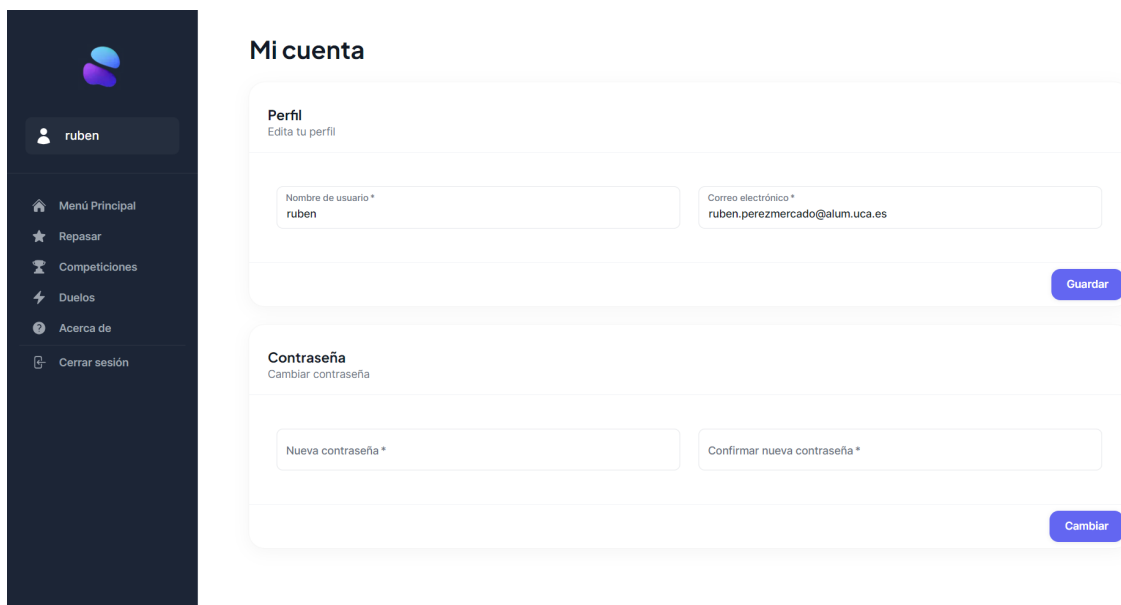


Figura B.7: Pantalla de edición del perfil

la plataforma y los participantes del proyecto, así como un enlace al repositorio del proyecto. En la figura B.8 podemos ver esta pantalla.



Figura B.8: Pantalla de información del proyecto

B.3.4. Alumno

Los usuarios que accedan a la plataforma con un perfil de alumno podrán ver el menú principal de la figura B.9. Las opciones que puede elegir el alumno son las siguientes:

- Repasar: Se redirige al modo de juego Repaso.

- Competiciones: Se redirige al usuario al modo de juego Competiciones
- Duelos: Se redirige al usuario al modo de juego Duelo
- Más información: Se redirige al usuario a la pantalla de información del proyecto.

También podemos observar en este menú y en la mayoría de pantallas de la plataforma una barra lateral que nos permitirá navegar por las distintas pantallas de forma más sencilla. También muestra el nombre del usuario actual, y pulsando sobre su nombre podremos acceder a la pantalla de edición de perfil.



Figura B.9: Menú principal del alumno

B.3.4.1. Modo Repaso

El modo Repaso es un modo de juego pensado para responder preguntas de un tema e idioma indicados por el propio usuario de forma continua. La selección de las preguntas se realiza de forma inteligente, de modo que el sistema calcula las preguntas que podrían ser más difíciles para el usuario y les otorga una mayor probabilidad de aparición.

Nada más acceder al modo Repaso el sistema nos mostrará un formulario simple como el de la figura B.10. Para comenzar una partida, tendremos que indicar un tema y un idioma. Puede darse el caso en que el sistema no sea capaz de comenzar una partida por no haber preguntas suficientes para garantizar un repaso de calidad. En caso de no saber cuáles son los temas con los que se pueden jugar partidas, contacte con el profesor.

Al pulsar en el botón *Jugar*, el sistema creará una partida y mostrará una pregunta por pantalla. La pregunta se verá igual que en la figura B.11. Podremos seleccionar cualquiera de las opciones pulsando sobre ella, cambiando la opción a un color morado, así como pulsar sobre la imagen para ampliarla. Cuando tengamos decidida nuestra respuesta, podremos pulsar el botón *Comprobar*, y nos mostrará cual era la respuesta

← Volver al menú

Elige un tema y un idioma para empezar a jugar

Tema
ÁLG

Idioma
Español

Jugar

Figura B.10: Formulario de inicio de repaso

correcta. En caso de haber acertado, se verá resaltada la opción elegida de color verde (figura B.12). En caso de fallo, se mostrará la opción seleccionada de color rojo, y la respuesta correcta de color verde (figura B.13). En cualquiera de los casos, nos aparecerá abajo de la pregunta un selector de cinco estrellas, con el que se pedirá valorar la dificultad de la pregunta, y un botón para reportar la pregunta, que abrirá un formulario para indicar el motivo y una breve descripción del reporte (figura B.14). Tras terminar la valoración (o el reporte) de cada pregunta, el usuario tendrá la opción de dar por finalizado el repaso, o de pedir otra pregunta.

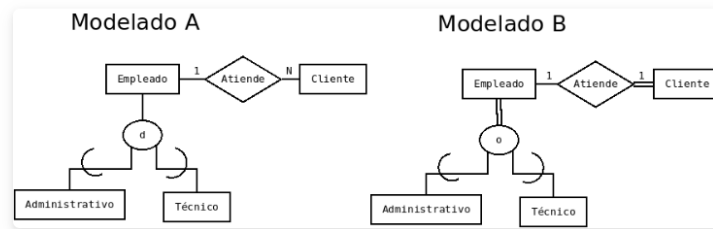
B.3.4.2. Modo Competición

Este modo de juego permite a los alumnos participar en eventos especiales creados por los profesores. Los alumnos pueden ganar puntos completando una serie de acciones, que les ayudarán a escalar puestos en la clasificación final de la competición. Los alumnos podrán ver todos los detalles de las competiciones accediendo a ellas, y verán una pantalla con toda la información (figura B.15) Las competiciones tienen un tema e idioma establecidos por el profesor, y están compuestas de varias fases:

- Fase de creación de preguntas: Se le pide al alumno crear una pregunta del tema e idioma indicados en la información de la competición. La creación de la pregunta es necesaria para participar en las fases siguientes de la competición. En esta fase, se le proporciona al usuario un formulario como el de la figura B.16 para que cree su pregunta. Una vez creada, tendrá que esperar a que la fase termine para acceder a la siguiente. Mientras tanto, podrá ver en el detalle de la participación la pregunta que ha creado (figura B.17). El alumno ganará puntos con la creación de la pregunta si, al terminar la competición, no existe ningún reporte asociado a la pregunta que el profesor haya dado como válido. En ese caso, el creador de la pregunta ganará 50 puntos, y en caso contrario, no ganará ninguno.

REPASO

Indique cuál de las siguientes es una diferencia entre los modelados A y B



En el modelado A pueden existir Empleados genéricos, Empleados tipo Administrativos y Empleados tipo Técnicos, pero no Empleados tipo Administrativo-Técnico
 En el modelado B se permite que existan Empleados tipo Administrativos, Empleados tipo Técnicos, y Empleados tipo Administrativo-Técnico

En el modelado A pueden existir Empleados genéricos, Empleados tipo Administrativos, Empleados tipo Técnicos y Empleados tipo Administrativo-Técnico
 En el modelado B NO se permite todas las posibilidades

En el modelado A pueden existir Empleados tipo Administrativos y Empleados tipo Técnicos, pero no Empleados tipo Administrativo-Técnico
 En el modelado B se permiten Empleados genéricos, Empleados tipo Administrativos, Empleados tipo Técnicos, y Empleados tipo Administrativo-Técnico

En el modelado A pueden existir Empleados tipo Administrativos y Empleados tipo Técnicos y Empleados tipo Administrativo-Técnico
 En el modelado B se permiten Empleados genéricos, Empleados tipo Administrativos y Empleados tipo Técnicos, pero no Empleados tipo Administrativo-Técnico

Comprobar

Figura B.11: Cuestionario de repaso

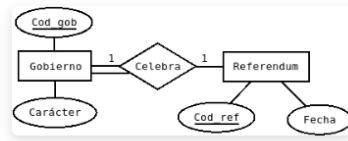
- Fase de cuestionario: En esta fase, los alumnos que hayan creado una pregunta antes podrán ver en la competición un botón para comenzar el cuestionario de la competición (figura B.18). Pulsando en este botón, se le mostrará al alumno la pantalla de la figura B.19, con información sobre el cuestionario a realizar y un botón para confirmar el inicio del cuestionario. Si el usuario decide iniciar el cuestionario, le aparecerá un cuestionario de 10 preguntas seleccionadas al azar, elegidas entre las que los compañeros han creado en la competición. En caso de no haber preguntas suficientes, se tomarán preguntas ya existentes en la plataforma, pero se dará prioridad siempre a las preguntas de los compañeros. El alumno tendrá un tiempo máximo de 15 minutos para responder las 10 preguntas, si bien puede ser menor en caso de que la fase de cuestionario termine en menos tiempo. El usuario gana 10 puntos por cada acierto conseguido en el cuestionario.

Puede darse el caso en que el alumno, por algún motivo, decida abandonar el cuestionario una vez abierto. En ese caso, antes de abandonar la página se le mostrará un mensaje de confirmación de abandono. Si realmente decide marcharse, el cuestionario no registrará ninguna de las respuestas que lleve hasta el momento.

- Fase de feedback: Esta fase está englobada dentro de la anterior. Cuando el alumno termine el cuestionario, el sistema mostrará la partida del usuario, pudiendo ver los aciertos y los errores cometidos (figura B.20). Dispondrá el alumno

REPASO

¿Qué modela el siguiente diagrama E/ER?



Nuestro sistema almacena información de cero o más Gobiernos, y de cero o más Referendums.
Cada Gobierno celebra un referendun o ninguno, y cada referendun siempre lo celebra un gobierno.

Nuestro sistema almacena información de uno o más Gobiernos, y de uno o más Referendums.
Cada Gobierno celebra un referendun o ninguno, y cada referendun siempre lo celebra un gobierno.

Nuestro sistema almacena información de uno o más Gobiernos, y de uno o más Referendums.
Cada Gobierno siempre celebra un referendun, y cada referendun lo celebra un gobierno o puede que ninguno.

Nuestro sistema almacena información de cero o más Gobiernos, y de cero o más Referendums.
Cada Gobierno siempre celebra un referendun, y cada referendun lo celebra un gobierno o ninguno.

Valora la dificultad de esta pregunta:



Enviar valoración

Figura B.12: Acierto en una pregunta de repaso

de una lista de las preguntas del cuestionario, numeradas del 1 al 10, y con colores diversos:

- Verde: Pregunta acertada
- Rojo: Pregunta fallada
- Pregunta leída, pero no respondida
- Pregunta no leída y no respondida

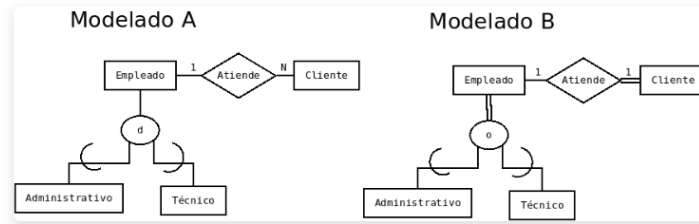
El usuario puede pulsar en cada una de los botones de la lista y podrá ver el detalle de esa pregunta concreta. También podrá reportar las preguntas que considere que están mal siempre que lo desee, pulsando sobre el icono correspondiente. Cada pregunta individual puede ser reportada una vez como máximo. Por cada reporte que el profesor considere válido, se añadirán 15 puntos al alumno que haya reportado.

Además, el usuario puede ganar 20 puntos extra si valora la competición en una escala del 1 al 5. Para ello cuenta con un selector de estrellas ubicado arriba del resumen de la partida (figura B.21).

- Cuando termine la fase de cuestionario, el evento quedará pendiente de la corrección del profesor. Esto es necesario ya que pueden existir reportes que decidan el resultado final de la competición. Una vez el profesor vea todos los reportes y termine la competición, los alumnos podrán ver las mejores cinco puntuación obtenidas por otros usuarios, y el informe completo de su participación, como se

REPASO

Indique cuál de las siguientes es una diferencia entre los modelados A y B



En el modelado A pueden existir Empleados genéricos, Empleados tipo Administrativos y Empleados tipo Técnicos, pero no Empleados tipo Administrativo-Técnico
En el modelado B se permite que existan Empleados tipo Administrativos, Empleados tipo Técnicos, y Empleados tipo Administrativo-Técnico

En el modelado A pueden existir Empleados genéricos, Empleados tipo Administrativos, Empleados tipo Técnicos y Empleados tipo Administrativo-Técnico
En el modelado B NO se permite todas las posibilidades

En el modelado A pueden existir Empleados tipo Administrativos y Empleados tipo Técnicos, pero no Empleados tipo Administrativo-Técnico
En el modelado B se permiten Empleados genéricos, Empleados tipo Administrativos, Empleados tipo Técnicos, y Empleados tipo Administrativo-Técnico

En el modelado A pueden existir Empleados tipo Administrativos y Empleados tipo Técnicos y Empleados tipo Administrativo-Técnico
En el modelado B se permiten Empleados genéricos, Empleados tipo Administrativos y Empleados tipo Técnicos, pero no Empleados tipo Administrativo-Técnico

Valora la dificultad de esta pregunta:



Enviar valoración

Figura B.13: Fallo en una pregunta de repaso

muestra en la figura B.22.

B.3.4.3. Modo Duelo

El último modo de juego que compone la plataforma son los Duelos. Los alumnos aquí podrán retarse los unos a los otros para realizar el mismo cuestionario sobre un tema e idioma determinados por el retador. Gana quien más aciertos consiga en el duelo.

Accediendo a esta sección, y tal y como muestra la figura B.23 el usuario podrá ver un listado de los duelos que ha creado o a los que ha sido retado a responder. Pulsando el botón morado de la parte superior derecha, podremos crear un duelo.

Para retar a otro compañero a un duelo, debemos conocer de antemano su nombre de usuario. Una vez lo tengamos, rellenamos el formulario de creación de duelo, en el que, además del rival, habrá que incluir un tema e idioma.

Cuando el duelo se cree, el sistema redirige al alumno al detalle del duelo. Inicialmente, el estado del duelo será “En Creación”, es decir, todavía tiene que jugar su partida el retador. No obstante, el retador verá en el detalle que el estado es “Su turno” (véase la figura B.24). Pulsando en el botón *Jugar*, aparecerá en la pantalla

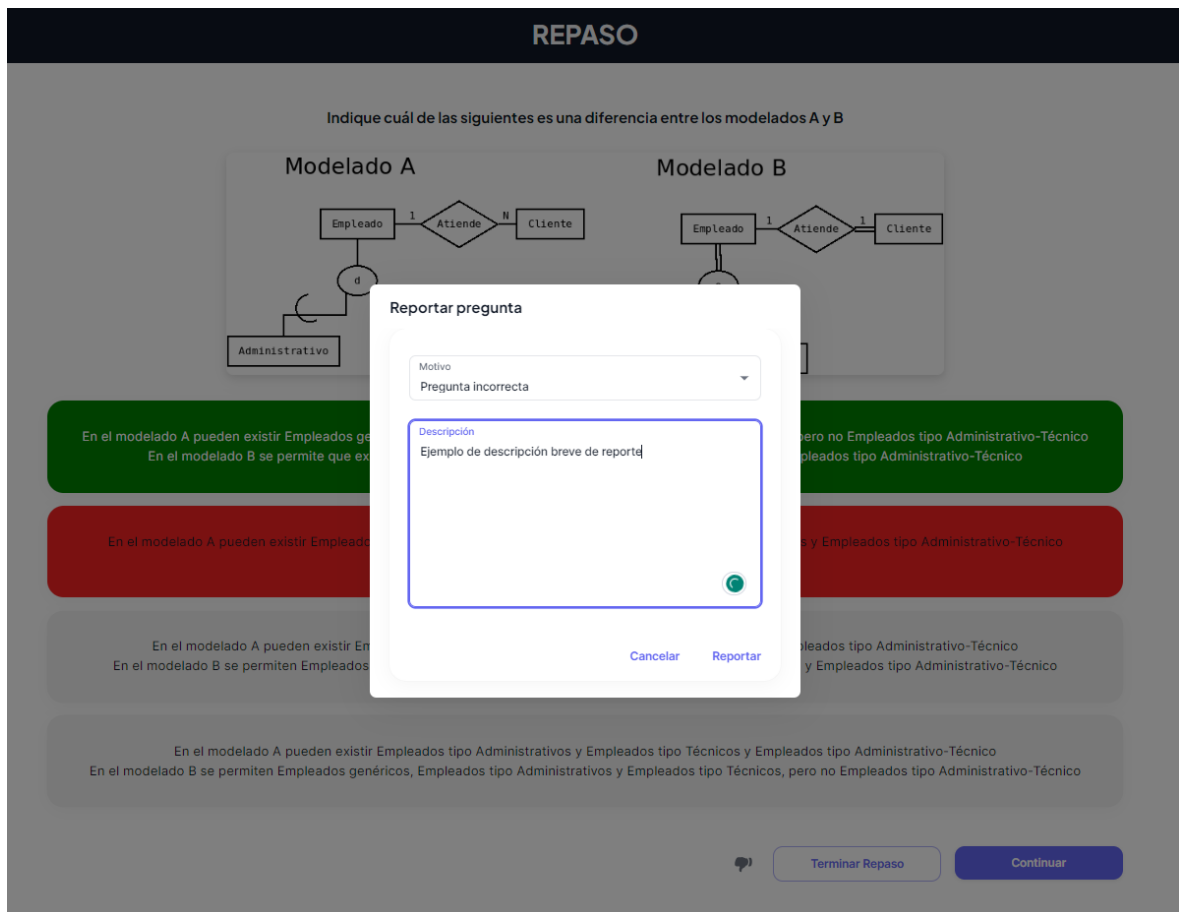


Figura B.14: Formulario de reporte

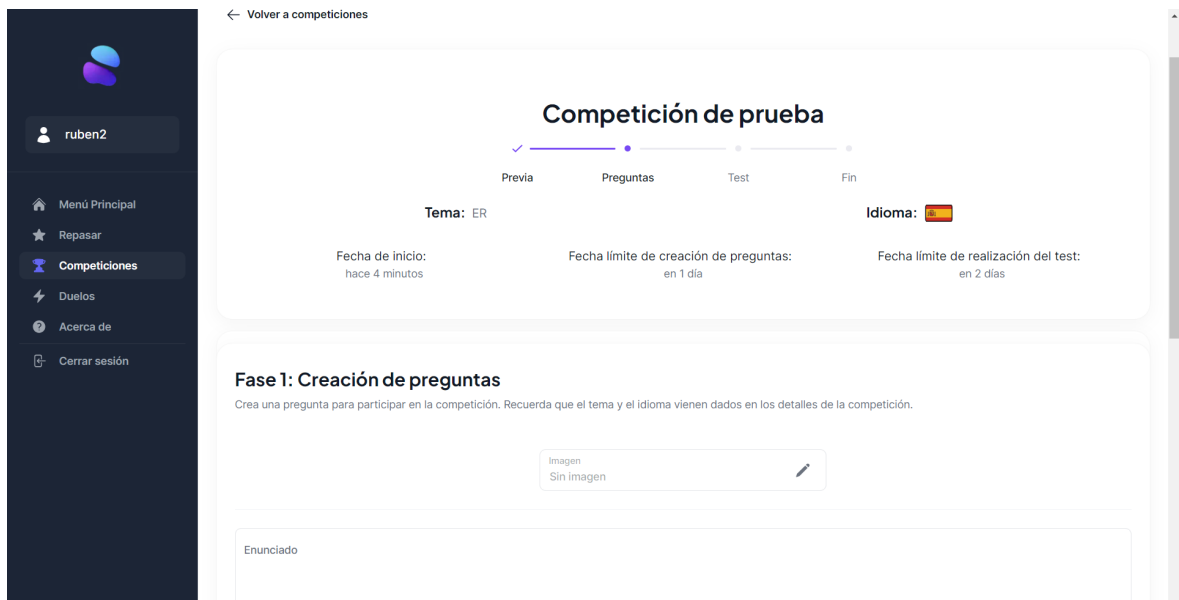


Figura B.15: Detalle de la competición

un cuestionario de 10 preguntas con una cuenta atrás de 15 minutos con preguntas del tema e idioma seleccionados (igual que en el modo Competición). La figura B.25 muestra el cuestionario con esa cuenta atrás.

Fase 1: Creación de preguntas

Creación de una pregunta para participar en la competición. Recuerda que el tema y el idioma vienen dados en los detalles de la competición.

Imagen
 Sin imagen ✎

Enunciado

Respuesta correcta

Opción incorrecta 1

Opción incorrecta 2

Opción incorrecta 3

Crear pregunta

Figura B.16: Formulario de creación de preguntas de competiciones

ruben2

- Menú Principal
- Repasar
- Competiciones
- Duelos
- Acerca de
- Cerrar sesión

Detalle competición

[← Volver a competiciones](#)

Competición de prueba

✓ Previa
Preguntas
Test
Fin

Tema: ER

Fecha de inicio: hace 0 minutos

Fecha límite de creación de preguntas: en 1 día

Idioma:

Fecha límite de realización del test: en 2 días

🔔 Ya has creado la pregunta. Prepárate para la siguiente fase.

Fase 1: Creación de preguntas

Esta es un pregunta sin imagen de prueba. La ha creado el usuario ruben2

Respuesta correcta

Respuesta incorrecta 1

Respuesta incorrecta 2

Respuesta incorrecta 3

Figura B.17: Detalle de la competición tras crear la pregunta



Figura B.18: Fase de cuestionario de la competición



Figura B.19: Inicio de cuestionario en competición

Terminado el cuestionario por parte del retador, se le redirigirá al detalle del duelo, y podrá ver que ahora el estado ha cambiado a “Pendiente” (figura B.26). A partir de este momento, el otro usuario podrá ver la partida en su listado de duelos, como en la figura B.27. Podrá rechazarlo (en cuyo caso no se notificará al otro usuario, pero la partida del mismo quedará registrada igualmente) o aceptarlo, siendo redirigido en el último caso al detalle del duelo. El retado no podrá ver el desempeño del retador en la partida antes de comenzar su partida. Una vez la termine, ambos usuarios podrán ver las partidas de ambos, seleccionando el usuario en las pestañas que podrá ver. Esto se muestra en la figura B.28. La estructura del informe de cada partida será la misma que la vista en el modo Competición. Destacar además que, si el usuario está viendo su propia partida, el botón para reportar preguntas estará visible, de modo que podrá notificar a los profesores de cualquier error en las preguntas.

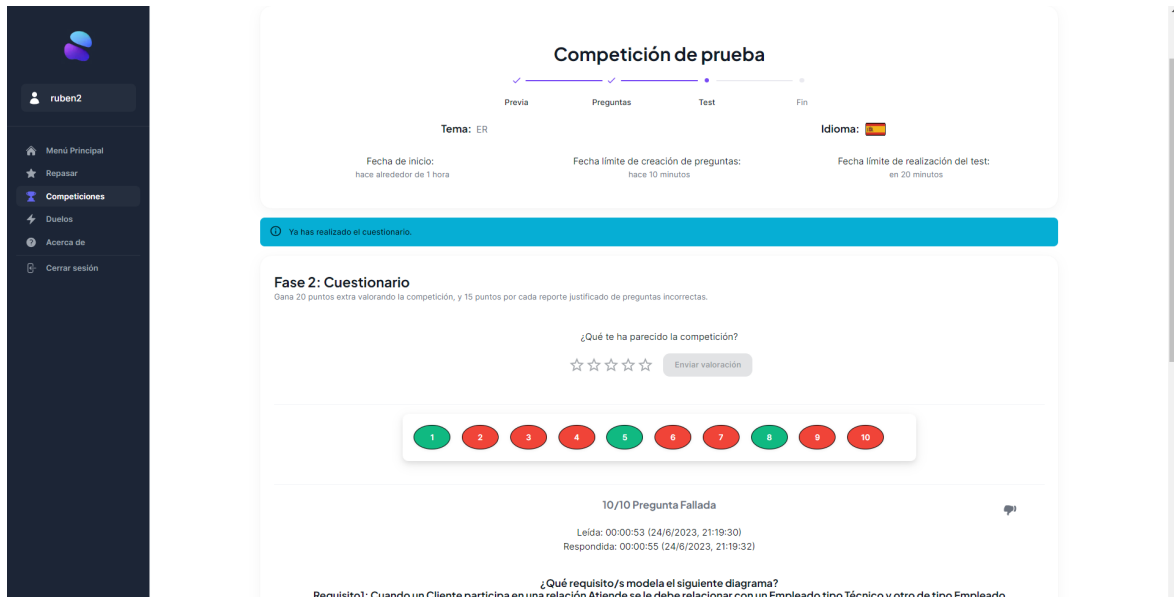


Figura B.20: Informe de la partida de la competición

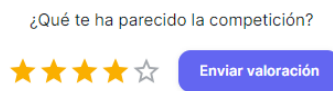


Figura B.21: Valoración de la competición

B.3.5. Profesor

La cuenta de profesor, por su parte, tendrá un menú distinto al del alumno. Este menú, visible en la figura B.29, contiene las siguientes opciones:

- Administración: Gestión del contenido principal de la plataforma: Preguntas, imágenes e idiomas.
- Competiciones: Gestión de las competiciones.
- Estadísticas: Estadísticas de uso de la plataforma e informes de las partidas de cada modo de juego.
- Más información: Pantalla de información del proyecto.

Asimismo, también cuenta con la barra lateral de navegación, con las secciones propias de su rol.

B.3.5.1. Administración

Nada más acceder al panel del administrador, el profesor podrá ver el listado de preguntas (figura B.30). En él, el profesor dispone de diversos filtros para acotar su búsqueda. Como se puede apreciar en la figura anterior, las preguntas reportadas

Detalle competición

← Volver a competiciones

Competición de prueba

✓ Previa ✓ Preguntas ✓ Test ✓ Fin

Tema: ER Idioma: 


Fecha de inicio: hace alrededor de 1 hora Fecha límite de creación de preguntas: hace 17 minutos Fecha límite de realización del test: hace 7 minutos

Mejores jugadores

POSICIÓN	NOMBRE	PUNTUACIÓN
1	ruben2	115
2	ruben	0

Resumen

Puntuación

-  Pregunta: 50
-  Cuestionario: 30
-  Feedback: 35

Valoración



★★★★☆

Pregunta creada

Figura B.22: Detalle de la competición finalizada

Duelos

+ Crear duelo

ENFRENTAMIENTO	TEMA	IDIOMA	ESTADO	ACCIONES
ruben vs. ruben2	ER		PENDIENTE	

1-1 of 1 < >

Figura B.23: Listado duelos

saldrán las primeras, y tendrán el número de reportes pendientes entre paréntesis al lado del estado.



Figura B.24: Fase inicial duelo

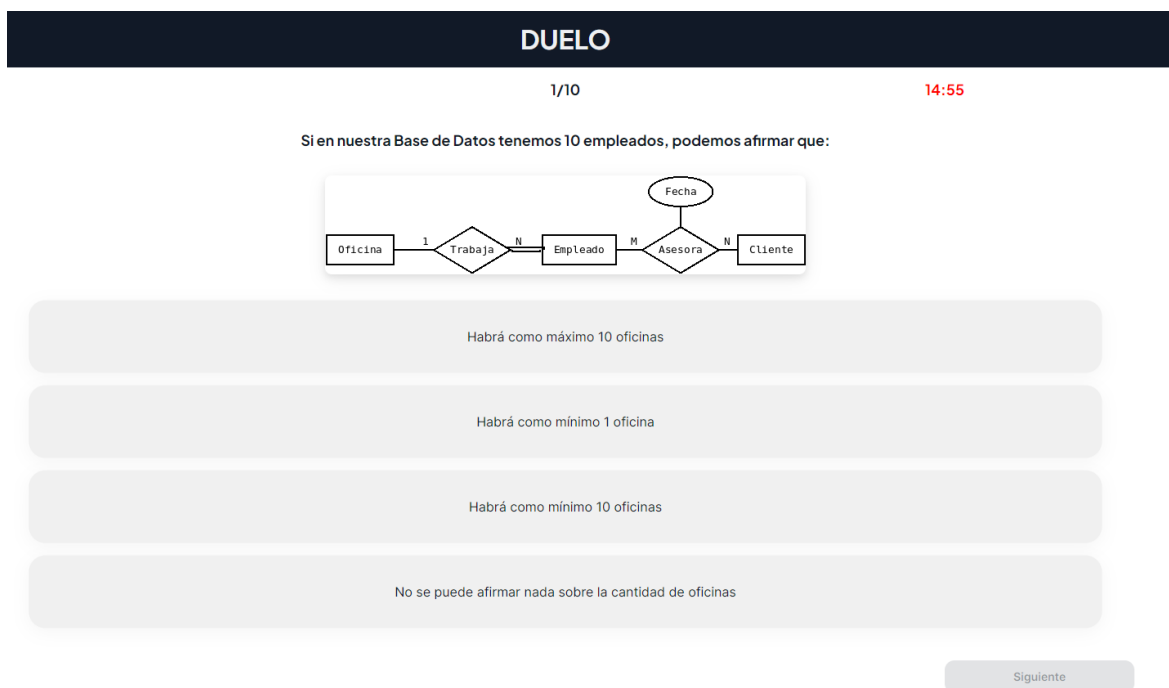


Figura B.25: Cuestionario de duelos

Si accedemos a cualquiera de las preguntas, podremos ver la previsualización de la pregunta, junto con el resto de información acerca de la misma (figura B.31). Además, más abajo podemos ver algunas estadísticas de la pregunta (si existen registros en la que la respondan) y el listado de reportes pendientes (figura B.32). Desde aquí, el profesor podrá decidir si los reportes son válidos o no. Se debe tener en cuenta que, en caso de preguntas creadas en eventos, aceptar un reporte añade 15 puntos a la persona que lo creó, mientras que rechazarlo no añade puntuación.



Figura B.26: Duelo pendiente

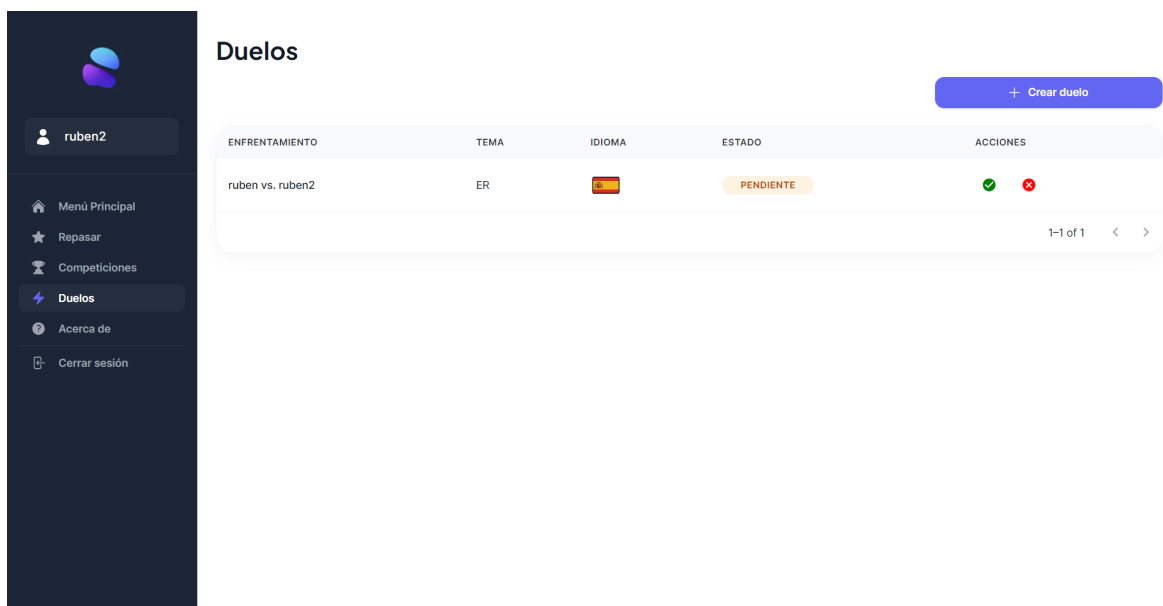


Figura B.27: Listado de duelos con duelo pendiente de decisión

Desde el detalle de la pregunta podemos habilitar el modo edición, con el que podremos editar o eliminar la pregunta.

Volviendo al listado de preguntas, podemos comprobar que existe un botón para crear preguntas en la parte superior derecha de la pantalla. Este botón abrirá un formulario con los datos necesarios para la creación de la pregunta.

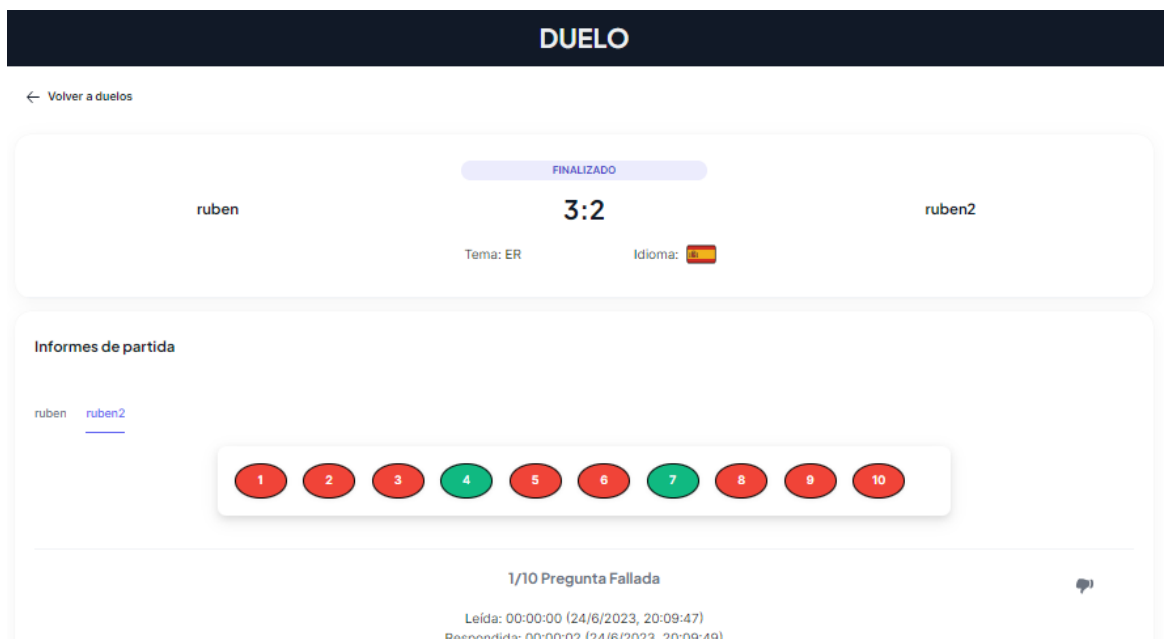


Figura B.28: Duelo finalizado

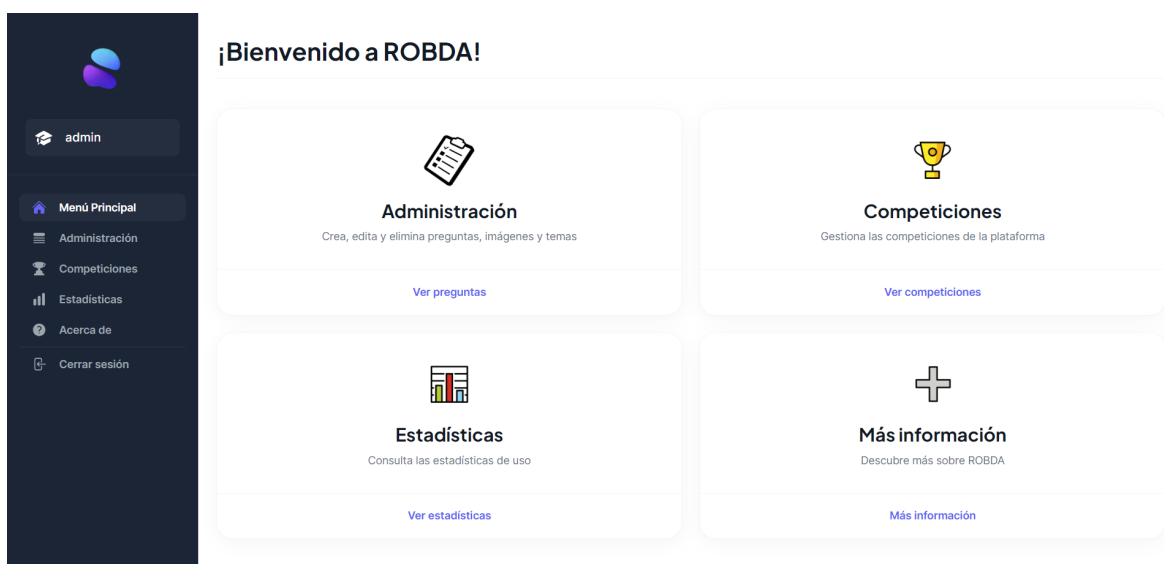


Figura B.29: Menú principal de los profesores

Moviéndonos por las pestañas del panel de administración, también podemos acceder al listado de imágenes y de temas, mostrados en las figuras B.33 y B.34 respectivamente. La creación, edición y la eliminación de preguntas y temas se asemeja a la de preguntas.

B.3.5.2. Competiciones

Este panel de administración es similar a los anteriores, pero está centrado en la gestión de las competiciones de la plataforma. El listado de competiciones será exactamente el mismo que podrán ver los alumnos, si bien el detalle de una competición

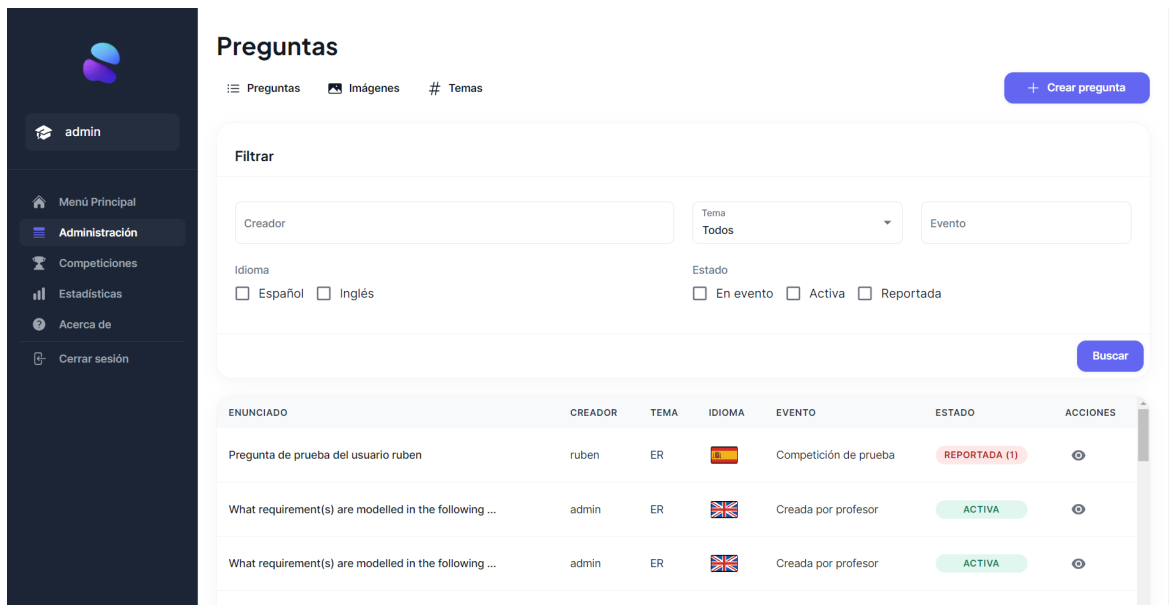


Figura B.30: Listado de preguntas

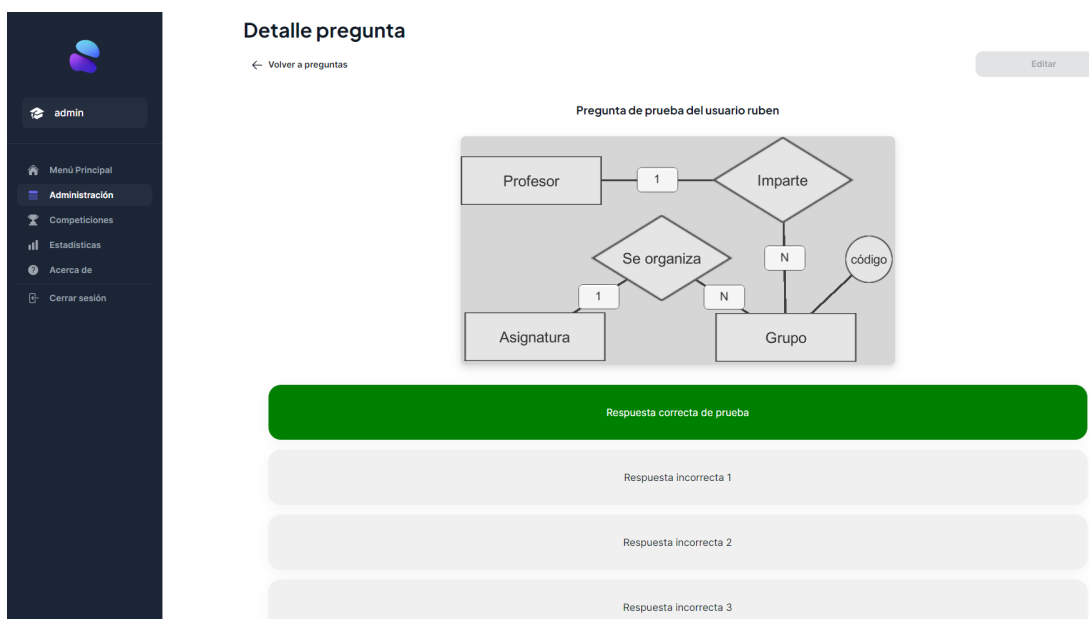


Figura B.31: Detalle de la pregunta

cambia, como puede verse en B.35. En el detalle de competiciones podremos ver dos botones distintos, uno para editar los datos de la misma y otro para darla por terminada. Este último botón estará activo si la competición está en la fase de corrección del profesor, y no existen reportes pendientes relacionados con la competición. Esto se debe a que las puntuaciones finales pueden variar con los reportes generados. Pulsando este botón (que se puede ver en la figura anterior), daremos por terminado el evento y publicaremos los resultados finales automáticamente. El botón desaparecerá, y aparecerá un ranking con las puntuaciones de todos los alumnos (véase la figura B.36).

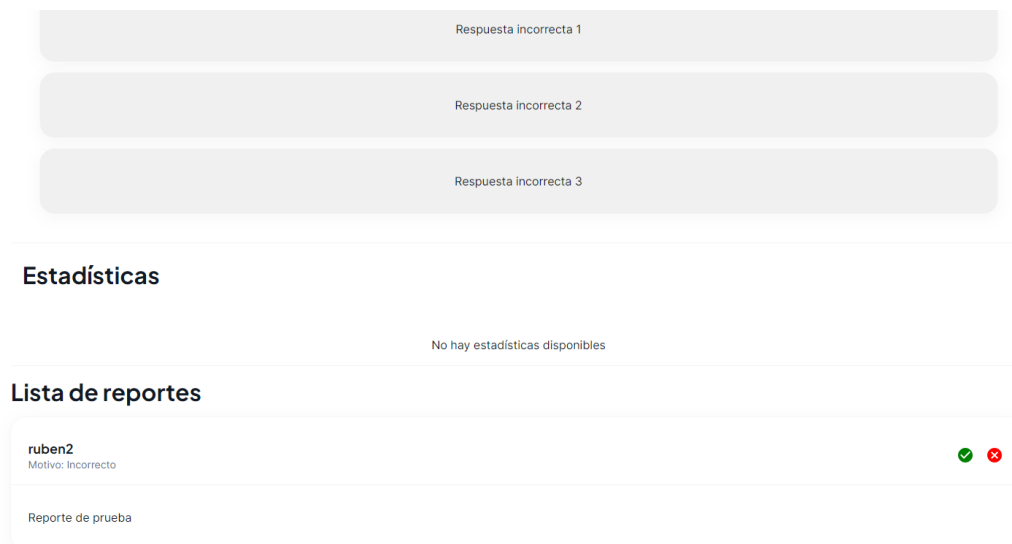


Figura B.32: Apartado de estadísticas y listado de reportes de una pregunta

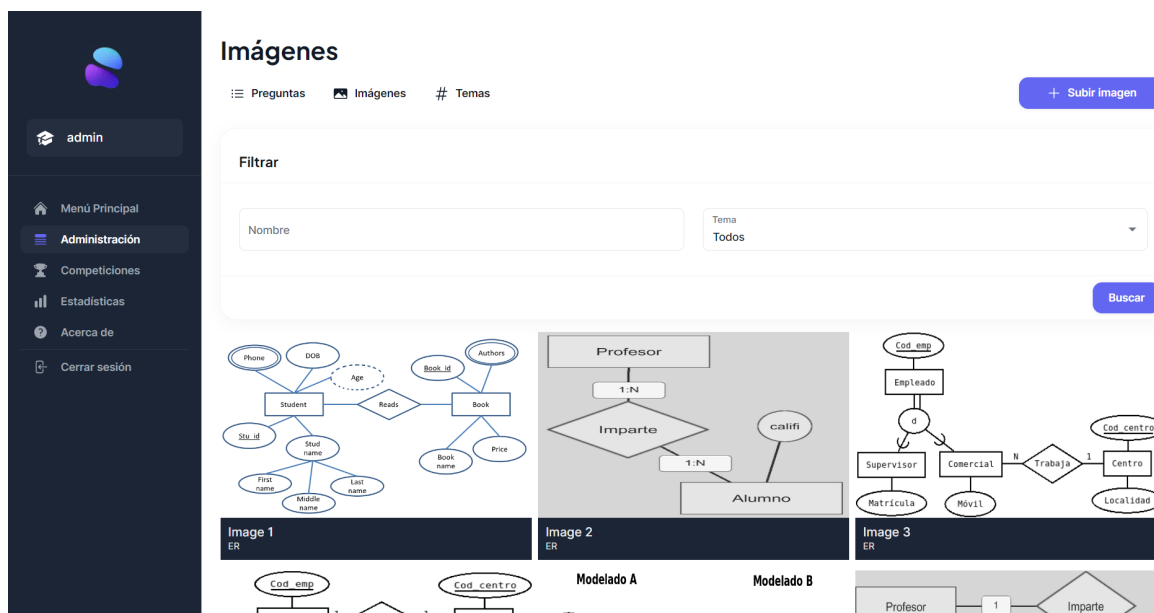


Figura B.33: Listado de imágenes

B.3.5.3. Estadísticas

Al entrar en este apartado, lo primero que veremos serán unas estadísticas generales de uso. Las figuras B.37 y B.38 representan algunas de las estadísticas visibles en esta pantalla.

Al igual que en el panel de administración, disponemos de distintas pestañas que nos permitirán ver más datos.

- Repasos: Nos ofrece un listado con todos los repasos de los alumnos.
- Competiciones: Nos lista todas las participaciones de competiciones llevadas a cabo en la plataforma.

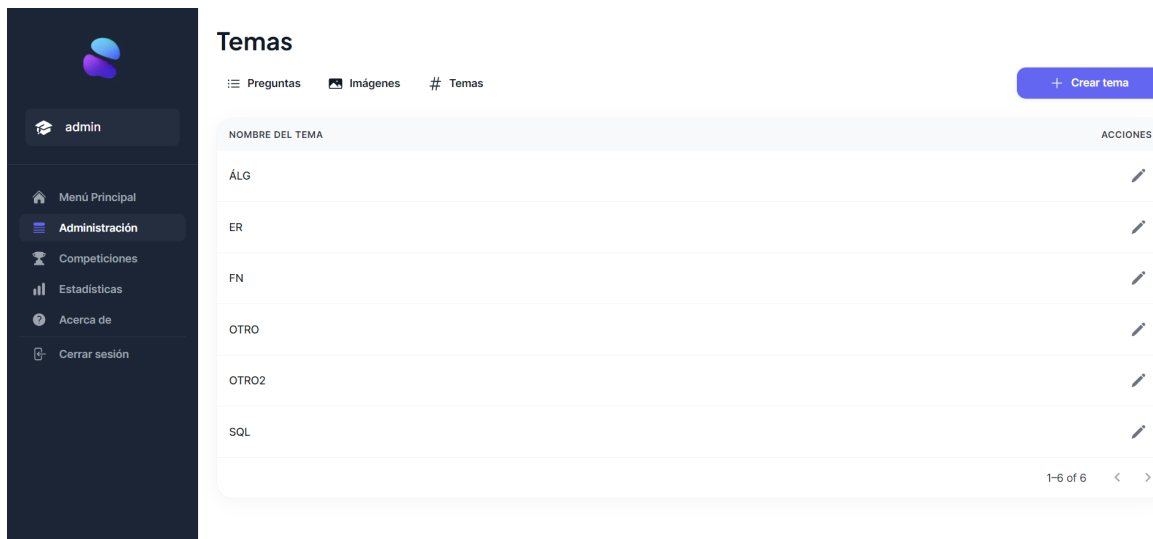


Figura B.34: Listado de temas

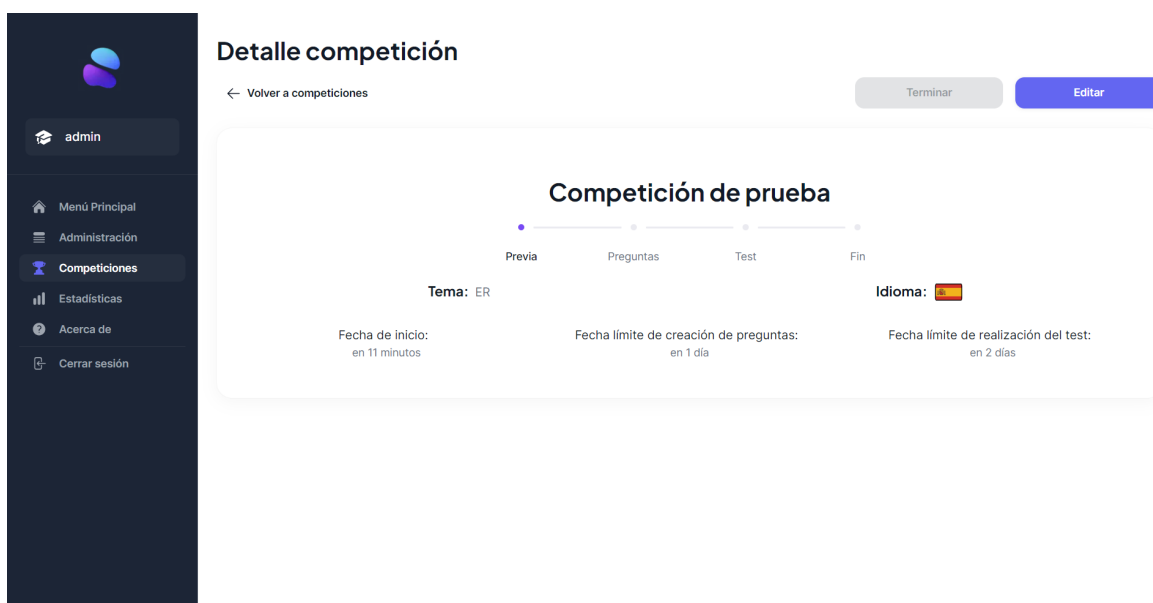


Figura B.35: Detalle de competición (profesores)

- Duelos: Muestra la lista de todos los duelos.

En todas estas secciones, existe un botón por cada registro que nos permitirá ver en detalle la partida seleccionada.



Figura B.36: Competición terminada (profesor)

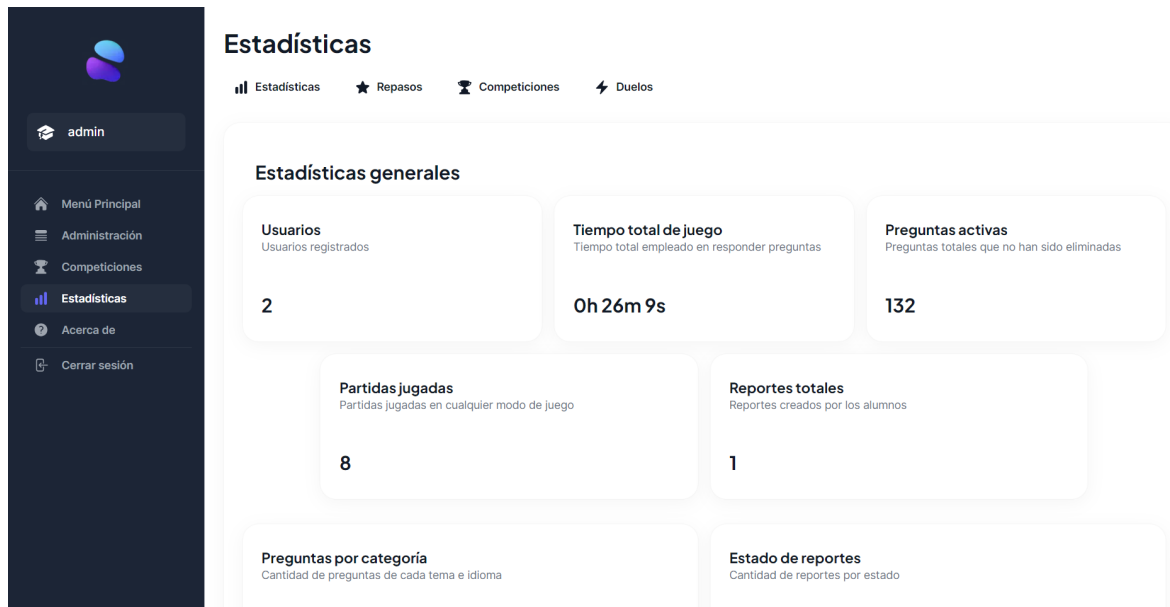


Figura B.37: Estadísticas I

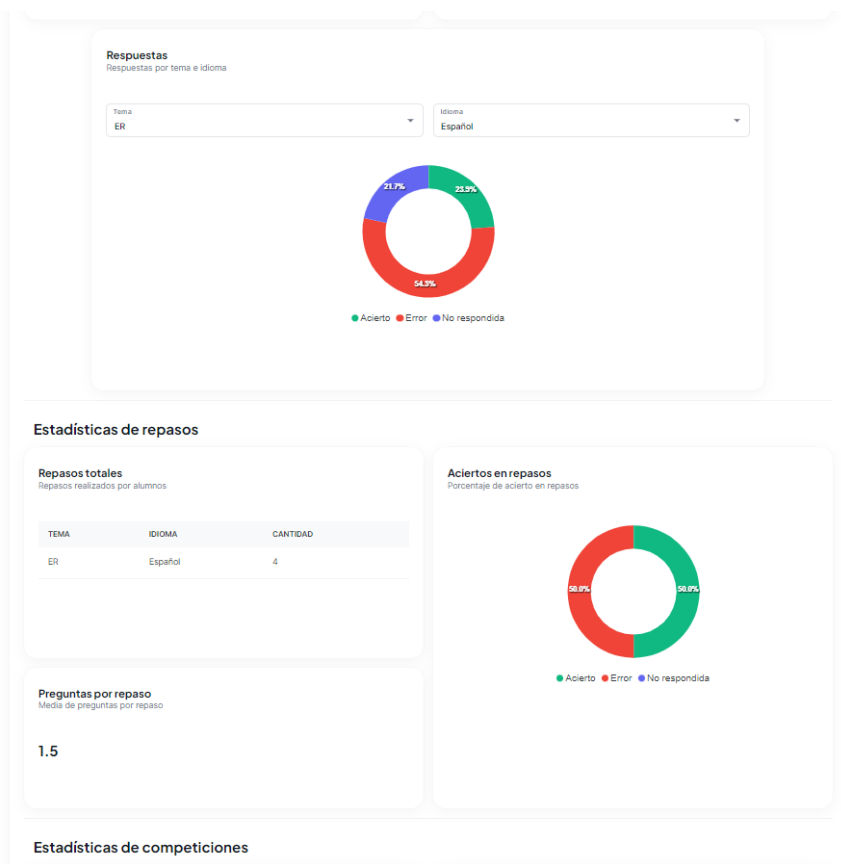


Figura B.38: Estadísticas II

