





ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA  
INFORMÁTICA

GRADO EN INGENIERÍA DEL SOFTWARE

**EDUKKA: PLATAFORMA DE GAMIFICACIÓN PARA EL  
APRENDIZAJE DE LOS NIÑOS**

**EDUKKA: GAMIFICATION PLATFORM FOR  
CHILDREN'S LEARNING**

Realizado por

**Javier Rosales Bañón**

Tutorizado por

**Eduardo Guzmán De los Riscos**

Departamento

**Lenguaje y Ciencias de la Computación**

UNIVERSIDAD DE MÁLAGA

MÁLAGA, JUNIO 2018

Fecha defensa:

El Secretario del Tribunal



**RESUMEN:** Este Trabajo de Fin de Grado (TFG) se enmarca en la temática de la educación en línea y, más concretamente, en el desarrollo de una plataforma para enseñanza en entorno web y dispositivos móviles. El e-learning permite la interacción del usuario con el material mediante la utilización de diversas herramientas informáticas. Este concepto educativo es una revolucionaria modalidad de formación que posibilitó Internet y se posiciona como la forma de aprendizaje predominante en el futuro. Para ello, la plataforma se ha dividido en dos módulos: una aplicación Web encargada de realizar las peticiones al servidor, y una aplicación Android encargada de cumplir los objetivos de aprendizaje para el usuario final. Ambos módulos han sido desarrollados de forma independiente, utilizando tecnologías distintas para la implementación de cada uno e integrándolos entre sí, formando la plataforma. Finalmente, el modelo de juego consigue motivar a los alumnos desarrollando un mayor compromiso de las personas e incentivando el ánimo de superación gracias a la gamificación, se han centrado los esfuerzos en ofrecer un producto que recoja cualidades del mercado y tecnologías actuales pero dedicado al público objetivo, los niños.

**PALABRAS CLAVE:** E-Learning, Gamificación, Android.

**ABSTRACT:** This Final Degree Project is based on the theme of online education and, more specifically, the development of a platform for teaching in web environment and mobile devices. The e-learning allows the user to interact with the material through the use of various computer tools. This educational concept is a revolutionary training modality that enabled the Internet, and it is positioned as the predominant form of learning in the future. For this, the platform has been divided into two modules: a Web application responsible of making requests to the server, and an Android application responsible of achieving the learning objectives for the end user. Both modules have been developed independently, using different technologies for the implementation of each one and integrating them with each other, forming the platform. Finally, the game model motivates students by developing a greater commitment of people and encouraging the spirit of improvement through gamification, I have focused to offer a product that reflects current market and technology qualities but dedicated to the target audience, children.

**KEYWORDS:** E-Learning, Gamification, Android.



# ÍNDICE GENERAL

CAPÍTULO 1. Introducción .....	1
1.1 Motivación .....	1
1.2 Objetivos .....	2
1.3 Estructura de la memoria .....	3
CAPÍTULO 2. Conocimientos previos .....	5
2.1 Aplicación Web .....	5
2.2 Aplicación Android.....	5
2.3 Herramientas utilizadas.....	6
2.4 Repositorio de código fuente .....	6
CAPÍTULO 3. Análisis .....	7
3.1 Requisitos Funcionales .....	7
3.2 Requisitos No Funcionales .....	9
3.3 Requisitos de Documentación .....	11
3.4 Descripción de Participantes y Usuarios .....	12
3.4.1 Participantes .....	12
3.4.2 Usuarios.....	13
3.5 Alternativas y Competencia.....	14
CAPÍTULO 4. Diseño .....	15
4.1 Casos de Uso.....	15
4.2 Modelo de Datos .....	36
4.3 Navegación .....	39
4.4 Servicios y Componentes.....	41
4.5 Arquitectura .....	43

CAPÍTULO 5. Implementación .....	45
5.1 Desarrollo del Back-end .....	45
5.2 Desarrollo del Front-end .....	48
5.2.1 Pantalla de inicio .....	51
5.2.2 Pantalla principal.....	53
5.2.3 Pantalla de perfil.....	55
5.2.4 Pantalla de clase .....	57
5.2.5 Pantalla de actividad.....	60
5.2.6 Pantalla de búsqueda .....	62
5.2.7 Pantalla de juego .....	63
5.3 Pruebas.....	68
5.4 Despliegue .....	69
5.4.1 Web Hosting.....	69
5.4.2 Google Play .....	70
CAPÍTULO 6. Conclusiones y líneas futuras .....	71
Referencias .....	73
Índice de figuras, tablas y códigos.....	75

# **CAPÍTULO 1. Introducción**

## **1.1 Motivación**

Los niños de Primaria, y quizás en especial aquellos de altas capacidades, deben adaptarse a un ritmo de clase que les puede parecer lento y aburrido, perdiendo el interés en el estudio. Para evitar esto, existen diversos programas de refuerzo que buscan ofrecer desafíos para entretenerlos y fomentar su aprendizaje a través de la gamificación, evitando posibles fracasos académicos a niños que tienen un futuro prometedor.

El e-learning para niños es el presente de la educación. Una de las claves importantes del funcionamiento de nuestra sociedad gira entorno a Internet. Estamos hablando de una comunicación absoluta, casi sin límites. Elementos como la emoción o la pasión, que nos decían antes que solo se podían transmitir personalmente, son perfectamente transmisibles a través de una pantalla (1).

La gamificación es una técnica de aprendizaje que traslada la mecánica de los juegos al ámbito educativo con el fin de conseguir mejores resultados, ya sea para absorber mejor algunos conocimientos, mejorar alguna habilidad, o bien recompensar acciones concretas, entre otros muchos objetivos.

Este tipo de aprendizaje gana terreno en las metodologías de formación debido a su carácter lúdico, que facilita la interiorización de conocimientos de una forma más divertida, generando una experiencia positiva en el usuario. Se utilizan una serie de técnicas mecánicas y dinámicas extrapoladas de los juegos. El modelo de juego realmente funciona porque consigue motivar a los alumnos, desarrollando un mayor compromiso de las personas, e incentivando el ánimo de superación (2).

Los niños de hoy en día aprenden con la tecnología porque les motiva, les divierte, les sorprende y es un medio natural para ellos. Ellos quieren y necesitan tener la capacidad de personalizar su experiencia de aprendizaje, de consumir información de forma rápida y de aportar experiencias de aprendizaje ricas en entornos inteligentes (3).

Esta generación TIC serán los consumidores de la formación e-learning del futuro.

## **1.2 Objetivos**

El objetivo principal consiste en desarrollar una plataforma con el estilo de arquitectura software REST donde los alumnos puedan realizar juegos y pruebas académicas basadas en contenidos de Educación Primaria. Para ello, se desarrollará un back-end basado en PHP desplegado en un servidor web junto a una base de datos SQL. Esta estructura enviará las respuestas en formato de texto ligero JSON al front-end para facilitar su posterior tratado, el cual consiste en la aplicación móvil de Android que se encargará de analizar y procesar los datos recibidos por el back-end para mostrarlos al usuario final. Se quiere implementar una plataforma robusta y ágil, que además permita el escalado en expansiones posteriores, ya sea añadir nuevas modalidades de juegos para los niños o incluir nuevos temarios aparte de los que se desarrollen como base.

Los alumnos podrán acceder fácilmente a la aplicación a través de su apodo y contraseña, una vez registrados tienen la posibilidad de ver y editar su propio perfil, ver los perfiles del resto de alumnos de su clase, realizar los juegos sobre las asignaturas disponibles y ver su progreso personal, que cambiará a medida que el alumno avance entre los juegos completados. El profesor es el otro perfil de usuario que ofrecerá la aplicación, tendrá las mismas herramientas que los alumnos y también se encargará de la gestión de la clase y de los juegos, una vez registrados tienen la posibilidad de crear y editar una clase, ver las soluciones de los juegos, ver los perfiles y el progreso general de los alumnos de su clase.

Dentro de la plataforma se desarrollará una herramienta para que el administrador pueda crear juegos usando la aplicación Web y que todos los alumnos puedan jugar a los nuevos juegos. Además, se van a realizar juegos de todas las asignaturas para mostrar su funcionalidad, que consistirá en varias preguntas quiz de diferentes tipos como son verdadero-falso, seleccionar uno, seleccionar varios, rellenar, etc.

### **1.3 Estructura de la memoria**

En esta sección se presenta brevemente la estructura de la memoria y una síntesis del contenido de cada capítulo.

- **Capítulo 1: Introducción**

En primer lugar, se expone el trabajo indicando los motivos por lo que surge la necesidad de la plataforma, los objetivos del mismo y un breve resumen del contenido de esta memoria.

- **Capítulo 2: Conocimientos previos**

Se expondrán las tecnologías empleadas durante la elaboración de la plataforma, además de los recursos utilizados como herramientas y librerías.

- **Capítulo 3: Análisis**

Se expondrán los requisitos funcionales, no funcionales y de documentación de la plataforma, así como la descripción de los usuarios y de las aplicaciones alternativas que existen actualmente en el mercado.

- **Capítulo 4: Diseño**

Se expondrán la especificación de los casos de uso y los diagramas sobre el diseño de la plataforma, que son el modelo de datos, navegación, servicios y arquitectura.

- **Capítulo 5: Implementación**

Se expondrá el desarrollo de la plataforma, se explicará con detalle la implementación del back-end y front-end, además de las pruebas ejecutadas y el despliegue.

- **Capítulo 6: Conclusiones y líneas futuras**

Por último, se expondrá lo aprendido durante la elaboración del trabajo y de discutir los posibles trabajos futuros que se podrían realizar.



## **CAPÍTULO 2. Conocimientos previos**

### **2.1 Aplicación Web**

En el desarrollo del back-end se ha utilizado NetBeans, que es un entorno de desarrollo integrado libre y, en este caso, enfocado al lenguaje de programación PHP.

La extensión PDO de PHP consiste en una capa de abstracción para acceder a diferentes tipos de bases de datos. Utilizando PDO se logran estandarizar los diferentes mecanismos para realizar la conexión a una base de datos, así como recuperar y modificar información. Además, se ha utilizado el micro framework Slim que ayuda a crear rápidamente la API de forma simple pero potente (4).

### **2.2 Aplicación Android**

En el desarrollo del front-end se ha utilizado Android Studio, que es el entorno de desarrollo integrado oficial para la plataforma Android, tiene una documentación completa y existen muchos recursos para el desarrollo (5).

La idea original está basada en la aplicación de Google llamada “Topeka”, la cual presenta la guía de estilo Material Design en Android, que es un nuevo sistema para el diseño visual, de interacción y de movimiento (6).

Además, para simplificar las tareas se han utilizado las siguientes librerías externas:

- Retrofit: realiza el envío de peticiones HTTP para comunicarse con el servidor web de manera muy simple y fácil de aprender, gestiona diferentes tipos de parámetros y parsea automáticamente la respuesta a un objeto de Java (7).
- StatusBarUtil: configura el estilo de la barra de estado (8).
- CircleImageView: muestra las imágenes de forma circular y adecuada para los perfiles de usuario (9).
- MaterialEditText: muestra los campos de texto con el estilo Material Design (10).
- MaterialBetterSpinner: muestra las listas desplegables de texto con el estilo Material Design (11).
- Picasso: permite cargar imágenes de manera simple y rápida (12).
- Lottie: analiza animaciones de Adobe After Effects exportadas como JSON y las renderiza de forma nativa en los dispositivos móviles (13).

## 2.3 Herramientas utilizadas

- **GitKraken:** herramienta para la gestión de los repositorios, es una potente y elegante interfaz gráfica multiplataforma para Git. De forma muy sencilla podemos llevar el completo seguimiento de nuestros repositorios, ver, crear y borrar ramas, observar todo el historial de nuestro trabajo, realizar commits, etcétera (14).
- **Postman:** herramienta para hacer peticiones a la API de la aplicación Web y generar colecciones que nos permitan probarlas de una manera rápida y sencilla (15).
- **MagicDraw:** herramienta CASE para modelar el diseño de la plataforma en UML, destinada a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero (16).
- **Balsamiq:** herramienta para crear una guía visual que representa el esqueleto o estructura visual de la aplicación Android. El prototipo se ha utilizado para generar ideas, descartar las malas y descubrir las mejores soluciones (17).
- **MySQL Workbench:** herramienta visual para el diseño de la base de datos de la plataforma que integra desarrollo de software, administración de bases de datos, diseño de bases de datos, gestión y mantenimiento para el sistema de base de datos MySQL (18).
- **WAMP:** sistema de infraestructura de internet Windows + Apache + MySQL + PHP para instalar la aplicación Web accesible desde nuestra red local (19).

## 2.4 Repositorio de código fuente

Los dos repositorios del trabajo están alojados en GitHub, la plataforma de programadores más grande del mundo, y de forma pública. De esta manera se da la oportunidad de que otras personas interesadas en este trabajo puedan tener acceso completo al código fuente e incluso colaborar en desarrollos posteriores.

Aplicación Web: <https://github.com/javoeria/edukka-server>

Aplicación Android: <https://github.com/javoeria/edukka-client>

## CAPÍTULO 3. Análisis

### 3.1 Requisitos Funcionales

En la Tabla 1 se enumeran los requisitos funcionales de la plataforma, que establecen los comportamientos del software y lo que el sistema hace para el cliente.

ID	Requisito	Descripción
RF01	Registrar	El sistema debe guardar la información personal de los nuevos alumnos y profesores, es obligatorio introducir estos datos correctamente en el inicio de sesión.
RF02	Iniciar Sesión	El sistema debe consultar los datos introducidos por los alumnos y profesores. Si el usuario está registrado en la base de datos entrará en la pantalla principal, en otro caso se mostrará un mensaje de error.
RF03	Ver Perfil	El sistema debe mostrar información de los alumnos y profesores que pertenecen a la clase del usuario.
RF04	Editar Perfil	El sistema debe permitir cambios en la información del perfil propio del usuario de una manera sencilla, clara y con una validación previa.
RF05	Borrar Perfil	El sistema debe permitir eliminar el perfil propio del usuario de la base de datos, el usuario debe confirmar esta acción destructiva.
RF06	Ver Clase	El sistema debe mostrar información de la clase a la que pertenece el usuario, también podrá ver una lista de usuarios ordenados por la puntuación obtenida en los juegos, de mayor a menor.
RF07	Crear Clase	El sistema debe permitir la creación de clases privadas a los profesores, es una característica importante para el buen funcionamiento del producto.
RF08	Editar Clase	El sistema debe permitir cambios en la información de la clase creada por el profesor de una manera sencilla, clara y con una validación previa.
RF09	Borrar Clase	El sistema debe permitir eliminar la clase creada por el profesor de la base de datos, los usuarios que estén dentro de dicha clase pasarán a la clase pública, el usuario debe confirmar esta acción destructiva.
RF10	Entrar Clase	El sistema debe permitir el acceso a una clase privada a los alumnos y profesores, para ello el usuario debe conocer el identificador de dicha clase y confirmar la salida de la clase pública.
RF11	Salir Clase	El sistema debe permitir el acceso a la clase pública a los alumnos y profesores, para ello el usuario debe confirmar la salida de la clase privada.
RF12	Ver Juego	El sistema debe mostrar información del juego seleccionado a los alumnos y profesores.

<b>RF13</b>	Crear Juego	El sistema debe permitir la creación de nuevos juegos y preguntas de una asignatura a los administradores. Esta es otra característica importante para mejorar el contenido del producto.
<b>RF14</b>	Editar Juego	El sistema debe permitir cambios en la información del juego a los profesores de una manera sencilla, clara y con una validación previa.
<b>RF15</b>	Borrar Juego	El sistema debe permitir eliminar el juego y todas sus preguntas de la base de datos a los profesores, el usuario debe confirmar esta acción destructiva.
<b>RF16</b>	Buscar Juego	El sistema debe permitir buscar los juegos deseados a los alumnos y profesores dentro de la lista de juegos disponibles para una asignatura.
<b>RF17</b>	Observar Juego	El sistema debe mostrar las soluciones de los juegos a los profesores, es un elemento fundamental para que el profesor pueda comparar las soluciones correctas con la respuesta media de la clase.
<b>RF18</b>	Realizar Juego	El sistema debe mostrar las preguntas de los juegos a los alumnos, la plataforma se basa principalmente en esta funcionalidad para el aprendizaje.
<b>RF19</b>	Ver Historial	El sistema debe mostrar el historial de los juegos completados con sus correspondientes resultados, la lista estará ordenada por fecha descendente. Los alumnos verán su propio historial, y los profesores verán el historial de todos los usuarios de la clase.
<b>RF20</b>	Ver Estadísticas	El sistema debe mostrar las estadísticas generales sobre los juegos completados. Los alumnos verán sus propias estadísticas, y los profesores verán las estadísticas de todos los usuarios de la clase.

*Tabla 1: Requisitos Funcionales*

## 3.2 Requisitos No Funcionales

En la Tabla 2 se enumeran los requisitos no funcionales de la plataforma, que son las características del sistema y las restricciones o condiciones que impone el cliente.

ID	Requisito	Descripción
<b>RNF1</b>	<b>Requisitos de Aspecto</b>	
<b>RNF1.1</b>	La interfaz	El sistema debe garantizar una interfaz intuitiva, simple y atractiva para los niños entre 6-12 años que están estudiando la Educación Primaria.
<b>RNF1.2</b>	El estilo del producto	El sistema debe asegurar un estilo sencillo que sea llamativo y colorido visualmente para los usuarios. Además, la guía de estilo del producto se centrará en Material Design de Google.
<b>RNF2</b>	<b>Requisitos de Facilidad de Uso y Aprendizaje</b>	
<b>RNF2.1</b>	Facilidad de Uso	El sistema va dedicado principalmente a niños de primaria, por lo que es vital realizar un diseño fácil que sea apto para un público de esa edad, se debe incluir una opción de ayuda para los usuarios.
<b>RNF2.2</b>	Facilidad de Aprendizaje	El sistema debe asegurar la visibilidad de todos los elementos que tengan funcionalidad y que queden remarcados por iconos ilustrativos de la acción a realizar cuando pulsas dicho elemento.
<b>RNF3</b>	<b>Requisitos de Funcionamiento</b>	
<b>RNF3.1</b>	Velocidad	El sistema debe garantizar la máxima fluidez en su funcionamiento, es decir, asegurar que la duración de las operaciones básicas sea menor a 5 segundos, y que la carga de las pruebas a realizar en un juego sea menor a 10 segundos.
<b>RNF3.2</b>	Seguridad Crítica	El sistema debe proteger la información personal de todos los usuarios para evitar posibles ataques externos a la plataforma, las contraseñas están cifradas para una mayor seguridad.
<b>RNF3.3</b>	Precisión	El sistema debe promover el aprendizaje del alumno con los juegos de las asignaturas, y garantizar un aumento en la puntuación personal de cada alumno.
<b>RNF3.4</b>	Fiabilidad y Disponibilidad	El sistema siempre debe estar disponible para el uso de los usuarios, excepto durante las tareas de mantenimiento o problemas graves de la aplicación.
<b>RNF3.5</b>	Capacidad	El sistema debe soportar una base de datos con una alta disponibilidad para todos los usuarios que accedan a ella. En la actualidad, el límite máximo del servidor web está en 30 conexiones simultáneas porque el alojamiento web es gratuito.
<b>RNF3.6</b>	Escalabilidad	El sistema debe ser capaz de atender 15000 peticiones de lectura y 5000 peticiones de escritura por hora a la base de datos, habría que mejorar la capacidad si el producto es un éxito en usuarios activos.

<b>RNF4</b>	<b>Requisitos Operacionales</b>	
<b>RNF4.1</b>	Entorno Físico	El sistema debe poder utilizarse por cualquier persona registrada, en cualquier lugar y en cualquier momento a través de su dispositivo móvil.
<b>RNF4.2</b>	Entorno Tecnológico	El sistema debe poder ejecutarse en la mayoría de los dispositivos con sistema operativo Android, cuyo sistema sea superior o igual a la versión 5.0 (Lollipop).
<b>RNF4.3</b>	Aplicaciones Relacionadas	El sistema debe interactuar con el protocolo HTTP de forma correcta para utilizar los servicios web RESTful de la plataforma.
<b>RNF4.4</b>	Soporte	El sistema debe proporcionar soporte para resolver los posibles problemas de los usuarios dentro de la aplicación y debe proporcionar un mantenimiento constante por parte de los administradores.
<b>RNF5</b>	<b>Requisitos de Mantenimiento y Portabilidad</b>	
<b>RNF5.1</b>	Dificultad de Mantenimiento	El mantenimiento está dividido en tres partes: mantenimiento del front-end (aplicación móvil), mantenimiento del back-end (base de datos), y mantenimiento de los servidores (realizada por la empresa que gestiona dichos servidores).
<b>RNF5.2</b>	Condiciones Especiales	El sistema operativo Android notificará al usuario cuando una nueva versión de la aplicación sea publicada, mientras que la propia aplicación avisará cuando se realicen tareas de mantenimiento.
<b>RNF5.3</b>	Portabilidad	En un principio, el sistema será exclusivo para Android y su portabilidad a iOS dependerá de las futuras extensiones al proyecto.
<b>RNF6</b>	<b>Requisitos de Seguridad</b>	
<b>RNF6.1</b>	Sistema Fiable	El sistema debe cumplir con la fiabilidad y seguridad de la protección de datos para todos los usuarios.
<b>RNF6.2</b>	Integridad de Ficheros	Los usuarios siempre tendrán la información del sistema actualizada si está conectado a Internet.
<b>RNF6.3</b>	Auditoría	En un principio no existe intención de generar dinero con el sistema, por lo que no hay necesidad de aportar los ingresos a la auditoría.
<b>RNF7</b>	<b>Requisitos Culturales y Políticos</b>	
<b>RNF7.1</b>	Factores Externos	El sistema está diseñado para niños y niñas, sin importar la raza o religión en la que crean, por lo que no existe ningún factor determinante.
<b>RNF8</b>	<b>Requisitos Legales</b>	
<b>RNF8.1</b>	Jurisdicción de Ley	La información personal de los usuarios del sistema debe protegerse de forma que se cumpla la Ley General de Protección de Datos.
<b>RNF8.2</b>	Estándar	El sistema debe cumplir con los estándares de calidad mediante un conjunto compacto de criterios de calidad y pruebas relacionadas.

*Tabla 2: Requisitos No Funcionales*

### 3.3 Requisitos de Documentación

- **Manual de Usuario**

La especificación de un manual de usuario es necesario para un correcto entendimiento de las funcionalidades del sistema y, como eje fundamental, se describirán todas las operaciones que el usuario podrá realizar sobre el sistema. Será un documento publicado en línea, con descripciones breves y usando el vocabulario adecuado.

- **Ayuda en Línea**

La aplicación proporcionará un botón de ayuda para asistir al usuario, se mostrará una pantalla con el manual de usuario y un método de contacto.

- **Guía de Instalación y Configuración**

La instalación de la aplicación estará disponible en la tienda de aplicaciones oficial de Android (Google Play Store) y se realizará de forma automática para todos los usuarios, también podrá desinstalar la aplicación sin problema alguno.

- **Empaquetado y Etiquetado**

La aplicación debe mostrar el logotipo de la aplicación Android en la pantalla de inicio (Figura 1), y además indicar que es un producto con copyright.

El desarrollador concede a Google una licencia no exclusiva y libre de derechos de autor para distribuir el producto de la forma adecuada en Google Play Console.



*Figura 1: Logotipo*

## 3.4 Descripción de Participantes y Usuarios

### 3.4.1 Participantes

Los participantes que influyen en el desarrollo del proyecto son:

- **Gerencia:** Persona que se encarga de dirigir, gestionar y administrar el proyecto para que cumpla los objetivos establecidos. Además, debe hacer cumplir las leyes, reglamentos y demás disposiciones generales establecidas.

Representante	Eduardo Guzmán De los Riscos
Tipo	Gerente del proyecto
Responsabilidades	Dirigir el proceso de desarrollo
Criterio de Éxito	Lograr los objetivos establecidos del proyecto
Entregables	Idea del proyecto

- **Desarrollador:** Persona que se encarga de iniciar el proyecto, hacer la implementación y poner a disposición de posibles actualizaciones o nuevas extensiones de contenido.

Representante	Javier Rosales Bañón
Tipo	Estudiante de Ingeniería del Software
Responsabilidades	Desarrollar la plataforma y juegos de varias asignaturas para completar su funcionalidad
Criterio de Éxito	Desplegar la plataforma cumpliendo todos los objetivos y que resulte útil en su propósito establecido
Entregables	Plataforma completada

- Otros agentes externos como los usuarios de la plataforma que ayuden a obtener nuevos requisitos del proyecto.

### 3.4.2 Usuarios

Los usuarios principales del proyecto son:

- **Alumno:** Usuario que aprende realizando juegos de las distintas asignaturas ofrecidas por la plataforma en ese momento.

Descripción	Persona que utiliza la plataforma para jugar y aprender
Tipo	Cliente
Responsabilidades	Registrarse e iniciar sesión Ver, editar y perfil propio Ver, entrar y salir de la clase Ver, buscar y realizar juegos de asignaturas Ver historial y estadísticas de sus actividades
Criterio de Éxito	La plataforma sea entretenida y fácil de utilizar

- **Profesor:** Usuario que gestiona las clases de los alumnos, puede crear nuevos juegos de cualquier asignatura y disponibles para todos los usuarios.

Descripción	Persona que utiliza la plataforma para observar y analizar
Tipo	Cliente
Responsabilidades	Registrarse e iniciar sesión Ver, editar y borrar perfil propio Ver, crear, editar, borrar, entrar y salir de la clase Ver, crear, editar, borrar, buscar y ver resultados de juegos Ver historial y estadísticas de las actividades de los alumnos
Criterio de Éxito	Los alumnos de su clase aprendan y obtengan nuevos conocimientos sobre las asignaturas

### **3.5 Alternativas y Competencia**

#### **- Duolingo**

Esta aplicación ofrece cursos completos de idiomas que se adaptan a las capacidades de los usuarios, ofreciendo pruebas iniciales para saber los conocimientos previos y situarlos en el punto correcto del curso directamente. Sin embargo, Duolingo sólo ofrece cursos de idiomas y no un temario infantil como el que quiere ofrecer Edukka, además la plataforma estará orientada a un público específico y ofrecerá más variedad de contenido.

#### **- Trivia 360**

Esta aplicación ofrece una forma interesante y competitiva de aprender conceptos de cultura general, a través de retos contra otros jugadores y haciendo un uso excelente de la gamificación. Pero la plataforma Edukka, a diferencia de Trivia 360, quiere ofrecer aprendizaje en unas asignaturas concretas, haciéndola más útil en cuanto a la enseñanza.

#### **- Khan Academy**

Esta aplicación ofrece educación gratuita para cualquier edad, desde niños de infantil hasta estudiantes de grados, a través de lecciones didácticas con vídeos, apuntes, ejemplos y por supuesto ejercicios para comprobar el aprendizaje. Pero no dispone de clases para los alumnos y profesores, es un enfoque diferenciador de Edukka, donde se fomentará las ganas de aprender, mejorar y ser el número 1.

#### **- Quizlet**

Esta aplicación ofrece herramientas sencillas para aprender sobre cualquier materia, a través de fichas educativas, juegos y más. Tiene muchas unidades de estudio diferentes creadas por usuarios que utilizan la aplicación. El principal problema es su diseño de página, la presentación es muy confusa y esto es un aspecto fundamental que vamos a dar mucha importancia en Edukka.

# CAPÍTULO 4. Diseño

## 4.1 Casos de Uso

Los casos de uso son una secuencia de interacciones que se desarrollarán entre el sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. En la Figura 2 se especifica la comunicación y el comportamiento del sistema mediante su interacción con los actores (Alumno y Profesor).

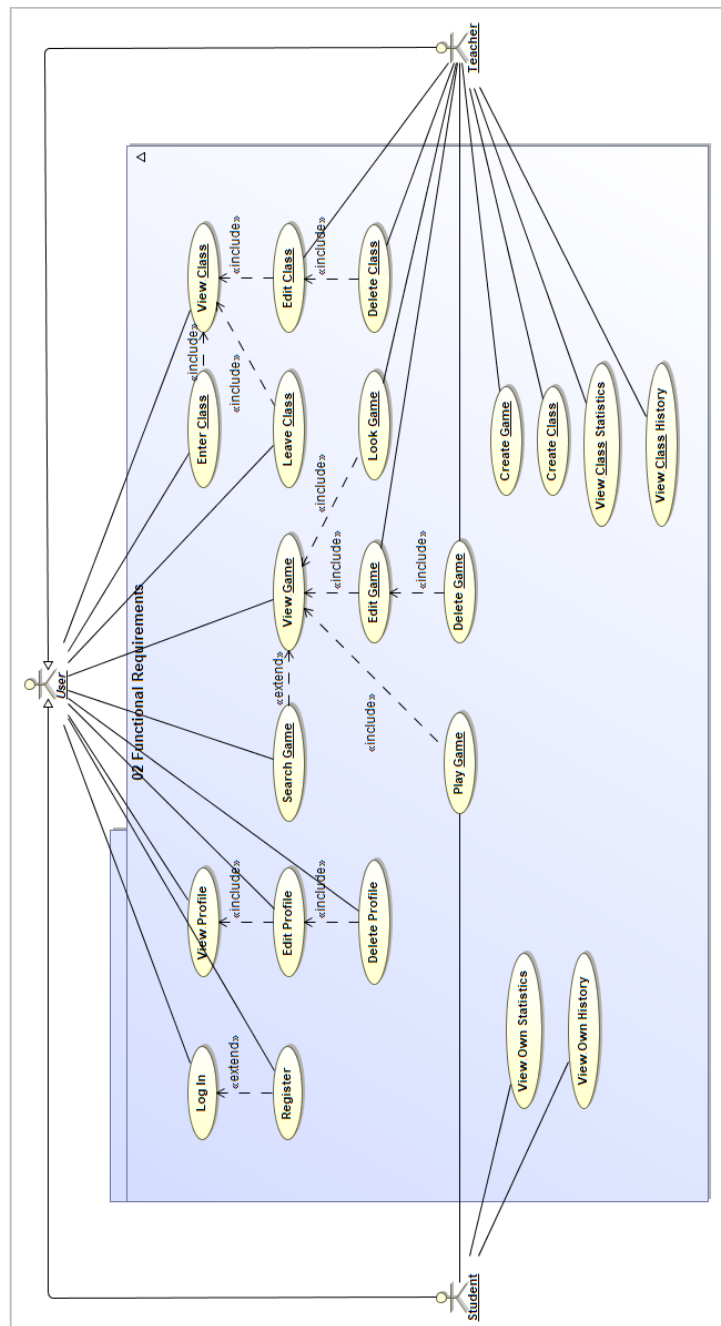


Figura 2: Diagrama de Casos de Uso

- **REGISTRAR**

<b>Caso de Uso</b>	Registrar	<b>Identificador:</b> CU-01
<b>Resumen</b>	Crear un nuevo usuario en la base de datos	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Caso de uso básico para el sistema	
<b>Precondición</b>	-	
<b>Postcondición</b>	El actor consigue registrarse correctamente en la plataforma	
<b>Descripción</b>	El nuevo usuario debe introducir sus datos personales y se insertan en la base de datos de la plataforma si los datos son válidos	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de nuevo usuario
2	Sistema	Muestra los datos a completar del nuevo usuario
3	Actor	Introducir los datos del usuario
4	Actor	Pulsar la acción de registrar
5	Sistema	Comprobar que los datos introducidos son válidos
6	Sistema	Guardar el usuario en la base de datos
7	Sistema	Acceso a la pantalla principal de la aplicación

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
5	Si los datos introducidos no son válidos aparece un mensaje de error, y el actor deberá introducir los datos de nuevo

- INICIAR SESIÓN

<b>Caso de Uso</b>	Iniciar Sesión	<b>Identificador:</b> CU-02
<b>Resumen</b>	Acceso de un usuario en la aplicación	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Registrar	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue iniciar sesión correctamente en la plataforma	
<b>Descripción</b>	Caso de uso obligatorio para poder usar la aplicación, el usuario debe introducir sus datos personales que deben coincidir con los datos guardados en la base de datos	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Introducir apodo y contraseña del usuario
2	Actor	Pulsar la acción de iniciar sesión
3	Sistema	Comprobar que los datos introducidos están registrados en la base de datos del sistema
4	Sistema	Acceso a la pantalla principal de la aplicación

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
3	Si los datos introducidos no son correctos aparece un mensaje indicando el error, y el actor deberá introducir los datos de nuevo

- **VER PERFIL**

<b>Caso de Uso</b>	Ver Perfil	<b>Identificador:</b> CU-03
<b>Resumen</b>	Ver los datos personales del usuario	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue ver su propio perfil	
<b>Descripción</b>	Se muestran todos los datos almacenados en la base de datos del actor, estos datos son públicos para los usuarios de la clase	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi perfil
2	Sistema	Muestra la información relacionada al usuario

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
-	-

- **EDITAR PERFIL**

<b>Caso de Uso</b>	Editar Perfil	<b>Identificador:</b> CU-04
<b>Resumen</b>	Editar los datos personales del usuario	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Perfil	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue actualizar los datos personales de su perfil	
<b>Descripción</b>	Se permiten realizar modificaciones a los datos personales del actor, que se guardarán en la base de datos si los datos son válidos y actualizando la información almacenada	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi perfil
2	Sistema	Muestra la información relacionada al usuario
3	Actor	Pulsar la acción de editar mi perfil
4	Sistema	Muestra los datos a modificar del usuario
5	Actor	Introducir los nuevos datos del usuario
6	Actor	Pulsar la acción de guardar
7	Sistema	Comprobar que los datos introducidos son válidos
8	Sistema	Actualizar los valores modificados del usuario

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
6	El actor también puede cancelar la edición de la información de su perfil, y el sistema vuelve a mostrar la sección de mi perfil
7	Si los valores actualizados no son válidos, el sistema debe cancelar la edición del perfil y mostrará un mensaje de error

- **BORRAR PERFIL**

<b>Caso de Uso</b>	Borrar Perfil	<b>Identificador:</b> CU-05
<b>Resumen</b>	Borrar los datos personales del usuario	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Editar Perfil	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue borrar su perfil y actividad de la base de datos	
<b>Descripción</b>	Se puede eliminar el usuario de la base de datos, también se borra la actividad si es alumno o la clase si es profesor	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi perfil
2	Sistema	Muestra la información relacionada al usuario
3	Actor	Pulsar la acción de editar mi perfil
4	Sistema	Muestra los datos a modificar del usuario
5	Actor	Pulsar la acción de borrar mi perfil
6	Sistema	Muestra la pregunta de confirmación
7	Actor	Pulsar la acción de aceptar
8	Sistema	Borrar usuario y salir de la aplicación

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
7	El actor también puede cancelar el borrado de su perfil, y el sistema vuelve a mostrar la sección de editar mi perfil

- VER CLASE

<b>Caso de Uso</b>	Ver Clase	<b>Identificador:</b> CU-06
<b>Resumen</b>	Ver la clase a la que pertenece el usuario	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue ver su propia clase	
<b>Descripción</b>	Se muestran la información y los usuarios que forman parte de la clase del actor, existe la posibilidad de ver los perfiles de los usuarios de la clase	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi clase
2	Sistema	Muestra la información relacionada a la clase del usuario, puede ser pública o privada

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
-	-

- **CREAR CLASE**

<b>Caso de Uso</b>	Crear Clase	<b>Identificador:</b> CU-07
<b>Resumen</b>	Crear una nueva clase en la base de datos	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado y no debe tener una clase creada	
<b>Postcondición</b>	El actor consigue crear su clase privada en la plataforma	
<b>Descripción</b>	Se creará una nueva clase que va a ser creada por un profesor y estará disponible para que los alumnos entren a dicha clase	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi clase
2	Actor	Pulsar la acción de crear clase
3	Sistema	Muestra los datos a completar de la nueva clase
4	Actor	Introducir los datos de la clase
5	Actor	Pulsar la acción de crear
6	Sistema	Comprobar que los datos introducidos son válidos
7	Sistema	Guardar la nueva clase en la base de datos

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
2	Si el actor ya tiene una clase creada, no puede crear otra y lo que puede hacer es editar la clase creada
6	Si los datos introducidos no son válidos, el sistema debe cancelar la creación de la clase y mostrar el mensaje de error

- **EDITAR CLASE**

<b>Caso de Uso</b>	Editar Clase	<b>Identificador:</b> CU-08
<b>Resumen</b>	Editar la clase creada por el actor	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Clase	
<b>Precondición</b>	El actor debe estar registrado y debe tener una clase creada	
<b>Postcondición</b>	El actor consigue editar la información de su propia clase	
<b>Descripción</b>	Se pueden realizar modificaciones a los datos de la clase, que se guardarán en la base de datos y actualizando la información para los alumnos de dicha clase	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi clase
2	Sistema	Muestra la información relacionada a la clase
3	Actor	Pulsar la acción de editar clase
4	Sistema	Muestra los datos a modificar de la clase
5	Actor	Introducir los nuevos datos de la clase
6	Actor	Pulsar la acción de guardar
7	Sistema	Comprobar que los datos introducidos son válidos
8	Sistema	Actualizar la clase en la base de datos

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
6	El actor también puede cancelar la edición de la información de la clase, y el sistema vuelve a mostrar la sección de mi clase
7	Si los datos introducidos no son válidos, el sistema debe cancelar la edición de la clase y mostrar el mensaje de error

- **BORRAR CLASE**

<b>Caso de Uso</b>	Borrar Clase	<b>Identificador:</b> CU-09
<b>Resumen</b>	Borrar la clase creada por el actor	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Editar Clase	
<b>Precondición</b>	El actor debe estar registrado y debe tener una clase creada	
<b>Postcondición</b>	El actor consigue borrar su clase de la base de datos	
<b>Descripción</b>	Se puede eliminar la clase creada por el actor de la base de datos, los usuarios de la clase se moverán a la clase pública	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi clase
2	Sistema	Muestra la información relacionada a la clase
3	Actor	Pulsar la acción de editar mi clase
4	Sistema	Muestra los datos a modificar de la clase
5	Actor	Pulsar la acción de borrar mi clase
6	Sistema	Muestra la pregunta de confirmación
7	Actor	Pulsar la acción de aceptar
8	Sistema	Borrar clase y volver al inicio de la aplicación

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
7	El actor también puede cancelar el borrado de su clase, y el sistema vuelve a mostrar la sección de editar mi clase

- **ENTRAR CLASE**

<b>Caso de Uso</b>	Entrar Clase	<b>Identificador:</b> CU-10
<b>Resumen</b>	Entrar en una clase privada	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Clase	
<b>Precondición</b>	El actor debe estar registrado y debe conocer el identificador de la clase privada a la que desea entrar	
<b>Postcondición</b>	El actor consigue actualizar su clase	
<b>Descripción</b>	Se permite modificar la clase a la que pertenece el actor, entrar en una clase privada implica la salida de la clase pública	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi clase
2	Sistema	Muestra la información relacionada a la clase pública
3	Actor	Pulsar la acción de entrar en clase
4	Sistema	Muestra la ventana de entrada a la clase
5	Actor	Introducir el identificador de la clase privada
6	Actor	Pulsar la acción de aceptar
7	Sistema	Comprobar que el identificador es válido
8	Sistema	Actualizar la clase del usuario

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
3	La acción no estará visible si el actor es el creador de la clase actual o si el actor ya forma parte de una clase privada
6	El actor también puede cancelar la entrada a la clase, y el sistema vuelve a mostrar la sección de mi clase
7	Si el identificador introducido no está relacionado con ninguna clase existente en la base de datos, el sistema mostrará un mensaje de error

- **SALIR CLASE**

<b>Caso de Uso</b>	Salir Clase	<b>Identificador:</b> CU-11
<b>Resumen</b>	Salir de una clase privada	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Clase	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue actualizar su clase	
<b>Descripción</b>	Se permite modificar la clase a la que pertenece el actor, entrar en la clase pública implica la salida de la clase privada	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de mi clase
2	Sistema	Muestra la información relacionada a la clase pública
3	Actor	Pulsar la acción de salir de clase
4	Sistema	Muestra la ventana de salida de la clase privada
5	Actor	Pulsar la acción de aceptar
6	Sistema	Actualizar la clase del usuario

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
3	La acción no estará visible si el actor es el creador de la clase actual o si el actor ya forma parte de la clase pública
5	El actor también puede cancelar la salida de la clase, y el sistema vuelve a mostrar la sección de mi clase

- **VER JUEGO**

<b>Caso de Uso</b>	Ver Juego	<b>Identificador:</b> CU-12
<b>Resumen</b>	Ver la información del juego seleccionado	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue ver el juego seleccionado	
<b>Descripción</b>	Se muestra la información del juego de una asignatura, estos datos son públicos para todos los usuarios	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar el juego que desea ver
6	Sistema	Muestra la información del juego

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
4	Si la base de datos no tiene juegos de la asignatura seleccionada, muestra un mensaje de error

## - CREAR JUEGO

<b>Caso de Uso</b>	Crear Juego	<b>Identificador:</b> CU-13
<b>Resumen</b>	Crear un nuevo juego de una asignatura en la base de datos	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue crear un nuevo juego en la plataforma	
<b>Descripción</b>	Se creará un nuevo juego con sus pruebas para una asignatura, y todos los usuarios podrán acceder a dicho juego	

### Curso Normal

Nro.	Ejecutor	Paso o Actividad
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar la acción de crear juego
6	Sistema	Muestra los datos a completar del nuevo juego
7	Actor	Introducir los datos del juego
8	Actor	Introducir las pruebas quiz del juego de uno en uno
9	Actor	Pulsar la acción de crear
10	Sistema	Guardar el juego y las pruebas en la base de datos

### Cursos Alternos

Nro.	Descripción de acciones alternas
9	Si los datos introducidos no son válidos, el sistema debe cancelar la creación del juego y mostrar el mensaje de error

- **EDITAR JUEGO**

<b>Caso de Uso</b>	Editar Juego	<b>Identificador:</b> CU-14
<b>Resumen</b>	Editar el juego de una asignatura	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Juego	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue editar la información del juego	
<b>Descripción</b>	Se pueden realizar modificaciones a los datos del juego, que se guardarán en la base de datos si los datos son válidos y actualizando la información para todos los usuarios	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar el juego que desea editar
6	Sistema	Muestra la información del juego
7	Actor	Pulsar la acción de editar juego
8	Sistema	Muestra los datos a modificar del juego
9	Actor	Introducir los nuevos datos del juego
10	Actor	Pulsar la acción de guardar
11	Sistema	Actualizar el juego en la base de datos

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
4	Si la base de datos no tiene juegos de la asignatura seleccionada, muestra un mensaje de error
10	El actor también puede cancelar la edición de la información del juego, y el sistema vuelve a mostrar el juego
11	Si los datos introducidos no son válidos, el sistema debe cancelar la edición del juego y mostrar el mensaje de error

## - BORRAR JUEGO

<b>Caso de Uso</b>	Borrar Juego	<b>Identificador:</b> CU-15
<b>Resumen</b>	Borrar el juego de una asignatura	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Editar Juego	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue borrar el juego de la base de datos	
<b>Descripción</b>	Se puede eliminar un juego de la base de datos, también se borrarán las preguntas que componen dicho juego	

### Curso Normal

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar el juego que desea borrar
6	Sistema	Muestra la información del juego
7	Actor	Pulsar la acción de editar juego
8	Sistema	Muestra los datos a modificar del juego
9	Actor	Pulsar la acción de borrar juego
10	Sistema	Muestra la pregunta de confirmación
11	Actor	Pulsar la acción de aceptar
12	Sistema	Borrar juego y volver al inicio de la aplicación

### Cursos Alternos

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
4	Si la base de datos no tiene juegos de la asignatura seleccionada, muestra un mensaje de error
11	El actor también puede cancelar el borrado del juego, y el sistema vuelve a mostrar la sección de editar el juego

- **BUSCAR JUEGO**

<b>Caso de Uso</b>	Buscar Juego	<b>Identificador:</b> CU-16
<b>Resumen</b>	Buscar juegos específicos de una asignatura	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Secundario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue una lista de juegos buscados	
<b>Descripción</b>	Un filtrado de juegos sobre una asignatura, el cual permite a los usuarios buscar el juego concreto que desea realizar	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego a buscar
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar la acción de buscar
6	Actor	Introducir el nombre del juego que desea buscar
7	Sistema	Muestra los juegos resultados de la búsqueda

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
4	Si la base de datos no tiene juegos de la asignatura seleccionada, muestra un mensaje de error
7	Si la base de datos no encuentra juegos con la búsqueda realizada, muestra un mensaje de error

- **OBSERVAR JUEGO**

<b>Caso de Uso</b>	Observar Juego	<b>Identificador:</b> CU-17
<b>Resumen</b>	Mostrar las soluciones del juego de una asignatura	
<b>Actores</b>	Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Juego	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue ver las soluciones del juego seleccionado	
<b>Descripción</b>	Se muestran las respuestas correctas de las preguntas del juego, es una herramienta útil para el profesor	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego a observar
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar el juego de la asignatura que desea observar
6	Sistema	Muestra la información del juego
7	Actor	Pulsar la acción de jugar
8	Sistema	Muestra las respuestas correctas del juego

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
4	Si la base de datos no tiene juegos de la asignatura seleccionada, muestra un mensaje de error

## - REALIZAR JUEGO

<b>Caso de Uso</b>	Realizar Juego	<b>Identificador:</b> CU-18
<b>Resumen</b>	Realizar un juego con pruebas quiz de una asignatura	
<b>Actores</b>	Alumno	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Ver Juego	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue realizar las pruebas del juego seleccionado	
<b>Descripción</b>	El caso de uso más importante para la plataforma porque los juegos es el elemento principal para que los alumnos aprendan, los juegos consisten en pruebas académicas de tipo quiz	

### Curso Normal

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de juegos
2	Sistema	Muestra la lista con todas las asignaturas
3	Actor	Pulsar la asignatura del juego a realizar
4	Sistema	Muestra la lista con los juegos de la asignatura
5	Actor	Pulsar el juego de la asignatura que desea realizar
6	Sistema	Muestra la información del juego
7	Actor	Pulsar la acción de jugar
8	Sistema	Muestra las preguntas del juego de uno en uno
9	Actor	Seleccionar la respuesta que considera correcta
10	Sistema	Al acabar, muestra los resultados del juego
11	Sistema	Guardar la actividad en la base de datos

### Cursos Alternos

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
4	Si la base de datos no tiene juegos de la asignatura seleccionada, muestra un mensaje de error
9	Si el juego seleccionado termina de mostrar todas las preguntas, el sistema muestra los resultados obtenidos al usuario

- **VER HISTORIAL**

<b>Caso de Uso</b>	Ver Historial	<b>Identificador:</b> CU-19
<b>Resumen</b>	Ver el historial de los alumnos de la clase	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue ver el historial propio o de los alumnos	
<b>Descripción</b>	Se muestran todos los juegos completados de cada alumno o de los alumnos que forman parte de la clase creada por el actor, es útil para analizar las calificaciones obtenidas	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de actividad
2	Actor	Pulsar la acción de historial
3	Sistema	Muestra la información relacionada al historial del alumno o de los alumnos de la clase

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
3	Si no existe actividad del alumno o de los alumnos, el sistema mostrará un mensaje comentando que no hay historial disponible

- **VER ESTADÍSTICAS**

<b>Caso de Uso</b>	Ver Estadísticas	<b>Identificador:</b> CU-20
<b>Resumen</b>	Ver las estadísticas de los alumnos de la clase	
<b>Actores</b>	Alumno y Profesor	
<b>Tipo</b>	Primario	
<b>Referencias</b>	Dependencia con el caso de uso Iniciar Sesión	
<b>Precondición</b>	El actor debe estar registrado previamente	
<b>Postcondición</b>	El actor consigue ver las estadísticas propias o de los alumnos	
<b>Descripción</b>	Se muestran todas las estadísticas de cada alumno o de los alumnos que forman parte de la clase creada por el actor, sirve para observar los resultados generales	

**Curso Normal**

<b>Nro.</b>	<b>Ejecutor</b>	<b>Paso o Actividad</b>
1	Actor	Pulsar la sección de actividad
2	Actor	Pulsar la acción de estadísticas
3	Sistema	Muestra la información relacionada a la estadística del alumno o de los alumnos de la clase

**Cursos Alternos**

<b>Nro.</b>	<b>Descripción de acciones alternas</b>
3	Si no existe actividad del alumno o de los alumnos, el sistema mostrará un mensaje comentando que no hay estadísticas disponibles

## 4.2 Modelo de Datos

En la Figura 3 se describe la estructura del sistema mostrando las clases, atributos, enumeraciones y relaciones entre los objetos. Esta estructura no es utilizada en el servidor, sino que sirve para ver mejor el diseño de la plataforma.

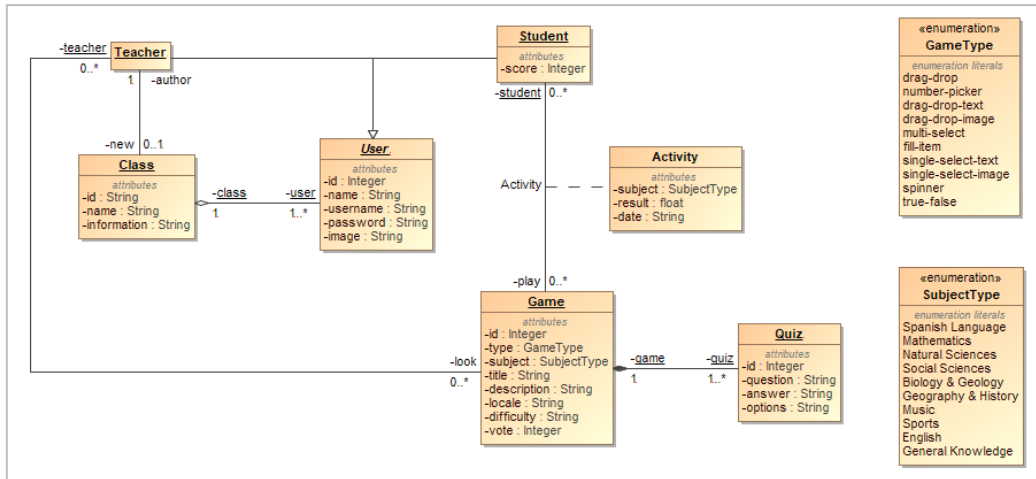


Figura 3: Diagrama de Clases

Los cambios aplicados en el servidor (Figura 4) son que no hay tipos enumerados y las subclases del usuario es sustituido por el atributo “role” para una mejor implementación.

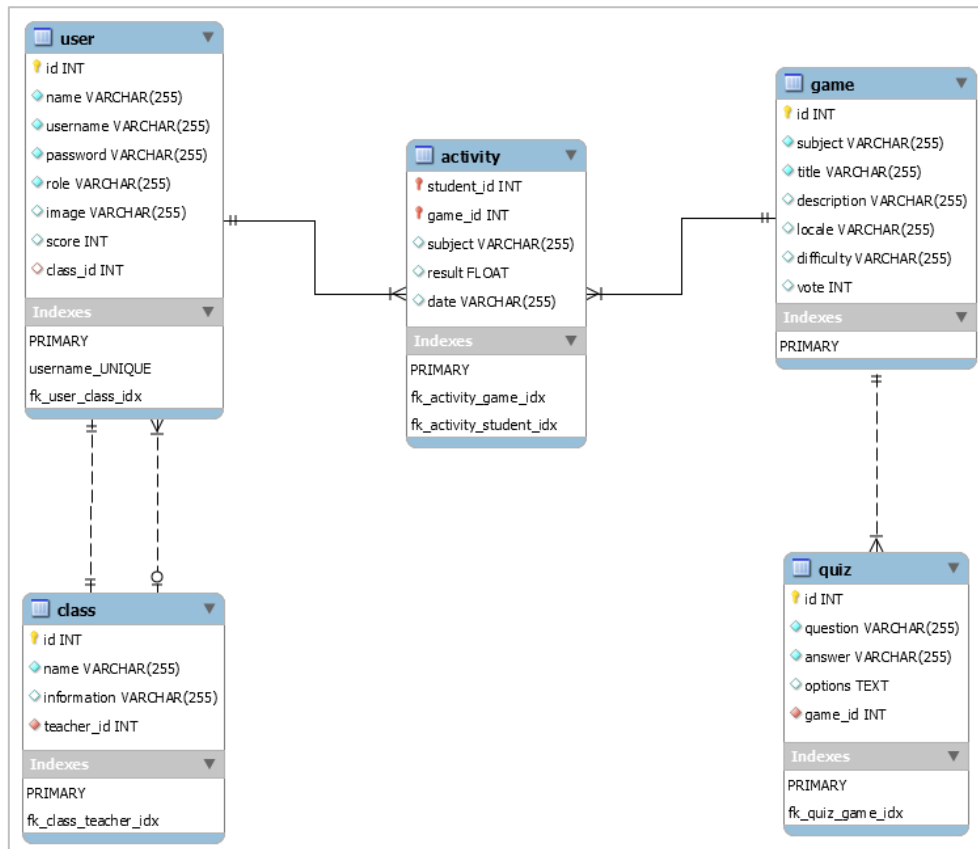


Figura 4: Diagrama de la Base de Datos

### - Usuario

Actor que tiene acceso a las funcionalidades de la aplicación según su rol y forma parte de una clase. Los profesores siempre tienen puntuación 0 y un único avatar.

Atributo	Descripción	Tipo
id	Identificador único del usuario	Integer
name	Nombre y apellidos del usuario	String
username	Apodo único personal del usuario	String
password	Contraseña personal del usuario	String
role	Rol elegido por el usuario	String
image	Avatar elegido por el usuario	String
score	Puntuación personal obtenida por el usuario	Integer
class_id	Identificador de la clase a la que pertenece	Integer

### - Clase

Grupo de usuarios que es creado por un profesor, los usuarios pueden entrar y salir de la clase en cualquier momento, además existe una clase pública por defecto para los usuarios que no tienen una clase seleccionada.

Atributo	Descripción	Tipo
id	Identificador único de la clase	Integer
name	Nombre de la clase	String
information	Información detallada de la clase	String
teacher_id	Identificador del profesor creador	Integer

### - Juego

Elemento fundamental de la plataforma, cada usuario solo podrá ver los juegos con el idioma que tenga su dispositivo móvil, la dificultad afecta a la puntuación obtenida para el usuario (fácil +10, normal +20 y difícil +30) si se aprueba el juego.

Atributo	Descripción	Tipo
id	Identificador único del juego	Integer
subject	Nombre de la asignatura a la que pertenece	String
title	Nombre del juego	String
description	Información detallada del juego	String
locale	Idioma elegido para el juego	String
difficulty	Dificultad elegida para el juego	String
vote	Votos obtenidos de los alumnos	Integer

- **Quiz**

Elemento que forma parte de un único juego, todos tienen una pregunta y su respuesta.

<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
id	Identificador único del quiz	Integer
question	Pregunta del quiz	String
answer	Respuesta correcta del quiz	String
options	Opciones disponibles del quiz	String
game_id	Identificador del juego al que pertenece	Integer

- **Actividad**

Elemento sobre los juegos completados por los alumnos, se almacena el resultado y la fecha de la última vez.

<b>Atributo</b>	<b>Descripción</b>	<b>Tipo</b>
student_id	Identificador del alumno	Integer
game_id	Identificador del juego	Integer
subject	Nombre de la asignatura del juego	String
result	Resultado obtenido por el alumno	Float
date	Fecha cuando se realizó el juego	String

### 4.3 Navegación

En la Figura 5 se modela el comportamiento de la aplicación Android, especificando la secuencia de eventos que el objeto atraviesa durante su tiempo de vida en respuesta a los eventos. El estado inicial es “LoginDialog”, el estado principal de la aplicación es “MainBrowser” y no existe un estado final.

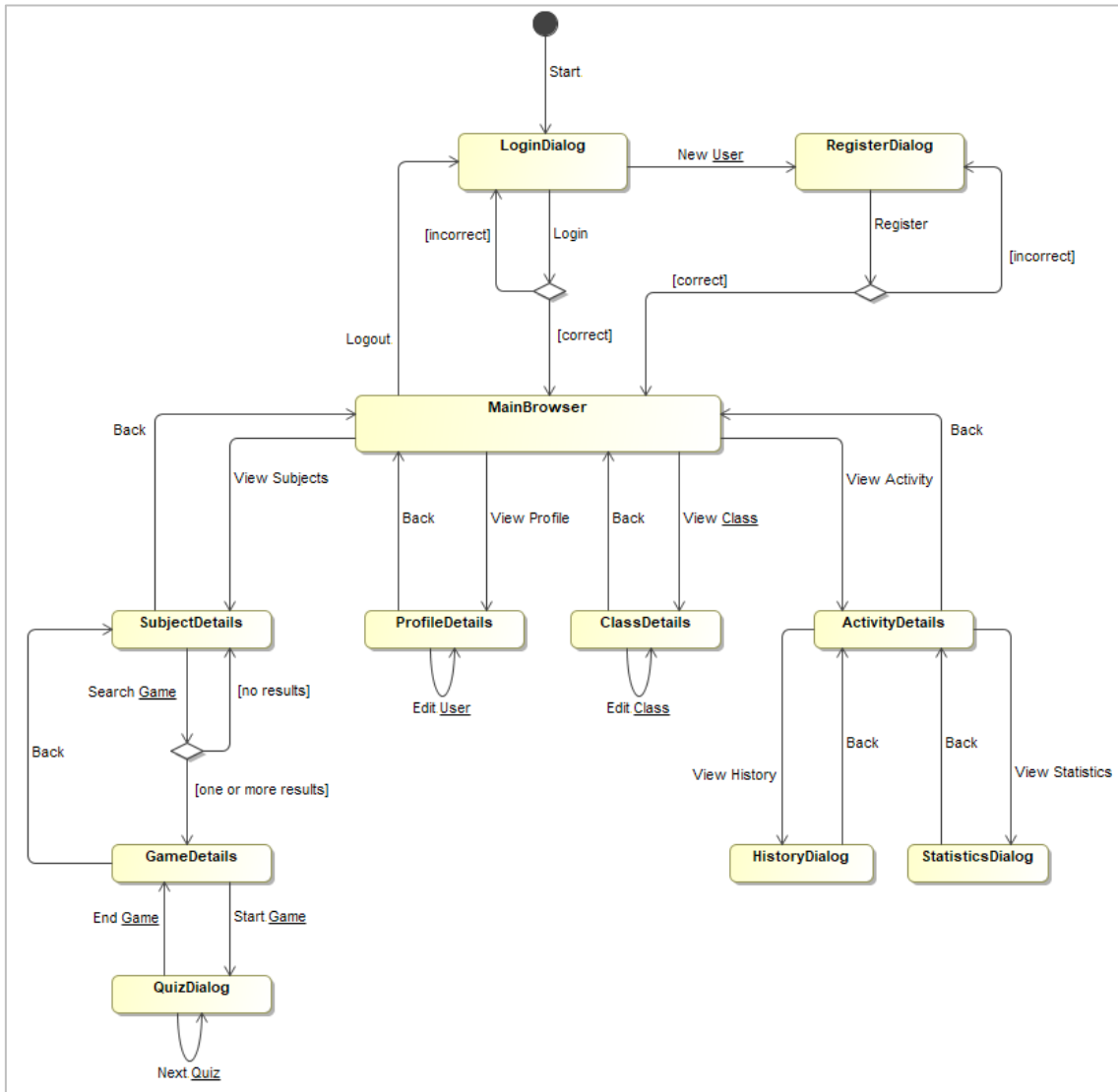


Figura 5: Diagrama de Máquinas de Estado

El diagrama se compone de 11 estados, que son:

- **LoginDialog**: Pantalla inicial donde se muestra el inicio de sesión del usuario.
- **RegisterDialog**: Pantalla donde se registra un nuevo usuario.
- **MainBrowser**: Pantalla principal que accede el usuario tras iniciar sesión.
- **SubjectDetails**: Pantalla donde se muestra los juegos disponibles para una asignatura.
- **GameDetails**: Pantalla donde se muestra la información del juego.
- **QuizDialog**: Pantalla donde se muestra la pregunta y las posibles opciones como respuesta, cada nuevo quiz del juego se actualiza la pantalla.
- **ProfileDetails**: Pantalla donde se muestra la información del usuario.
- **ClassDetails**: Pantalla donde se muestra la información de la clase.
- **ActivityDetails**: Pantalla donde se muestra la actividad disponible para el usuario, que se divide en el historial y las estadísticas.
- **HistoryDialog**: Pantalla donde se muestra el historial del usuario o de la clase.
- **StatisticsDialog**: Pantalla donde se muestra las estadísticas del usuario o de la clase.

## 4.4 Servicios y Componentes

En la Figura 6 se modela la interfaz de programación de aplicaciones (API) de la plataforma, con un conjunto de subrutinas, funciones y procedimientos para ser utilizado por la aplicación Android como una capa de abstracción. La API está dividida en cuatro servicios (Usuario, Clase, Juego y Quiz) según su funcionalidad y que será explicado con mayor detalle en la parte de implementación de la aplicación Web.

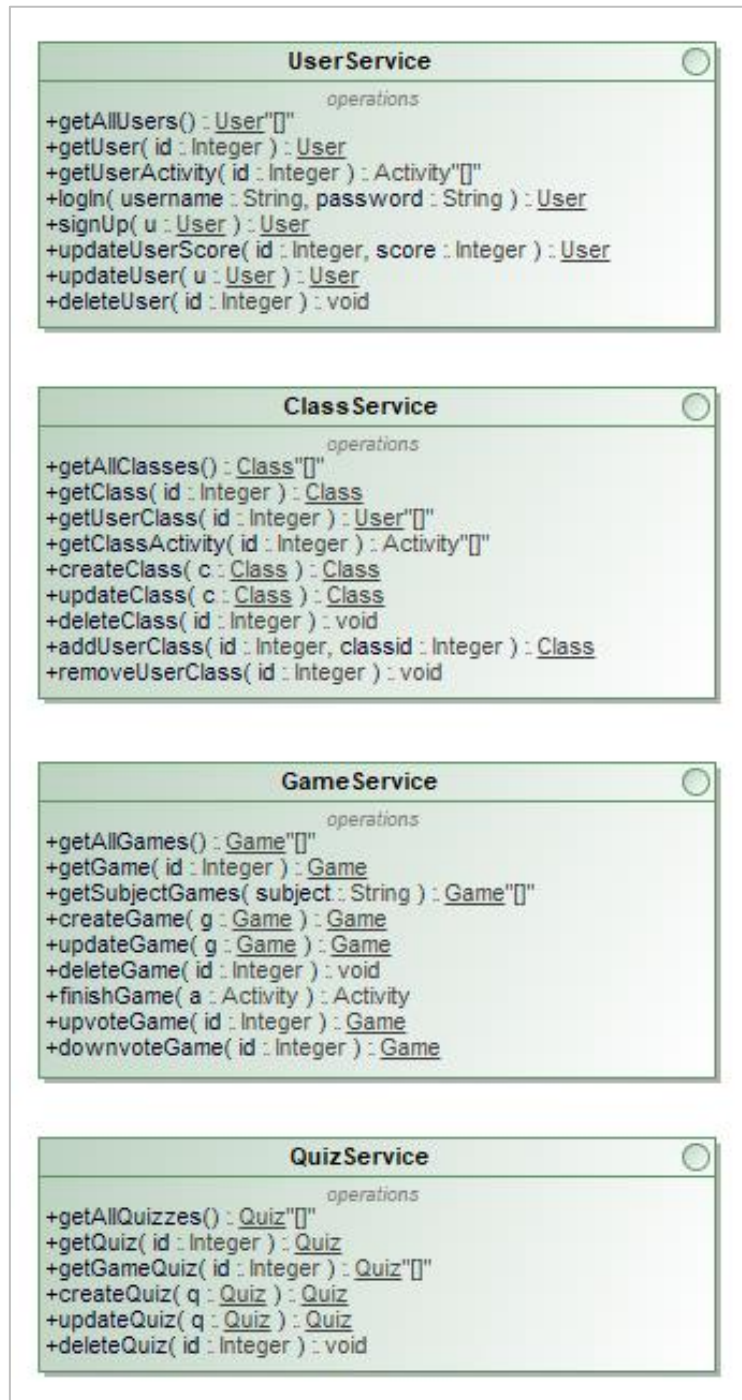
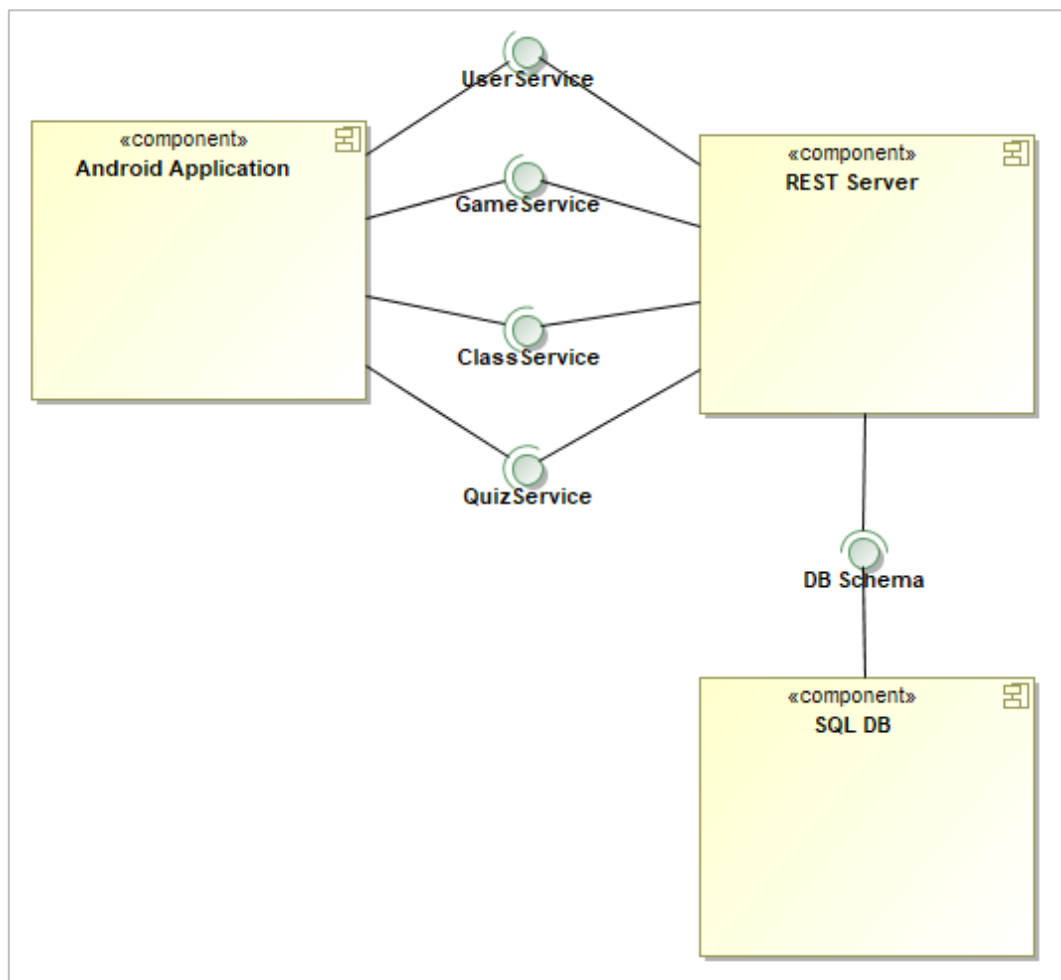


Figura 6: Diagrama de la Interfaz

En la Figura 7 se modela los componentes de la plataforma y muestra las dependencias entre estos componentes a través de los servicios.

El componente Android se conecta con el componente Web a través de los servicios implementados, y también el componente Web se conecta con la base de datos a través de una interfaz SQL.



*Figura 7: Diagrama de Componentes*

## 4.5 Arquitectura

En la Figura 8 se modela la arquitectura en tiempo de ejecución de la plataforma, que describe el despliegue físico de información generada por el programa de software en los componentes de hardware y muestra las relaciones de los artefactos con esos nodos.

Hay dos dispositivos principales, el dispositivo móvil con la aplicación Android instalada y el servidor con la aplicación Web instalada y conectada a la base de datos MySQL, la comunicación entre ellos se realiza mediante el protocolo HTTP.

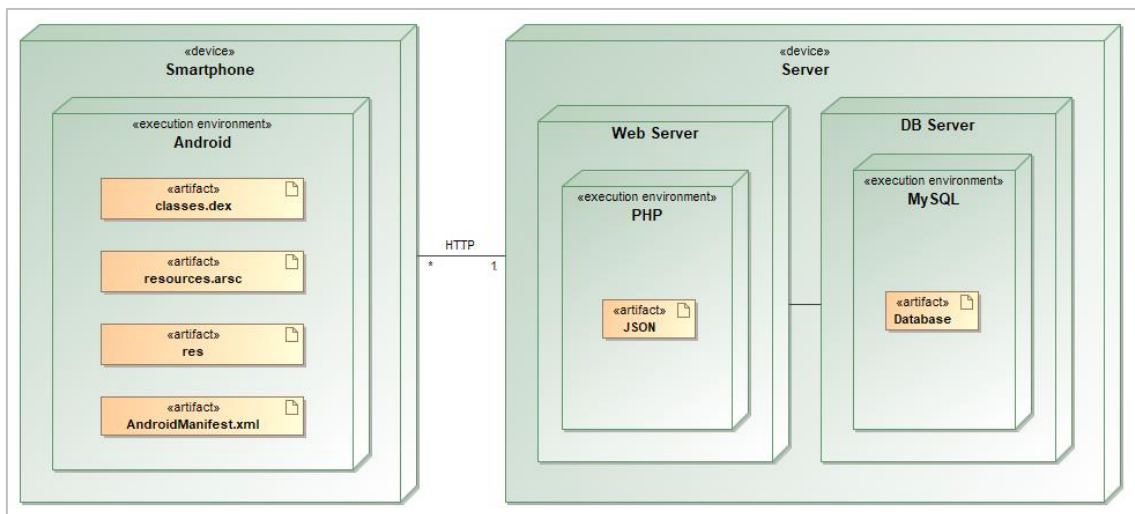


Figura 8: Diagrama de Despliegue



## CAPÍTULO 5. Implementación

### 5.1 Desarrollo del Back-end

La aplicación Web está compuesta por cinco ficheros PHP, cuatro de ellos corresponden a los servicios ya comentados y el fichero “index” que es el más importante. Además, como podemos ver en la Figura 9, incluye la carpeta del framework Slim, varios ficheros git y un fichero “.htaccess” que es un tipo de archivo especial que se utiliza dentro de los alojamientos web que funcionan con servidores Apache y permite modificar diferentes variables en la configuración de nuestra cuenta de alojamiento en el servidor.

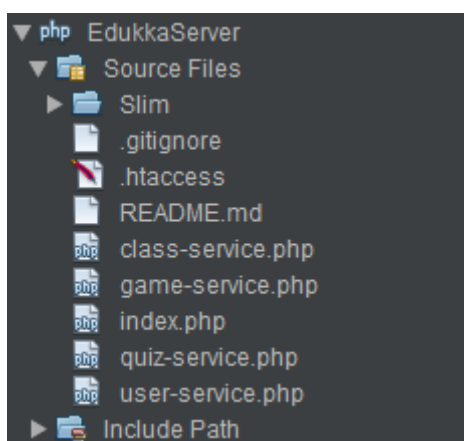


Figura 9: Estructura del back-end

En el fragmento de código, Código 1, se muestra parte del fichero “index”, cuya funcionalidad es inicializar la aplicación y la base de datos. Señalar que algunas líneas del código varían si el servidor es local o remoto. Las primeras líneas consisten en importar el framework Slim y los servicios que va a utilizar la aplicación, después hay que crear una instancia de Slim e indicar que la salida de las funciones es en formato JSON; por último, se especifica todas las funciones y ejecutamos la instancia. Además, la función “getDB” devuelve la conexión a la base de datos utilizando la extensión PDO.

En el fragmento de código, Código 2, se muestra la implementación general de todas las funciones, cada una con su particularidad, y como ejemplo vemos la función para obtener los datos de un usuario. El proceso es siempre el mismo: obtener la conexión a la base de datos, ejecutar la sentencia SQL, obtener la respuesta, comprobar valores nulos y producir la salida de la función. Todo está dentro de un try-catch para detectar errores en el PDO.

```

require 'C:\wamp64\www\edukka\Slim\Slim.php';
//require '../Slim/Slim.php';

require_once 'user-service.php';
require_once 'class-service.php';
require_once 'game-service.php';
require_once 'quiz-service.php';
\Slim\Slim::registerAutoloader();

$app = new \Slim\Slim();
$app->contentType('application/json');

//...
$app->get('/user/:id', 'getUser');
//...

$app->run();

function getDB() {
    $dbhost = 'localhost';
    $dbuser = 'root';           //$dbuser = 'id5255892_root';
    $dbpass = '';              //$dbpass = '';
    $dbname = 'edukka';        //$dbname = 'id5255892_edukka';

    $mysql_conn_string = "mysql:host=$dbhost;
                           dbname=$dbname;
                           charset=utf8mb4";
    $dbConnection = new PDO($mysql_conn_string, $dbuser, $dbpass);
    $dbConnection->setAttribute(PDO::ATTR_ERRMODE,
                                PDO::ERRMODE_EXCEPTION);
    return $dbConnection;
}

```

*Código 1: Fichero index*

```

function getUser($id) {
    $sql = 'SELECT * FROM user WHERE id = ?';
    try {
        $db = getDB();
        $stmt = $db->prepare($sql);
        $stmt->bindValue(1, $id);
        $stmt->execute();
        $user = $stmt->fetchObject();
        if ($user === false) {
            $user = ['id'=>null];
        }
        $db = null;
        echo json_encode($user);
    } catch(PDOException $e) {
        echo json_encode($e->getMessage());
    }
}

```

*Código 2: Función getUser*

La aplicación tiene 33 funciones en total, en la Tabla 3 aparecen información más detallada de cada una, pero hay varias que no se utilizan debido a que solo sirven para el desarrollador y no tienen funcionalidad dentro de la aplicación Android (por ejemplo, obtener todos los usuarios de la plataforma). También hay funciones con funcionalidad que es obligatorio ejecutarla en la aplicación Web, las cuales son crear juegos y preguntas.

<b>Método</b>	<b>URL</b>	<b>Descripción</b>
GET	/	Página de introducción y ayuda
GET	/users	Datos de todos los usuarios
GET	/user/:id	Datos de un usuario
GET	/user/activity/:id	Actividad de un usuario
POST	/login	Comprobar datos sobre un usuario
POST	/signup	Crear nuevo usuario
POST	/user/score	Actualizar puntuación de un usuario
POST	/user/edit	Editar datos de un usuario
POST	/user/delete	Borrar un usuario
GET	/classes	Datos de todas las clases
GET	/class/:id	Datos de una clase
GET	/myclass/:id	Datos de los usuarios de una clase
GET	/class/activity/:id	Actividad de los usuarios de una clase
POST	/class/new	Crear nueva clase
POST	/class/edit	Editar datos de una clase
POST	/class/delete	Borrar una clase
POST	/class/adduser	Agregar usuario a una clase
POST	/class/remuser	Eliminar usuario de una clase
GET	/games	Datos de todos los juegos
GET	/game/:id	Datos de un juego
GET	/games/:sub	Datos de los juegos de una asignatura
POST	/game/new	Crear nuevo juego
POST	/game/edit	Editar datos de un juego
POST	/game/delete	Borrar un juego
POST	/game/finish	Crear actividad de un juego y usuario
POST	/game/upvote	Actualizar votos de un juego +1
POST	/game/downvote	Actualizar votos de un juego -1
GET	/quizzes	Datos de todas las preguntas
GET	/quiz/:id	Datos de una pregunta
GET	/play/:id	Datos de las preguntas de un juego
POST	/quiz/new	Crear nueva pregunta
POST	/quiz/edit	Editar datos de una pregunta
POST	/quiz/delete	Borrar una pregunta

Tabla 3: Funciones

## 5.2 Desarrollo del Front-end

La aplicación Android implementa el patrón Modelo-Vista-Controlador (MVC), que separa los datos y la lógica de negocio de la aplicación de su representación y el módulo encargado de gestionar los eventos y las comunicaciones. Para ello, se basa en las ideas de reutilización de código y la separación de conceptos, características que buscan facilitar la tarea de desarrollo de aplicaciones y su posterior mantenimiento.

Además, se aplica el patrón Singleton para restringir la creación de objetos pertenecientes a una clase o el valor de un tipo a un único objeto, su intención consiste en garantizar que una clase solo tenga una instancia y proporcionar un punto de acceso global a ella. Y también para las listas se utiliza el patrón ViewHolder, con el que podemos conseguir un procesamiento un 15% más rápido de las listas mejorando la fluidez y, por consiguiente, la experiencia de usuario.

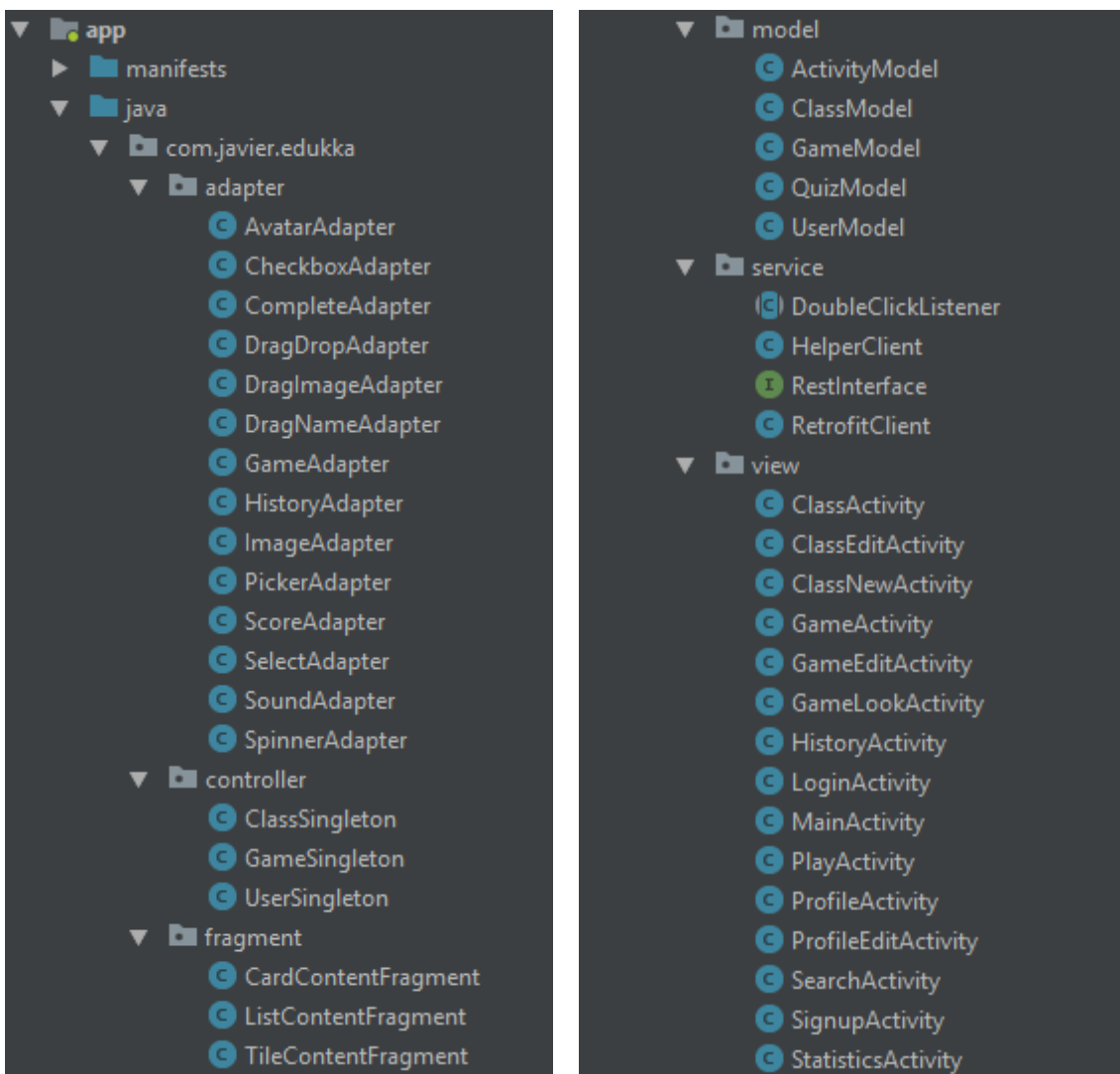


Figura 10: Estructura del front-end

Como podemos ver en la Figura 10, la aplicación se compone de seis carpetas, que son:

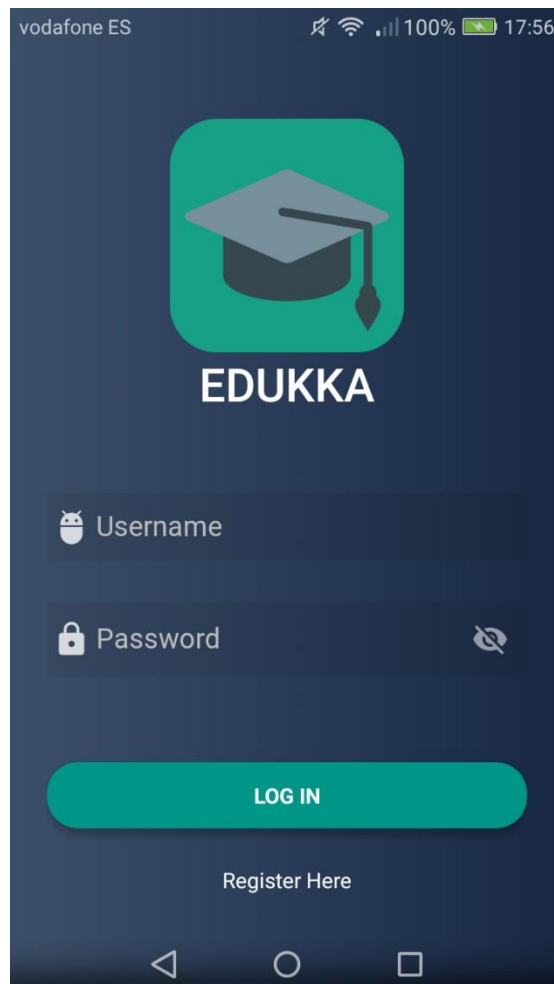
- **Adapter:** Adaptadores personalizados para las listas RecyclerView.
  - AvatarAdapter: Adaptador para elegir la imagen personal del usuario.
  - CheckboxAdapter: Adaptador para los juegos con casillas de verificación y es utilizado en la asignatura “Biología y Geología”.
  - CompleteAdapter: Adaptador para los juegos con campos de texto y es utilizado en la asignatura “Geografía e Historia”.
  - DragDropAdapter: Adaptador para los juegos que ordenan palabras y es utilizado en la asignatura “Lengua Castellana”.
  - DragImageAdapter: Adaptador para los juegos que arrastran imágenes y es utilizado en la asignatura “Ciencias Sociales”.
  - DragNameAdapter: Adaptador para los juegos que arrastran palabras y es utilizado en la asignatura “Ciencias Naturales”.
  - GameAdapter: Adaptador para mostrar los juegos de una asignatura.
  - HistoryAdapter: Adaptador para mostrar el historial de juegos completados.
  - ImageAdapter: Adaptador para los juegos que muestran imágenes y es utilizado en la asignatura “Deportes”.
  - PickerAdapter: Adaptador para los juegos con una barra numérica y es utilizado en la asignatura “Matemáticas”.
  - ScoreAdapter: Adaptador para mostrar los resultados obtenidos en el juego y, también, sus respuestas correctas.
  - SelectAdapter: Adaptador para los juegos que muestran palabras y es utilizado en la asignatura “Conocimiento General”.
  - SoundAdapter: Adaptador los juegos que reproducen sonidos y es utilizado en la asignatura “Música”.
  - SpinnerAdapter: Adaptador para los juegos con desplegable de palabras y es utilizado en la asignatura “Inglés”.
- **Controller:** Controladores para responder a eventos del usuario.
  - ClassSingleton: Clase para obtener la instancia única de la clase del usuario.
  - GameSingleton: Clase para obtener la instancia única del juego seleccionado.
  - UserSingleton: Clase para obtener la instancia única del usuario.

- **Fragment:** Fragmentos que representan una parte de la interfaz en una actividad, y combinando múltiples fragmentos para crear una interfaz de usuario multipanel.
  - CardContentFragment: Fragmento con la lista de actividades.
  - ListContentFragment: Fragmento con la lista de usuarios de una clase.
  - TileContentFragment: Fragmento con la lista de asignaturas.
- **Model:** Modelos generados para representar la información de la plataforma (20).
  - ActivityModel: Modelo del objeto Actividad.
  - ClassModel: Modelo del objeto Clase.
  - GameModel: Modelo del objeto Juego.
  - QuizModel: Modelo del objeto Quiz.
  - UserModel: Modelo del objeto Usuario.
- **Service:** Servicios que realizan operaciones de larga ejecución en segundo plano y que no proporcionan una interfaz de usuario.
  - DoubleClickListener: Clase para interceptar el evento de pulsar dos veces.
  - HelperClient: Clase con métodos para facilitar la traducción español-inglés.
  - RestInterface: Interfaz con los servicios de la plataforma.
  - RetrofitClient: Clase para obtener la conexión a la base de datos.
- **View:** Vistas que presentan el modelo en un formato adecuado para interactuar.
  - ClassActivity: Vista con la información de la clase.
  - ClassEditActivity: Vista para editar y borrar la información de la clase.
  - ClassNewActivity: Vista para crear una nueva clase.
  - GameActivity: Vista con la información del juego.
  - GameEditActivity: Vista para editar y borrar la información del juego.
  - GameLookActivity: Vista para ver las soluciones correctas del juego.
  - HistoryActivity: Vista con el historial de juegos de un alumno o de la clase.
  - LoginActivity: Vista inicial de la aplicación para iniciar sesión.
  - MainActivity: Vista principal de la aplicación que tiene los tres fragmentos.
  - PlayActivity: Vista para realizar las preguntas de un juego y ver el resultado.
  - ProfileActivity: Vista con la información del usuario.
  - ProfileEditActivity: Vista para editar y borrar la información del usuario.
  - SearchActivity: Vista con la lista de juegos de una asignatura.
  - SignupActivity: Vista para registrar un nuevo usuario.
  - StatisticsActivity: Vista con las estadísticas de un alumno o de la clase.

### 5.2.1 Pantalla de inicio

Al pulsar el icono de la aplicación Android en el dispositivo móvil, la primera pantalla que aparece es la pantalla de iniciar de sesión (Figura 11), con el nombre de la plataforma y el logotipo en la parte superior de forma destacada; en el centro aparecen los campos de texto del apodo y la contraseña que se solicitan para comprobar la identidad del usuario; y en la parte inferior, los enlaces para confirmar el inicio de sesión o para el registro de un nuevo usuario en la base de datos.

El campo de texto de la contraseña oculta el texto introducido, pero se puede ver pulsando el botón de la izquierda. Además, si los datos no son válidos aparece un mensaje indicando que el apodo o la contraseña es incorrecto durante un corto periodo de tiempo.



*Figura 11: Clase LoginActivity*

En la pantalla de registrar (Figura 12) aparecen todos los campos que el nuevo usuario debe introducir para poder iniciar sesión en la aplicación; el campo de texto del identificador de la clase es opcional y, si el nuevo usuario no introduce nada, se le asigna la clase pública de forma automática.

Los dos campos de texto de la contraseña deben coincidir, el apodo del usuario es un atributo único por lo que se comprueba que esté disponible; también hay que comprobar que exista la clase con el identificador introducido y, si el usuario es profesor, tiene una imagen de avatar fijada que no puede modificar. Además, si los datos no son válidos, aparece un mensaje indicando que el apodo o el identificador de la clase es incorrecto, durante un corto periodo de tiempo.

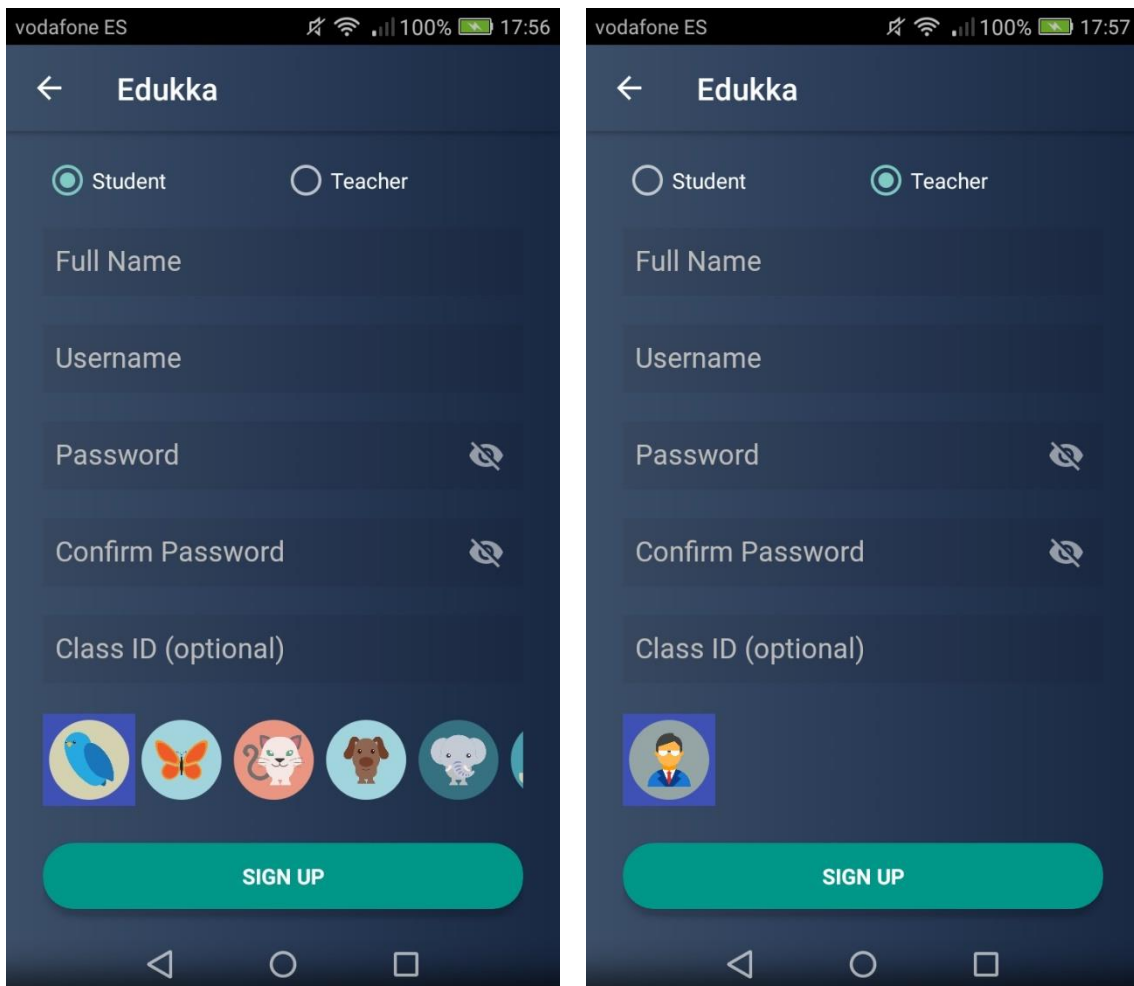


Figura 12: Clase SignupActivity

## 5.2.2 Pantalla principal

La pantalla está dividida en tres fragmentos, en todas aparece una barra superior con la imagen, el apodo del usuario, el icono de cada sección y un menú con cuatro opciones: Mi Perfil, Mi Clase, Ayuda y Salir.



Figura 13: Clase *TileContentFragment*

En el fragmento de código, Código 3, se muestra la creación de los tres fragmentos en el adaptador del objeto *ViewPager* que utiliza la clase *TabLayout*.

```
private void setupViewPager(ViewPager viewPager) {  
    Adapter adapter = new Adapter(getSupportFragmentManager());  
    adapter.addFragment(new TileContentFragment(), "Subjects");  
    adapter.addFragment(new ListContentFragment(), "Class");  
    adapter.addFragment(new CardContentFragment(), "Activity");  
    viewPager.setAdapter(adapter);  
    viewPager.setOffscreenPageLimit(3);  
}
```

Código 3: Método *setupViewPager*

El primer fragmento (Figura 13 izquierda) muestra la lista de asignaturas con su icono y el nombre de la asignatura en el idioma del dispositivo móvil.

El segundo fragmento (Figura 14 izquierda) muestra la lista de usuarios de la clase a la que pertenece el usuario con su imagen, apodo y nombre. Los tres alumnos con mayor puntuación tendrán una medalla especial de oro, plata o bronce como recompensa.

El tercer fragmento (Figura 14 derecha) muestra la actividad del usuario con un icono, nombre y descripción en el idioma del dispositivo móvil.

Todos los iconos de la aplicación se han obtenido de la página web “Icons8” (21).

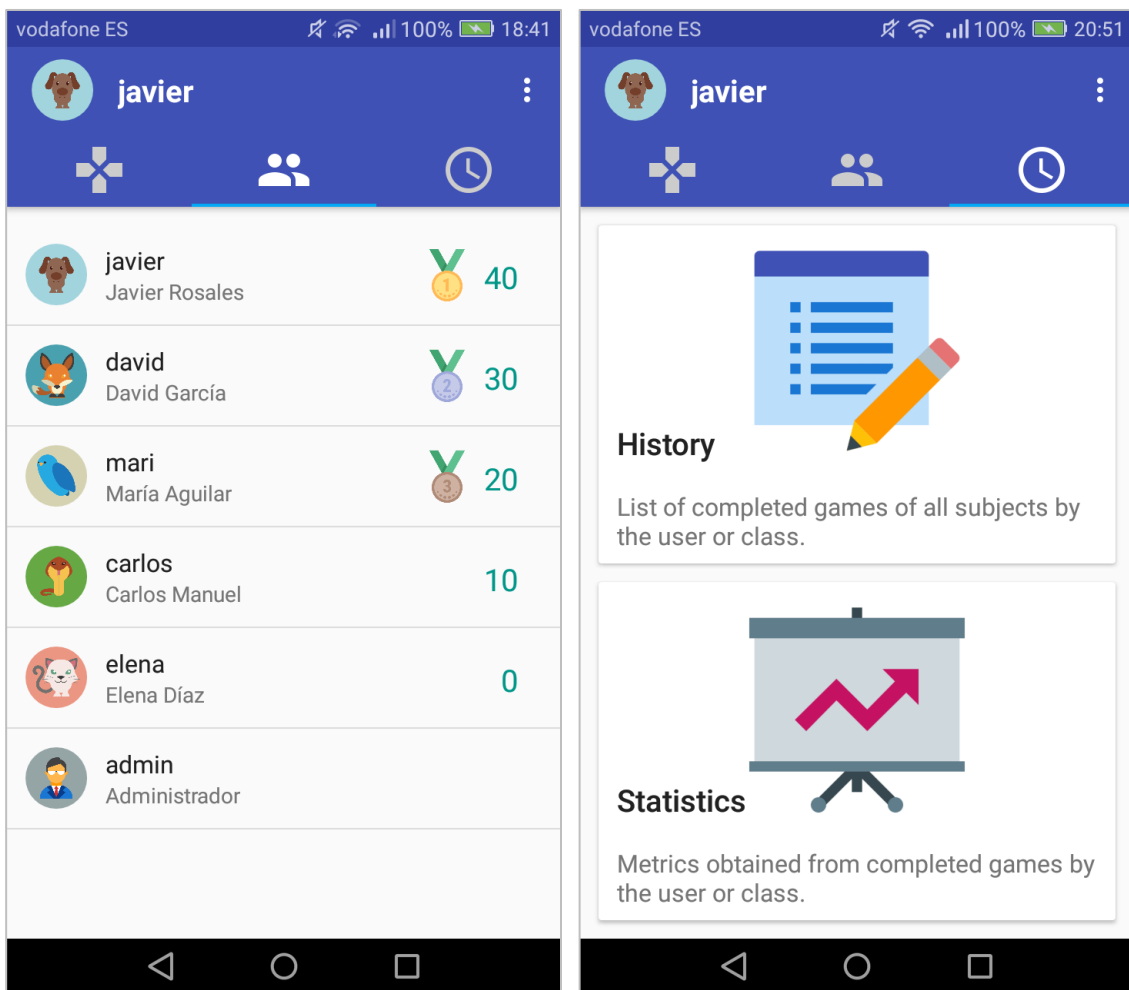


Figura 14: Clase ListContentFragment y CardContentFragment

### 5.2.3 Pantalla de perfil

La pantalla muestra la información del usuario seleccionado, con una barra que muestra la imagen y el apodo; también aparece el nombre, rol y, si es alumno, puntuación.

En la esquina inferior derecha hay un botón flotante para editar o borrar dicha información si es el perfil de tu usuario (Figura 15).

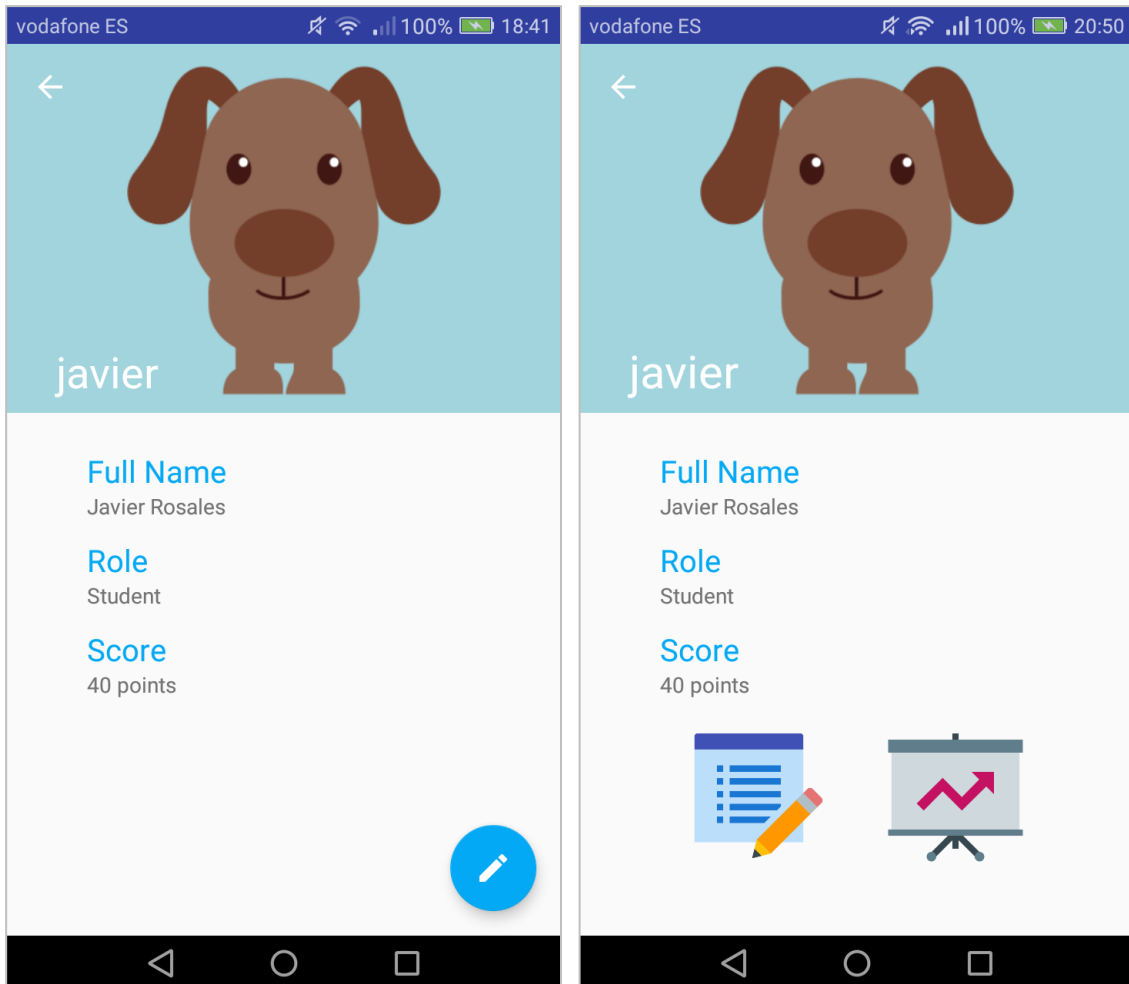


Figura 15: Clase ProfileActivity

En el fragmento de código, Código 4, se muestra la carga de la información mediante el cliente de Retrofit.

```
private void loadJSON() {
    RestInterface restInterface = RetrofitClient.getInstance();
    Call<UserModel> call = restInterface.getUser(position);
    call.enqueue(new Callback<UserModel>() {
        @Override
        public void onResponse(@NonNull Call<UserModel> call,
            @NonNull Response<UserModel> response) {
            UserModel jsonResponse = response.body();
            //...
        }
    });
}
```

Código 4: Método loadJSON

Al pulsar el botón flotante, aparece una pantalla con campos de texto para editar la información del usuario; hay que comprobar que el usuario está disponible y que los dos campos de texto de la nueva contraseña coincidan.

Para salir de esta pestaña sin guardar los cambios realizados, hay que pulsar la flecha en la esquina superior izquierda; para borrar el perfil hay que pulsar el icono de la papelera en la esquina superior derecha y aceptar esta acción irreversible en la ventana.

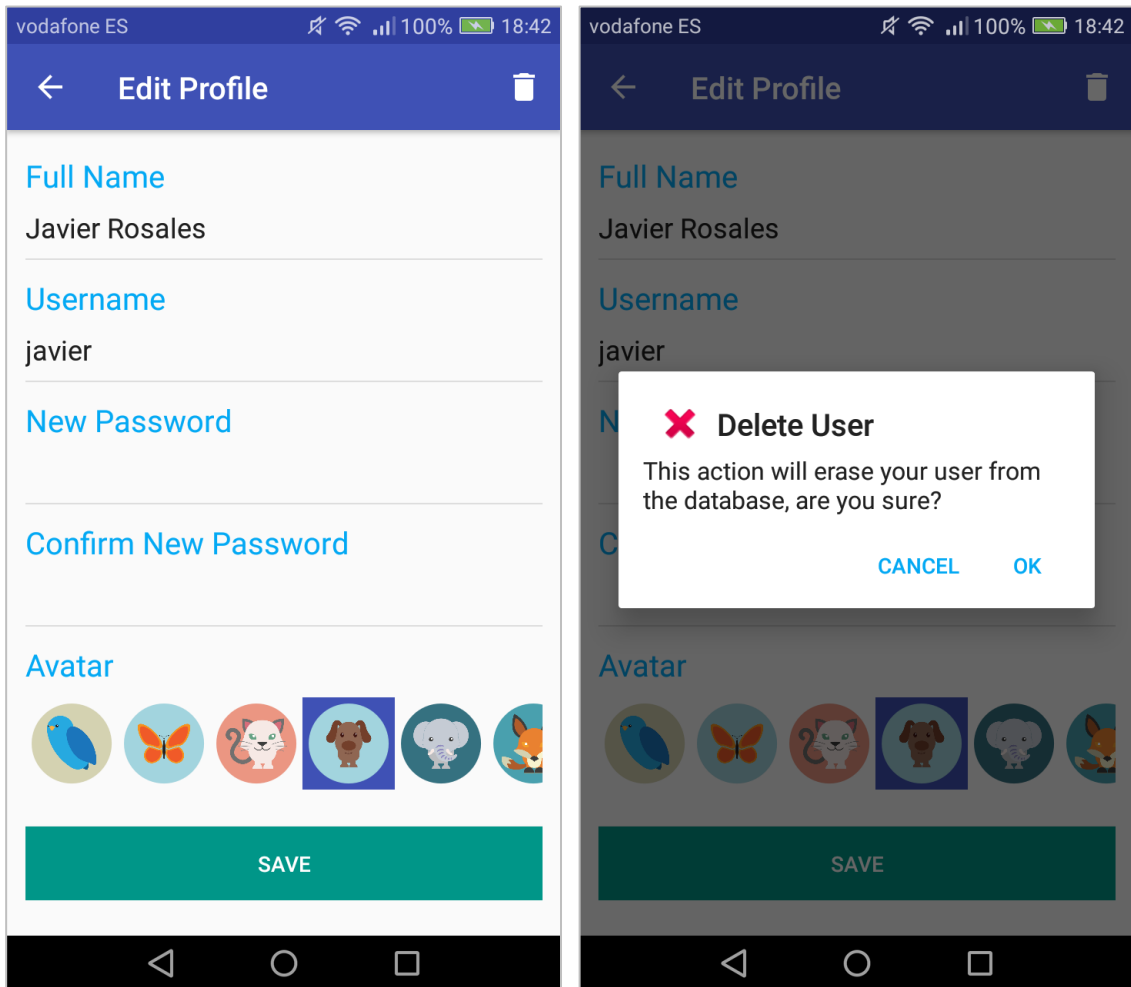


Figura 16: Clase ProfileEditActivity

### 5.2.4 Pantalla de clase

La pantalla muestra la información de la clase a la que pertenece el usuario, con una barra que muestra el icono y el nombre; también aparece el identificador, información y el número de usuarios de la clase. El identificador es el número que los usuarios deben introducir si quieren entrar a una clase privada.

Si el usuario es un profesor que se encuentra en la clase pública, aparece un botón flotante en la esquina inferior derecha para crear una nueva clase privada (Figura 17 izquierda) y, si el profesor es el creador de la clase, aparece otro botón flotante en la esquina inferior derecha para editar o borrar dicha información (Figura 17 derecha).

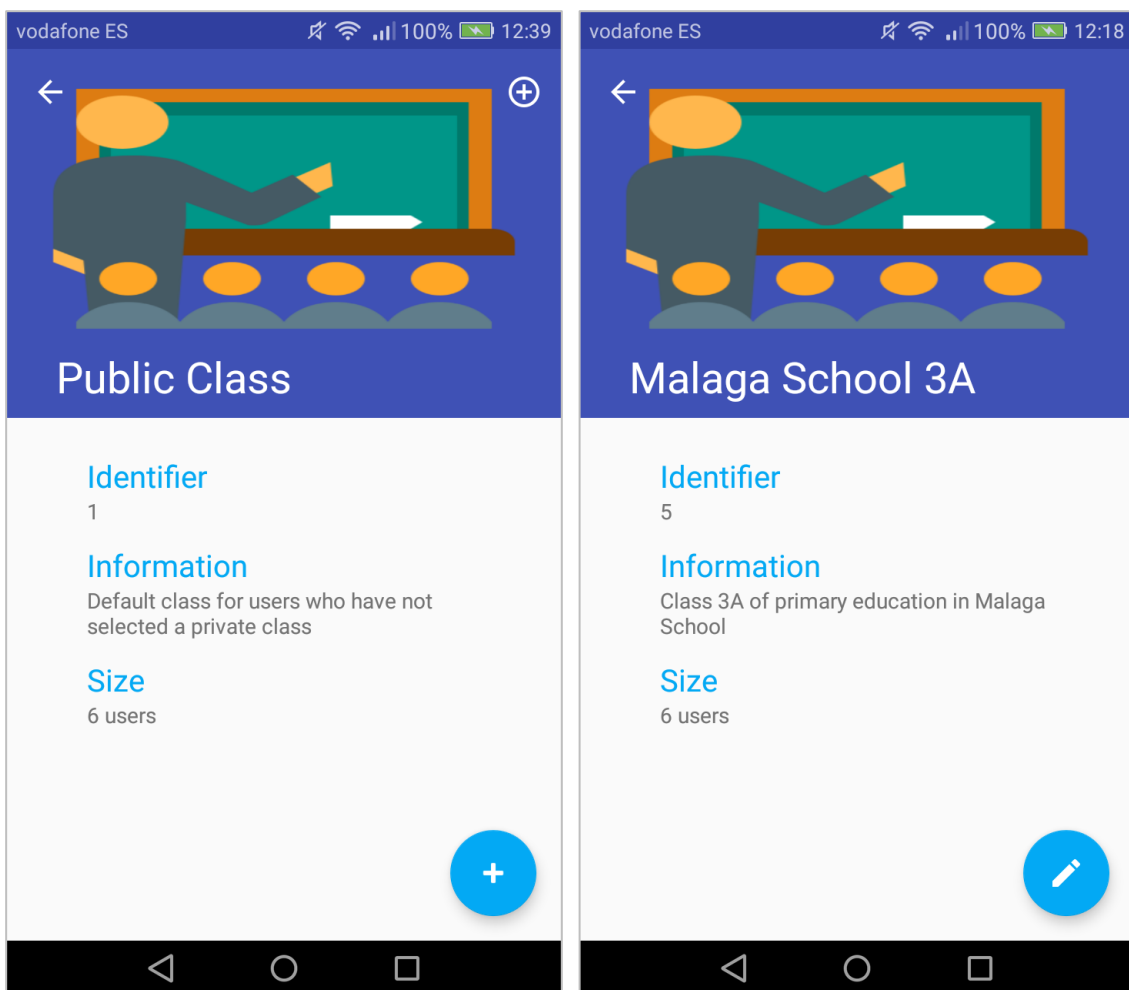


Figura 17: Clase ClassActivity (1)

Además, si el usuario se encuentra en la clase pública, puede entrar a una clase privada pulsando el icono + en la esquina superior derecha (Figura 18 izquierda) y hay que comprobar que exista la clase con el identificador introducido. Y si el usuario se encuentra en una clase privada, puede salir a la clase pública pulsando el icono - en la esquina superior derecha (Figura 18 derecha).

Los profesores no pueden salir de la clase que hayan creado, primero deben eliminar su clase. No se puede salir de la clase pública, todos los usuarios deben estar en una clase.

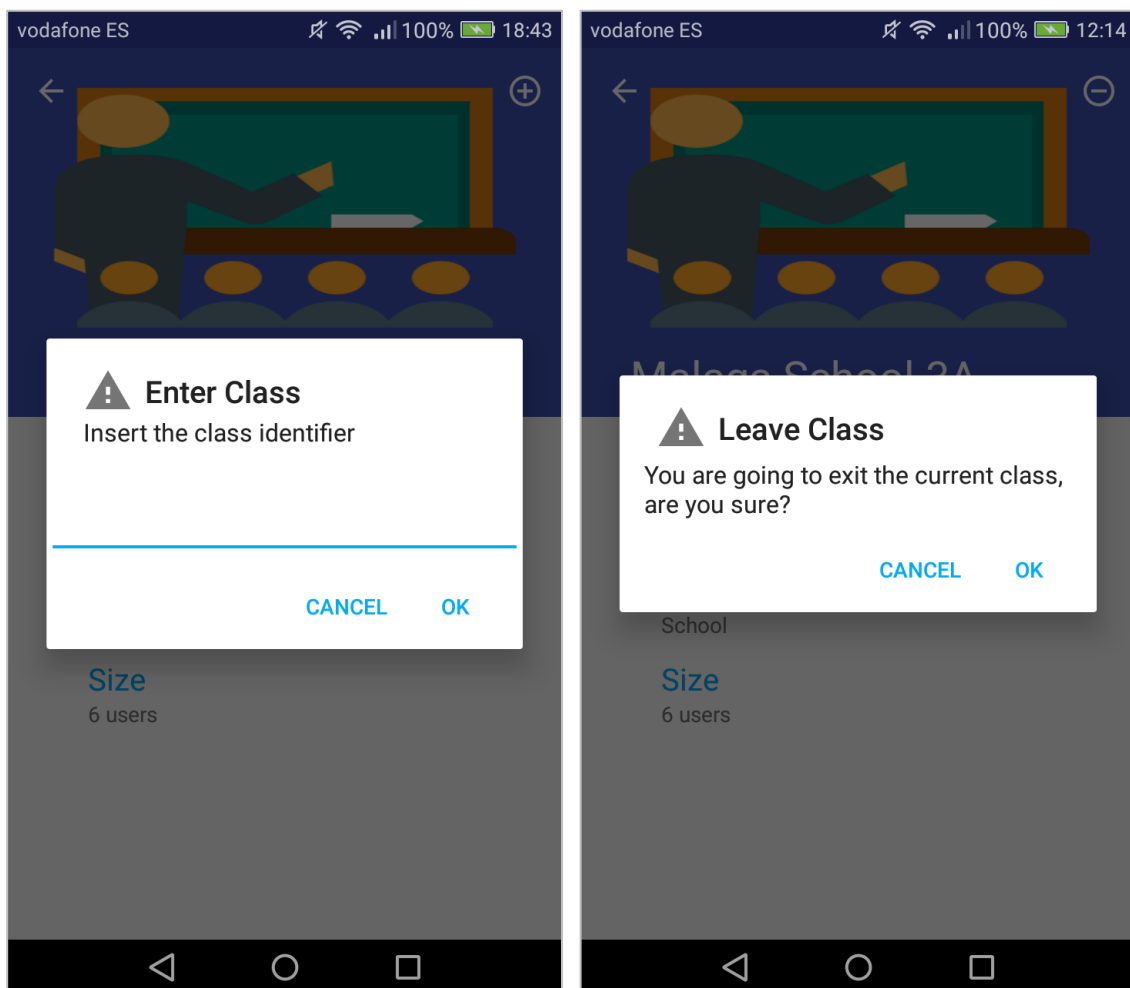
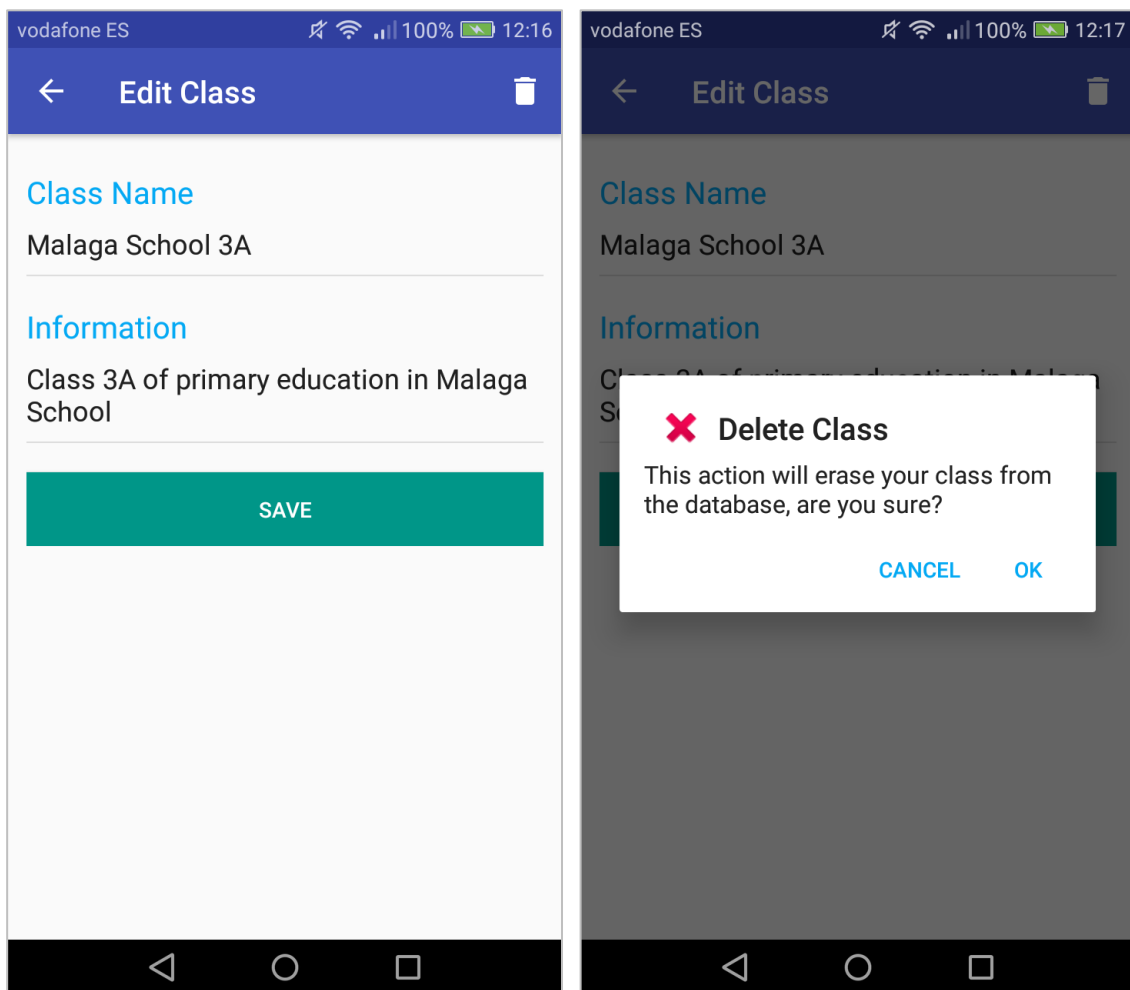


Figura 18: Clase ClassActivity (2)

En las pantallas de crear y editar clase (Figura 19) aparecen los campos de texto del nombre y la información.

Para salir de esta pestaña sin guardar los cambios realizados hay que pulsar la flecha en la esquina superior izquierda; para borrar la clase hay que pulsar el icono de la papelera en la esquina superior derecha y aceptar esta acción irreversible en la ventana.



*Figura 19: Clase ClassEditActivity*

### 5.2.5 Pantalla de actividad

La pantalla del historial muestra la lista de los juegos completados con el título del juego, la fecha de realización y la nota obtenida en forma de estrellas (Figura 20). Se puede actualizar la lista arrastrando la pantalla hacia abajo sin tener que abandonar la pantalla, aparece un icono circular de carga en el centro.

Para el profesor, el historial de la clase muestra el nombre del usuario al lado de la fecha y, también, puede ver el historial de un único usuario desde la pantalla de perfil del usuario pulsando el icono del historial (Figura 15 derecha).

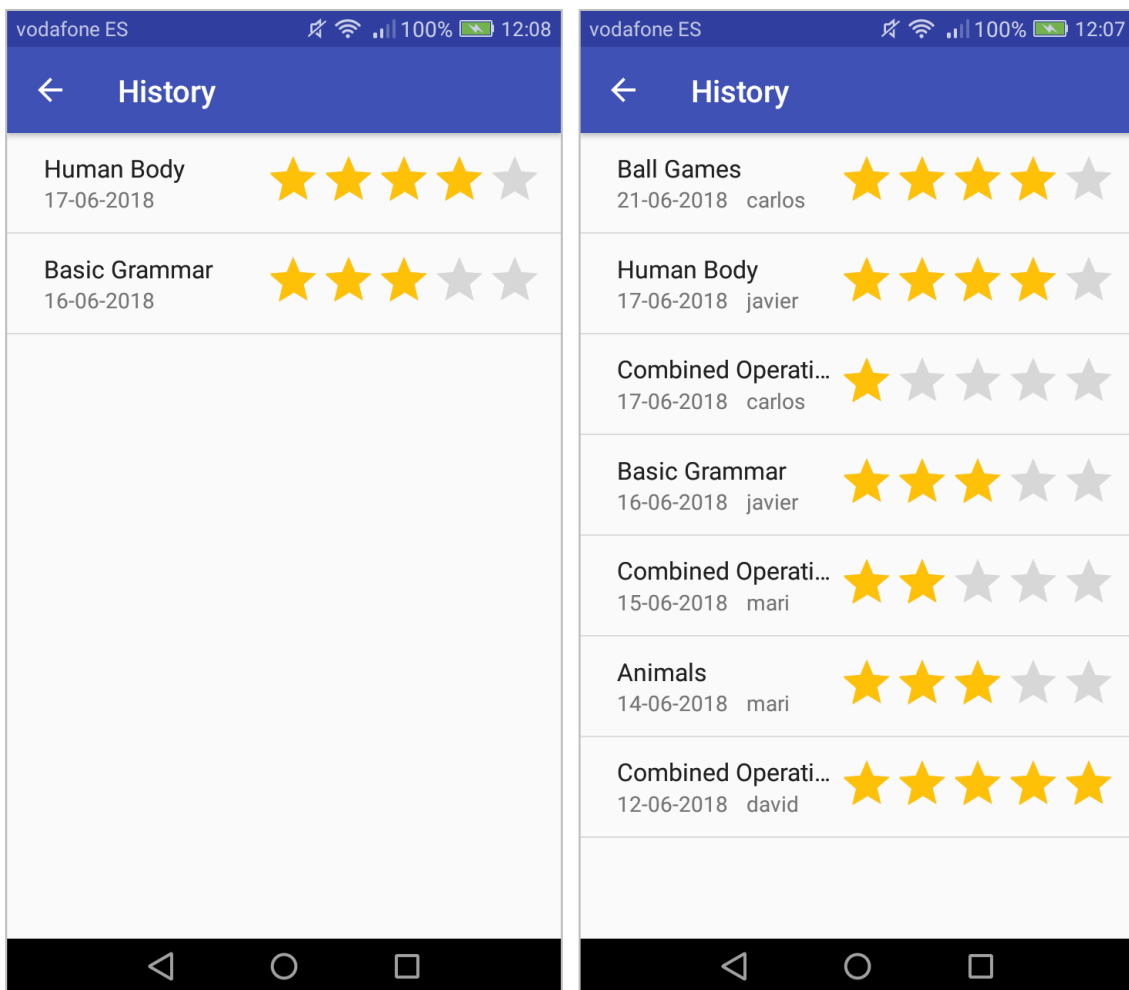


Figura 20: Clase HistoryActivity

La pantalla de las estadísticas muestra valores generales de los juegos completados, juegos aprobados, nota media, porcentaje de suspensos, asignatura más y menos jugada (Figura 21). Se puede actualizar los valores arrastrando la pantalla hacia abajo sin tener que abandonar la pantalla, aparece un icono circular de carga en el centro.

Para el profesor, además de las estadísticas de la clase, puede ver las estadísticas obtenidas por un único usuario desde la pantalla de perfil del usuario pulsando el icono de las estadísticas (Figura 15 derecha).

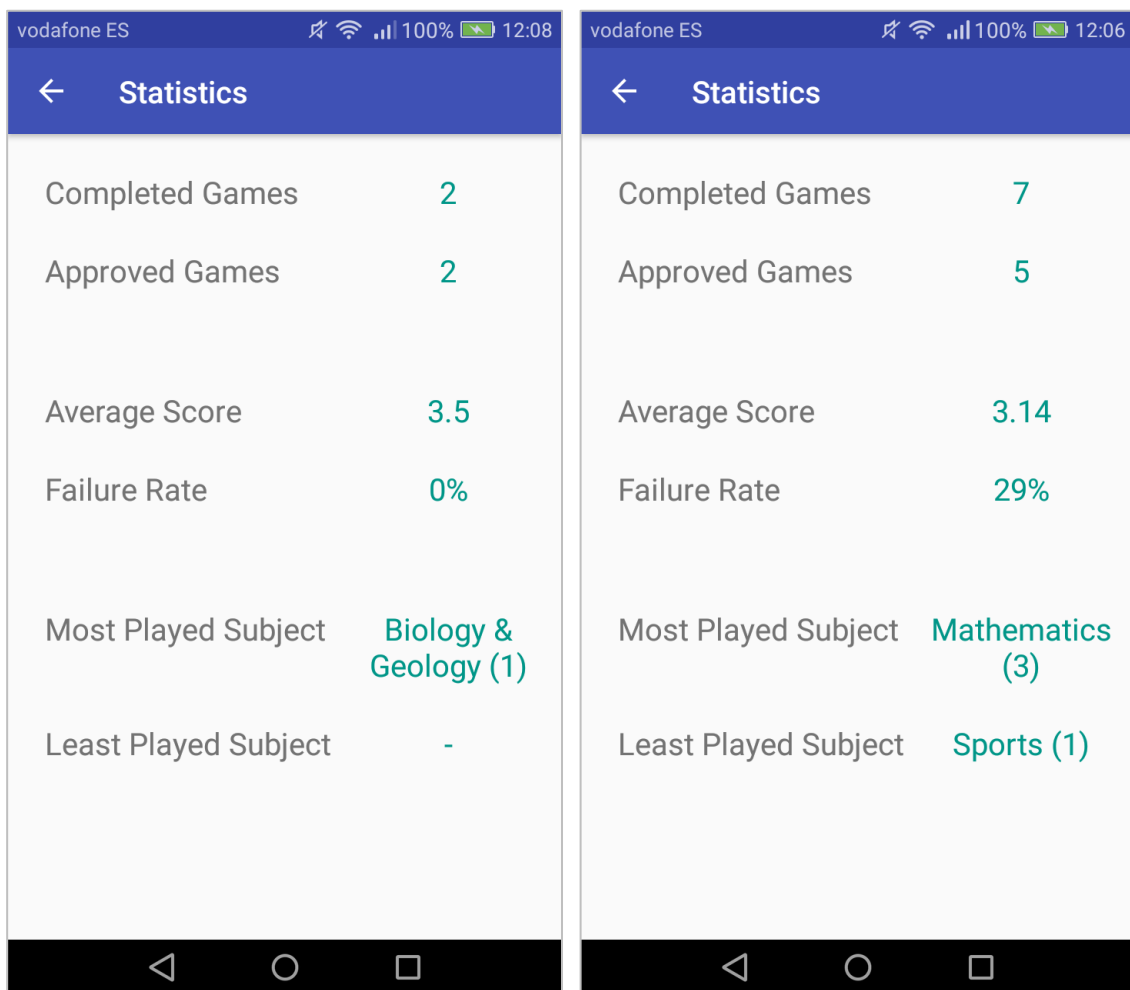
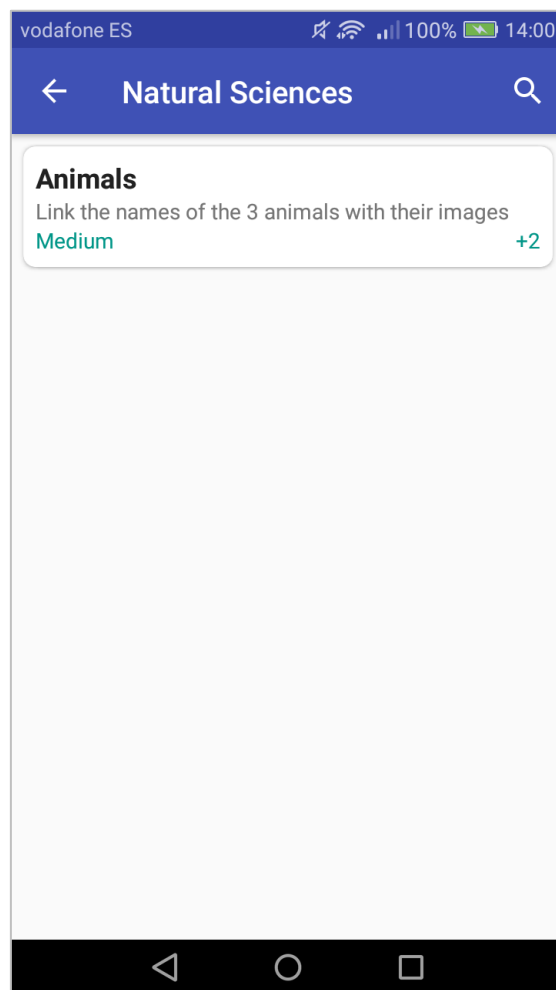


Figura 21: Clase StatisticsActivity

### 5.2.6 Pantalla de búsqueda

La pantalla muestra la lista de juegos de la asignatura seleccionada con el título, descripción, dificultad y los votos de los usuarios que pueden ser positivos, negativos o nulos. Se puede actualizar la lista arrastrando la pantalla hacia abajo sin tener que abandonar la pantalla, aparece un icono circular de carga en el centro.

Para una búsqueda personalizada en la lista, hay que pulsar el icono de la lupa en la esquina superior derecha y el filtrado es instantáneo (Figura 22).



*Figura 22: Clase SearchActivity*

### 5.2.7 Pantalla de juego

La pantalla muestra la información del juego seleccionado, con una barra que muestra el icono y el título; también aparece la descripción, dificultad y los votos de los usuarios que pueden ser positivos, negativos o nulos.

Cada asignatura tiene colores diferentes, para diferenciar todas las asignaturas de una forma visual (por ejemplo, en la Figura 23 vemos el color morado, que es el color propio de la asignatura “Ciencias Naturales”).

Además, en la esquina inferior derecha hay un botón flotante para editar o borrar dicha información si el profesor es un administrador de la plataforma.

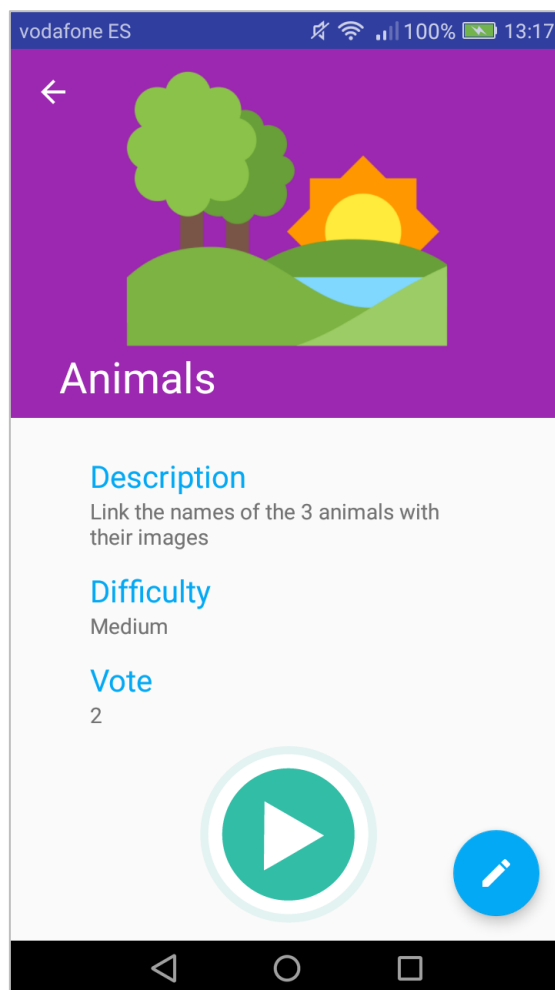


Figura 23: Clase GameActivity

En la pantalla de editar juego (Figura 24) aparecen los campos de texto del título, descripción y dificultad.

Para salir de esta pestaña sin guardar los cambios realizados hay que pulsar la flecha en la esquina superior izquierda; para borrar el juego hay que pulsar el icono de la papelera en la esquina superior derecha y aceptar esta acción irreversible en la ventana.

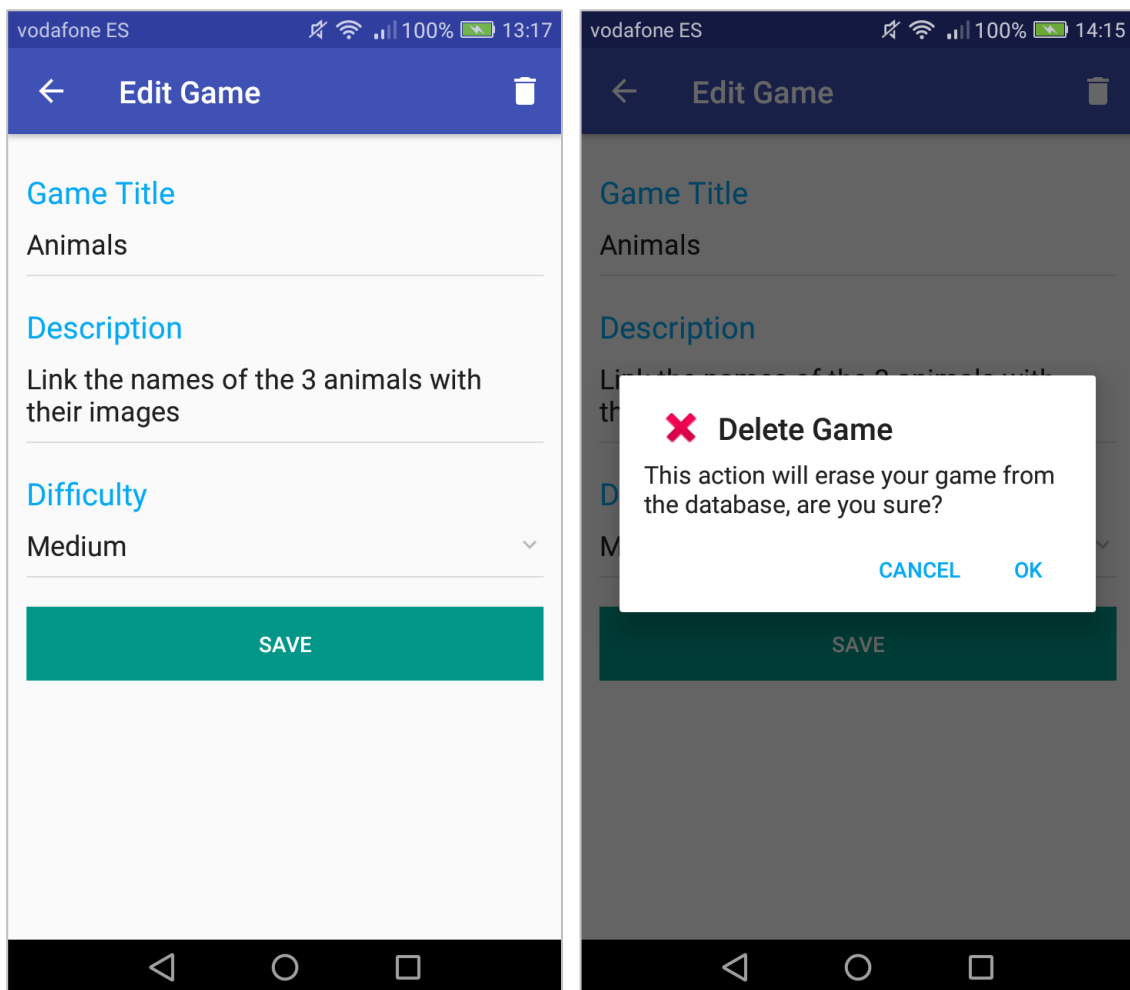


Figura 24: Clase GameEditActivity

Al pulsar el icono de jugar, aparece el primer quiz del juego seleccionado en la pantalla. En esta pantalla hay una parte común para todos los juegos y otra diferente para cada uno; la parte común consiste en la barra superior con el título del juego y un icono de ayuda para mostrar un mensaje con la descripción del juego, y también la barra inferior con la imagen del usuario, número actual, número total y línea de progreso del juego. Además, hay un botón flotante en la parte central derecha para continuar al siguiente quiz. Indicar que no es posible volver al anterior quiz, y si es el último quiz aparecen los resultados obtenidos en el juego.



Figura 25: Clase PlayActivity (1)

Al finalizar el juego, aparecen la lista con las preguntas, los resultados seleccionados y un icono que revela si el resultado es correcto o no. La barra inferior es modificada, ahora aparece la nota obtenida en forma de estrellas y los puntos que se sumarán a la puntuación del usuario si ha sacado una nota mayor o igual a 2,5.

Tras pulsar la flecha en la esquina superior izquierda para salir del juego, aparece una ventana para puntuar el juego de forma positiva o negativa, o cancelar para no puntuar.

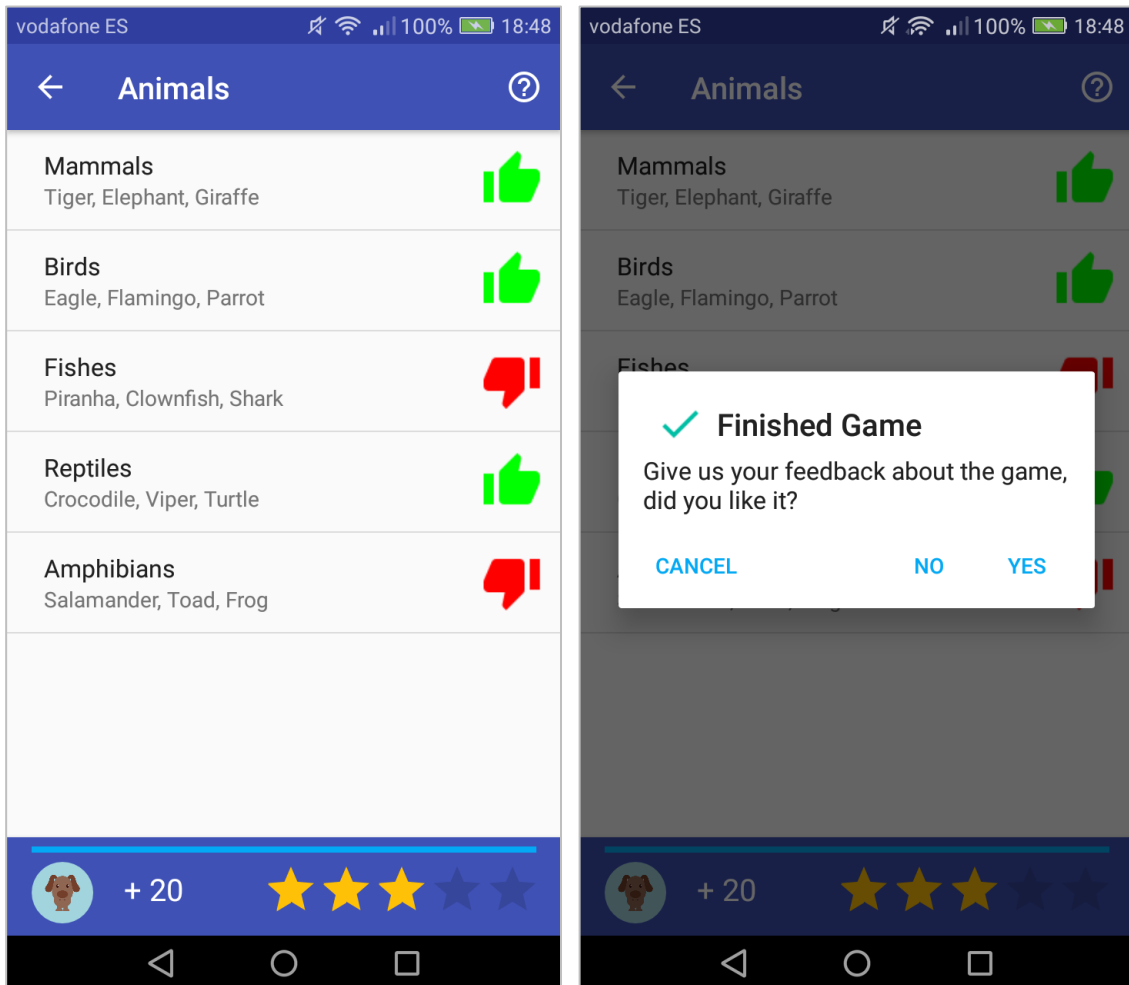


Figura 26: Clase PlayActivity (2)

En el fragmento de código, Código 5, se muestra parte de la funcionalidad del botón flotante que carga el siguiente quiz con el adaptador AdapterViewFlipper; muestra la pantalla con los resultados si el juego termina y llama a la función que guarda la nota en la base de datos, este código es igual para todos los juegos.

```

//...
flipper.showNext();

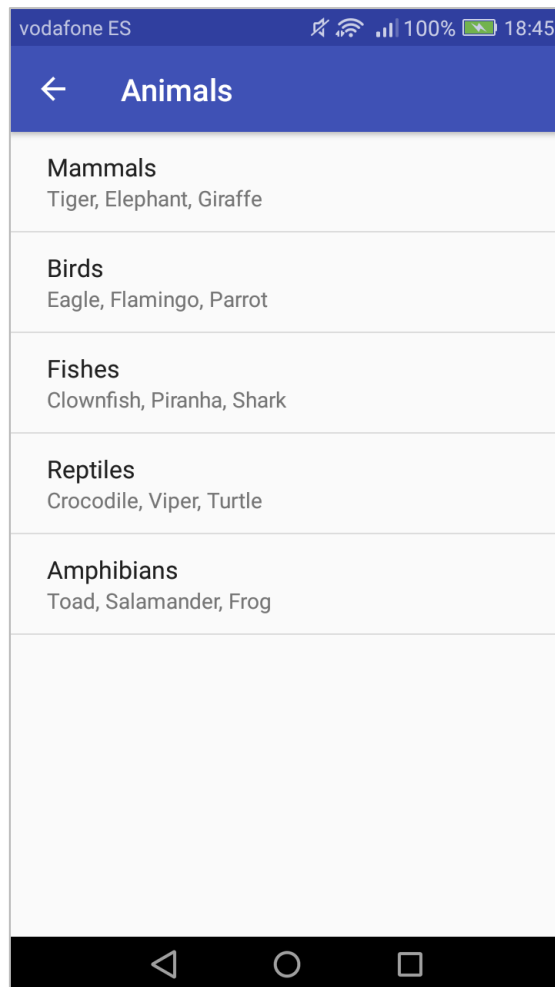
if (end) {
    RatingBar ratingBar = (RatingBar) findViewById(R.id.ratingBar);
    ratingBar.setRating((correct*5.0f) / questions.size());
    ratingBar.setVisibility(View.VISIBLE);
    fab.setVisibility(View.INVISIBLE);
    ScoreAdapter scoreAdapter = new ScoreAdapter(questions, values,
                                                results);

    recyclerView.setAdapter(scoreAdapter);
    finishGame();
}
//...

```

*Código 5: Terminar juego*

Como los profesores no pueden realizar juegos, al pulsar el icono de jugar, la pantalla muestra las soluciones correctas del juego (Figura 27).



*Figura 27: Clase GameLookActivity*

### 5.3 Pruebas

Para comprobar el funcionamiento de la plataforma se han realizado varios tests con la grabadora de pruebas “Espresso”. Es una herramienta de Android Studio que permite crear tests de IU sin la necesidad de escribir código de prueba. Al grabar un escenario de prueba, puedes grabar tus interacciones con un dispositivo y agregar asertos para verificar elementos de la IU en determinadas capturas de pantalla de la aplicación Android. Espresso usa la grabación guardada para generar de manera automática una prueba de IU correspondiente que puedes ejecutar para probar la plataforma.

Se han realizado un total de 10 tests, con resultados positivos (Figura 20), que son:

- **ClassActivityTest:** prueba para comprobar la salida de una clase privada y entrada a la clase pública, y también la salida de la clase pública y entrada a una clase privada.
- **ClassEditActivityTest:** prueba para comprobar que los datos modificados de la clase por el profesor se actualizan correctamente.
- **DeleteActivityTest:** prueba para comprobar el borrado de un usuario de la aplicación y, por tanto, de la plataforma.
- **GameLookActivityTest:** prueba para comprobar que el profesor tiene acceso a las soluciones de los juegos.
- **GamePlayActivityTest:** prueba para comprobar que el alumno tiene acceso a las preguntas de los juegos.
- **HistoryActivityTest1:** prueba para comprobar la actividad del alumno, es decir, el historial y las estadísticas personales.
- **HistoryActivityTest2:** prueba para comprobar la actividad del profesor, es decir, el historial y las estadísticas de los alumnos de la clase.
- **ProfileActivityTest:** prueba para comprobar que los datos modificados en el perfil del usuario se actualizan correctamente.
- **RegisterActivityTest:** prueba para comprobar la creación de un nuevo usuario en la aplicación y, por tanto, en la plataforma.
- **ViewActivityTest:** prueba grande para comprobar accesos a diferentes partes.

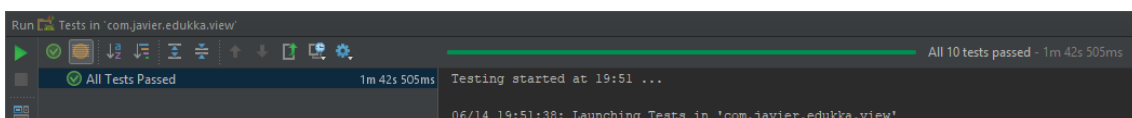


Figura 28: Resultados de los tests

## 5.4 Despliegue

### 5.4.1 Web Hosting

Para el alojamiento de la aplicación Web se ha utilizado 000webhost, que es uno de los servicios de hosting gratuito más grandes de la red, con alojamiento de dominio gratis, un panel de control simple y fácil de utilizar. También se puede actualizar la cuenta de hosting gratuito a hosting premium, con lo cual si tenemos un proyecto que crece mucho en tráfico podemos mejorar la cuenta y obtener un mayor rendimiento de nuestro alojamiento. Tiene soporte de PHP y MySQL, con gran capacidad de almacenamiento y ancho de banda, sin publicidad y ofrece una alta garantía en sus servidores (Figura 29).

Dirección web: <https://edukka.000webhostapp.com/>

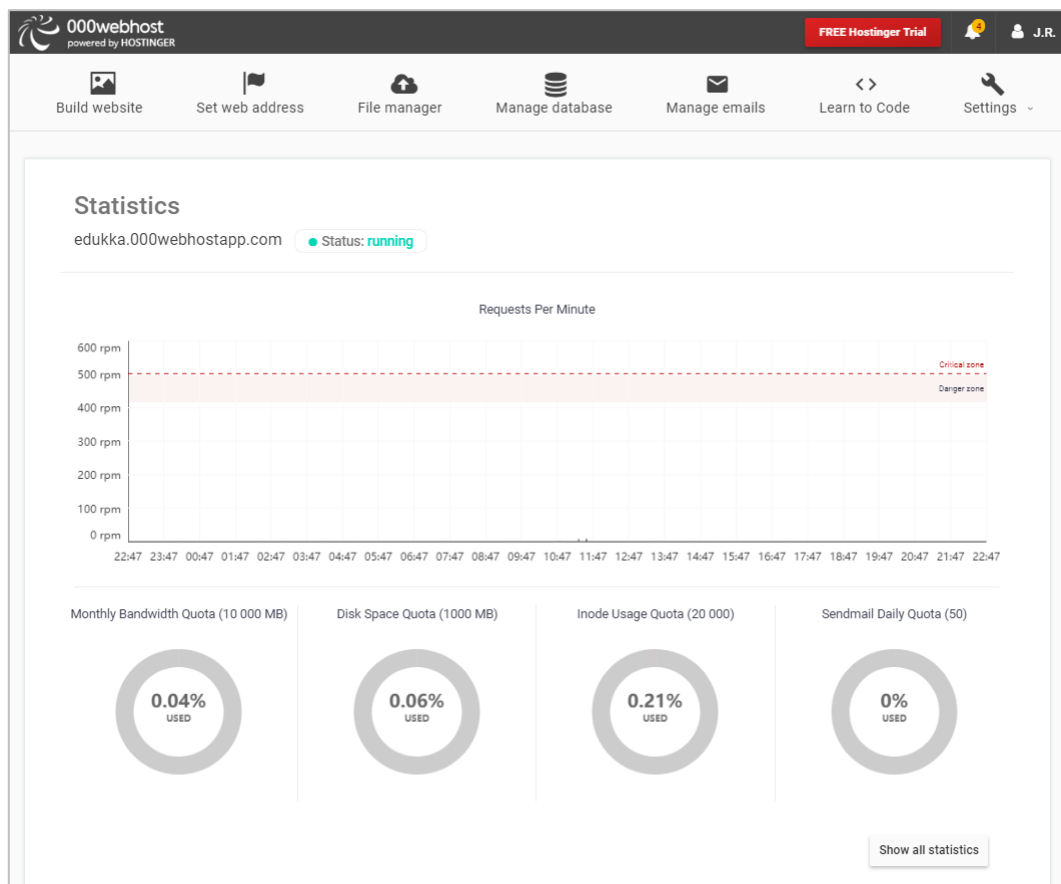


Figura 29: Panel de control en 000webhost

### 5.4.2 Google Play

Para el alojamiento de la aplicación Android se ha utilizado Google Play Console, que es una herramienta que permite a los programadores publicar aplicaciones y distribuirlas con facilidad directamente a los usuarios de teléfonos compatibles con Android. Se pueden administrar todas las fases de publicación y llegar a todos los usuarios activos de Android en Google Play (Figura 30). Además, existe la posibilidad probar la aplicación y recopilar información antes de su lanzamiento, administrar la distribución y los precios en las plataformas de Android, comprender y mejorar el rendimiento de la aplicación, encontrar nuevos usuarios, ver informes y más.

Enlace de descarga: <https://play.google.com/store/apps/details?id=com.javier.edukka>



Figura 30: Aplicación en Google Play

## CAPÍTULO 6. Conclusiones y líneas futuras

Con el uso de la tecnología en incremento, el aprendizaje en línea o e-learning continúa desarrollándose como una opción accesible a la educación y la práctica.

La plataforma Edukka proporciona muchas ventajas a los usuarios: mayor flexibilidad, facilidad de acceso, reducción de los tiempos de aprendizaje, aumento de la retención, compatibilidad de actividades, actualización inmediata de los contenidos, comodidad, formación personalizada y seguimiento exhaustivo del proceso de formación.

Durante la realización de este TFG he tenido la oportunidad de aprender el lenguaje de programación PHP y de programar aplicaciones Android, son elementos que no se enseñan en el Grado de Ingeniería del Software y, además, es la primera vez que publico una aplicación en la tienda Google Play.

A continuación, se describen un conjunto de mejoras que podrían ser muy interesantes:

- **Creación de juegos dentro de la aplicación**

Actualmente los juegos son creados por el administrador desde la aplicación Web, pero puede ser interesante que los profesores puedan crear juegos completos desde la aplicación Android y todos los usuarios tengan acceso a los nuevos juegos, lo cual podría generar una buena comunidad de usuarios.

- **Desafíos**

Una idea muy buena es tener una sección de desafíos para los alumnos. Consistiría en una lista de tareas aleatoria (por ejemplo, realizar tres juegos sin fallos) y completar dichos desafíos subiría la puntuación del alumno, lo cual sería útil para fomentar más la actividad y, también, que cada semana se fueran cambiando los desafíos.

- **Portabilidad a iOS**

Otra buena opción es publicar la aplicación en App Store para los dispositivos iOS, pero no es una tarea sencilla porque la aplicación se ha desarrollado de forma nativa para Android y habría que volver a codificar todo el proyecto.

- **Migración de Java a Kotlin**

Kotlin aparece como una nueva apuesta dentro del desarrollo como un lenguaje moderno, rápido, flexible y, sobre todo, pensando en la productividad y conceptos sencillos. Un aspecto importante es que Kotlin ya tiene soporte en Android Studio.



## Referencias

1. Ricardo Ducazal. Educación e-learning para niños: retos y características. 2016  
<http://www.avanzo.com/elearning-ninos-educacion/>
2. Virginia Gaitán. Gamificación: el aprendizaje divertido. 2013  
<https://www.educativa.com/blog-articulos/gamificacion-el-aprendizaje-divertido/>
3. Nuria Vallejo. Diseñar eLearning para niños y adolescentes. 2017  
<https://ojulearning.es/2017/02/disenar-elearning-para-ninos-y-adolescentes/>
4. Documentación Slim: <https://www.slimframework.com/>
5. Documentación Android: <https://developer.android.com/>
6. Documentación Material Design: <https://material.io/>
7. Sitio oficial de Retrofit: <http://square.github.io/retrofit/>
8. Repositorio de StatusBarUtil: <https://github.com/laobie/StatusBarUtil>
9. Repositorio de CircleImageView: <https://github.com/hdodenhof/CircleImageView>
10. Repositorio de MaterialEditText: <https://github.com/rengwuxian/MaterialEditText>
11. Repositorio de MaterialBetterSpinner: <https://github.com/Lesilva/BetterSpinner>
12. Sitio oficial de Picasso: <http://square.github.io/picasso/>
13. Sitio oficial de Lottie: <http://airbnb.io/lottie/>
14. Sitio oficial de GitKraken: <https://www.gitkraken.com/>
15. Sitio oficial de Postman: <https://www.getpostman.com/>
16. Sitio oficial de MagicDraw: <https://www.nomagic.com/products/magicdraw>
17. Sitio oficial de Balsamiq: <https://balsamiq.com/>
18. Sitio oficial de MySQL Workbench: <https://www.mysql.com/products/workbench/>
19. Sitio oficial de WAMP: <http://www.wampserver.com/en/>
20. Generar objetos Java desde JSON: <http://www.jsonschema2pojo.org/>
21. Sitio oficial de Icons8: <https://icons8.com/>



## Índice de figuras, tablas y códigos

Figura 1: Logotipo .....	11
Figura 2: Diagrama de Casos de Uso .....	15
Figura 3: Diagrama de Clases.....	36
Figura 4: Diagrama de la Base de Datos .....	36
Figura 5: Diagrama de Máquinas de Estado.....	39
Figura 6: Diagrama de la Interfaz.....	41
Figura 7: Diagrama de Componentes .....	42
Figura 8: Diagrama de Despliegue .....	43
Figura 9: Estructura del back-end.....	45
Figura 10: Estructura del front-end .....	48
Figura 11: Clase LoginActivity .....	51
Figura 12: Clase SignupActivity .....	52
Figura 13: Clase TileContentFragment .....	53
Figura 14: Clase ListContentFragment y CardContentFragment.....	54
Figura 15: Clase ProfileActivity.....	55
Figura 16: Clase ProfileEditActivity .....	56
Figura 17: Clase ClassActivity (1) .....	57
Figura 18: Clase ClassActivity (2) .....	58
Figura 19: Clase ClassEditActivity .....	59
Figura 20: Clase HistoryActivity.....	60
Figura 21: Clase StatisticsActivity .....	61
Figura 22: Clase SearchActivity.....	62
Figura 23: Clase GameActivity .....	63
Figura 24: Clase GameEditActivity .....	64
Figura 25: Clase PlayActivity (1).....	65
Figura 26: Clase PlayActivity (2).....	66
Figura 27: Clase GameLookActivity.....	67
Figura 28: Resultados de los tests .....	68
Figura 29: Panel de control en 000webhost .....	69
Figura 30: Aplicación en Google Play .....	70

Tabla 1: Requisitos Funcionales.....	8
Tabla 2: Requisitos No Funcionales.....	10
Tabla 3: Funciones .....	47
Código 1: Fichero index .....	46
Código 2: Función getUser.....	46
Código 3: Método setupViewPager.....	53
Código 4: Método loadJSON .....	55
Código 5: Terminar juego .....	67

