

# **DESARROLLO DE UNA APLICACIÓN EDUCATIVA PARA LAS ASIGNATURAS DE IS1 E IS2**

**RUBÉN GARCÍA MATEOS**

**GRADO EN INGENIERÍA DEL SOFTWARE  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID**



**TRABAJO DE FIN DE GRADO  
CURSO 2019-2020**

**DIRECTOR:  
ANTONIO SARASA CABEZUELO**

# **DEVELOPMENT OF AN EDUCATIONAL APPLICATION FOR THE SUBJECTS OF IS1 AND IS2**

**RUBÉN GARCÍA MATEOS**

**GRADO EN INGENIERÍA DEL SOFTWARE  
FACULTAD DE INFORMÁTICA  
UNIVERSIDAD COMPLUTENSE DE MADRID**



**FINAL DEGREE PROJECT**

**COURSE 2019-2020**

**DIRECTOR:**

**ANTONIO SARASA CABEZUELO**



# ÍNDICE

CAPÍTULO 1: INTRODUCCIÓN.....	1
1.1 Motivación .....	1
1.2 Objetivos .....	1
1.3 Estructura de la memoria.....	1
CAPÍTULO 2: ESTADO DEL ARTE .....	3
2.1 Juegos educativos y de aprendizaje .....	3
2.2 Software destinado a la realización de cuestionarios y exámenes.....	3
CAPÍTULO 3: TECNOLOGÍA EMPLEADA .....	5
3.1 Herramientas back-end.....	5
3.1.1 Java .....	5
3.1.2 Spring.....	5
3.1.3 Tomcat.....	6
3.1.4 JWT .....	6
3.1.5 JPA/Hibernate .....	6
3.1.6 Bases de datos MySQL/MariaDB.....	7
3.1.7 Spring Tools Suite (STS) .....	7
3.2 Herramientas front-end .....	7
3.2.1 Android.....	7
3.2.2 Retrofit .....	7
3.2.3 Android Studio .....	8
3.3 Otras herramientas .....	8
3.3.1 GitHub .....	8
3.11 Ngrok.....	8
CAPÍTULO 4: ESPECIFICACIÓN DE REQUISITOS .....	9
4.1 Actores .....	9
4.2 Profesor .....	9
4.3 Alumno .....	14
4.4 Administrador .....	16
4.5 Casos de uso comunes .....	18
CAPITULO 5: MODELO DE DATOS .....	23
5.1 Modelo E-R.....	23
5.2 Implementación de la base de datos .....	25
5.2.1 Tabla User.....	25
5.2.2 Tabla Tema .....	26
5.2.3 Tabla Question .....	26

5.2.4 Tabla NodoCA.....	26
5.2.5 Tabla NodoAlternativo .....	27
5.2.6 Tabla Logros .....	27
5.2.7 Tabla Answer .....	27
5.2.8 Tabla Estadísticas .....	27
5.2.9 Tabla Question_Suspended.....	28
CAPÍTULO 6: ARQUITECTURA DE LA APLICACIÓN .....	29
6.1. Modelo cliente-servidor.....	29
6.2 Patrones de diseño.....	29
6.2.1 Capa de cliente .....	29
6.2.2 Capa de servidor.....	30
CAPÍTULO 7: DISEÑO DE LA APLICACIÓN .....	31
7.1 Colores, tipografía e imágenes.....	31
7.2 Funcionalidad del cliente .....	32
7.2.1 Ciclo de vida de una actividad y carga del menú .....	33
7.2.2 Módulo Inicial, Login y Registro .....	37
7.2.3 Módulo Home o Perfil. ....	40
7.2.4 Módulo Ayuda.....	47
7.2.5 Módulo Ranking. ....	49
7.2.6 Módulo Juego.....	53
7.3 API REST.....	71
7.3.1 Módulo Usuario.....	74
7.3.2 Módulo Question .....	77
7.3.3 Módulo Tema .....	79
7.3.4 Módulo NodoCA.....	79
7.3.5 Seguridad de la API.....	82
CAPITULO 8: EVALUACIÓN .....	84
8.1 Diseño de la evaluación.....	84
8.2 Resultados de la evaluación .....	88
8.2.1 Preguntas referidas al rol alumno.....	88
8.2.2 Preguntas referidas al rol profesor. ....	91
8.2.3 Preguntas referidas al rol administrador. ....	93
8.2.4 Valoraciones generales .....	94
CAPÍTULO 9: CONCLUSIONES Y TRABAJO FUTURO .....	97
9.1 Conclusiones.....	97
9.2 Trabajo futuro .....	97

CHAPTER 9: CONCLUSIONS AND FUTURE WORK.....	98
9.1 Conclusions.....	98
9.2 Future work.....	98
REFERENCIAS.....	99
ANEXO .....	101
Anexo I. Guía de uso.....	101
Anexo II. Guía de instalación .....	129
1. Instalación y arranque del servidor y Base de datos.....	129
2. Instalación y arranque de la aplicación móvil .....	132
Anexo III. Preguntas de evaluación completas .....	136
1. Primer bloque.....	136
2. Segundo bloque.....	138
3. Tercer bloque .....	149



## **RESUMEN**

En este trabajo se describe una aplicación educativa e interactiva para el ámbito de la ingeniería del software. El objetivo de esta es complementar la formación que reciben los alumnos de forma presencial mediante un juego basado en la realización de preguntas que están agrupadas por niveles de dificultad. La superación de un nivel permite obtener una medalla que actúa a modo de reconocimiento del aprendizaje realizado. Se espera que la motivación que produce obtener un obsequio virtual haga que los estudiantes se interesen por la aplicación y que, de forma no directa, el estudiante aprenda.

Se ha implementado como una app para móviles Android, y ofrece servicio tanto a profesores como alumnos. Los profesores pueden diseñar caminos de aprendizaje formados por un conjunto de preguntas que se proponen al alumno de forma secuencial, y organizadas dinámicamente de acuerdo con las calificaciones y resultados que va obteniendo el alumno. Así mismo existe el rol del administrador que realiza operaciones de mantenimiento en la aplicación.

## **SUMMARY**

This work describes an educational and interactive application for the field of software engineering. The objective of the same is complement the training that students receive in person through a game based on asking questions that are grouped by levels of difficulty. The overcoming of a level allows to obtain a medal that acts as a recognition of the learning done. The motivation of obtaining a virtual gift is expected to make students interested in the application and not directly, the student learns.

It has been implemented as an Android mobile app and offers service to both teachers and students. Teachers can design learning paths formed by a set of questions that are proposed to the student in a sequential way and dynamically organized according to the grades and results that the student is obtaining. There is also the role of the administrator who performs maintenance operations on the application.

# **CAPÍTULO 1: INTRODUCCIÓN**

En este primer capítulo de la memoria se va a explicar la motivación de la aplicación desarrollada, cuáles han sido sus objetivos y se describe la estructura de la memoria.

## **1.1 Motivación**

La causa principal para implementar la aplicación ha sido la desmotivación del alumnado debido en gran parte a ser una asignatura que se sale de lo “normal” pues no es tan práctica como otras de programación.

También la dificultad procedente de la propia naturaleza de la asignatura dado que para ilustrar algunos conceptos es complicado, pues sólo se puede ver en proyectos grandes y reales.

Por esto se ha desarrollado una aplicación que sea de ayuda a los profesores ofreciéndoles un medio adicional para poder enseñar de una forma divertida para el estudiante, y, por otro lado, al alumno le ofrece la posibilidad de aprender de una forma motivadora y, además, muy accesible pues solo necesita el uso del móvil.

## **1.2 Objetivos**

El objetivo principal de este proyecto era desarrollar un sistema que favoreciera el aprendizaje de los alumnos de Ingeniería del software de la Facultad de informática en la Universidad Complutense de Madrid.

Los objetivos específicos planteados en el proyecto fueron los siguientes:

- Desarrollar una aplicación que permitiera a los profesores de IS observar el interés de los alumnos por la asignatura y otorgar una puntuación extra a aquellos que participen de forma activa en el juego.
- Permitir a los profesores de IS mantener un repositorio común de preguntas y exámenes visible para todos ellos.
- Permitir a los alumnos de IS adquirir conocimientos de una forma más divertida e interactiva, con objetivos preestablecidos y de una manera eficaz para el aprendizaje.
- Permitir a los profesores una alternativa a los porcentajes variables que se otorgan por trabajo en prácticas.
- Permitir a los profesores crear caminos de aprendizaje de una manera rápida y sencilla
- Permitir a los alumnos realizar estos caminos de aprendizaje publicados por los profesores
- Permitir a los alumnos de intercambio que no dominan correctamente el idioma español, se integren con la asignatura al ofrecer la aplicación soporte para el inglés.

## **1.3 Estructura de la memoria**

La memoria de este proyecto se estructura de la siguiente manera:

- En el capítulo 1 se introduce el trabajo realizado y la motivación para realizar su desarrollo. Así mismo se describe la estructura de la memoria.
- En el capítulo 2 se revisan algunas herramientas con una funcionalidad semejante a la desarrollada en este trabajo.
- En el capítulo 3 se comenta la tecnología empleada para la implementación.

- El capítulo 4 se describen los casos de uso de la aplicación.
- En el capítulo 5 se describe la forma en la que se ha realizado la persistencia de la información del sistema creado.
- El capítulo 6 trata sobre la arquitectura y los patrones de diseño utilizados en la aplicación.
- En el capítulo 7 se trata el diseño y funcionalidad de la aplicación, dividido en dos secciones, en primer lugar, la aplicación web y, en segundo lugar, la API REST.
- En el capítulo 8 se desarrolla la evaluación por parte de usuarios encuestados
- En el capítulo 9 se describen las conclusiones y el trabajo futuro.
- En último lugar se encuentra la bibliografía y el anexo. Este último incluye una guía de uso, así como una guía de instalación tanto de la aplicación Android como de la API

## CAPÍTULO 2: ESTADO DEL ARTE

En el presente capítulo se recogerán las características más importantes acerca de otros proyectos semejantes que hay actualmente en el mercado.

La aplicación implementada es un “juego” de aprendizaje y formación para alumnos, así como una herramienta de evaluación para profesores. En este ámbito se han encontrado dos tipos de aplicaciones: los juegos educativos y de aprendizaje, y el software destinado a la realización de cuestionarios y exámenes.

### 2.1 Juegos educativos y de aprendizaje

Son aplicaciones que permiten a los usuarios adquirir y reforzar el aprendizaje en cualquier área de conocimiento. Algunos ejemplos de este tipo de juegos podrían ser:

- **Duolingo**<sup>[1]</sup>: Es una plataforma de aprendizaje de idiomas mediante la realización de cuestionarios. Estos cuestionarios pueden ser de diferentes tipos como por ejemplo completar frases, adivinar palabras, relacionar vocabulario...etc.

La aplicación ayuda a traducir textos y entre los idiomas sobre los que se puede estudiar destacan el español, francés, inglés, alemán e italiano.

El juego consiste en ir ganando puntos por resolver preguntas de varios tipos en cada tema, a medida que se avanza en el juego aumenta la dificultad y se va teniendo un nivel superior del idioma.

- **M.A.R.S (Misión para Aprender Repasando con SM)**<sup>[2]</sup>. Se trata de una aplicación que adapta el contenido de asignaturas de educación primaria, hasta sexto, para el aprendizaje y repaso de los contenidos dados durante el curso. Cuenta con bastantes juegos para el repaso de las asignaturas basados en misiones.

El juego comienza seleccionando un curso de primaria sobre el que se quiera repasar y el usuario deberá resolver los minijuegos que se proponen.

- **Oxford Learner’s Quick Grammar** <sup>[3]</sup>: Aplicación destinada al aprendizaje de inglés, con vocabulario de distintos niveles. Se puede repasar contenido como tiempos verbales, adverbios, expresiones gramaticales y varios ejemplos. Cuenta también con una serie de consejos sobre ciertas expresiones del idioma.

El juego cuenta con un reconocimiento de voz donde poder practicar las expresiones más usuales del idioma y perfeccionar el acento.

### 2.2 Software destinado a la realización de cuestionarios y exámenes.

Se trata de aplicaciones que permiten generar test de preguntas esencialmente. Algunos de ellos también permiten la comunicación por foros y la visualización de resúmenes de teoría. Unos posibles ejemplos de este tipo de plataformas podrían ser:

- **Todotest**<sup>[4]</sup>: Aplicación web para prepararse el carné de conducir. Ofrece la posibilidad de realizar test para que el alumno se forme en cualquiera de los carnés, ya sea de motocicleta, coche, camión y autobús. También brinda la posibilidad de enviar las dudas a través de distintos foros y cuenta con un apartado de resúmenes sobre los distintos temas.

Al entrar en la aplicación el usuario puede seleccionar si quiere realizar los cuestionarios oficiales de la DGT, que contienen las últimas preguntas subidas por este organismo, cuestionarios con preguntas de exámenes, que contienen preguntas de todos los temas, o si quiere realizar cuestionarios de temas en particular.

El usuario puede visualizar todos los resultados obtenidos en los cuestionarios y, además, ver las preguntas en las que ha fallado.

- **EasyLMS**<sup>[5]</sup>: Herramienta para las empresas destinada principalmente a la formación de sus trabajadores. Se pueden crear cursos para la formación y cuestionarios, además de llevar un control sobre los resultados del personal.

También da la opción de crear exámenes más elaborados, visualizar las estadísticas de cada curso como el tiempo de visualización de los videos, resultados de cuestionarios, personas que han realizado los cursos...etc.

Se pueden otorgar certificaciones empresariales a través de esta plataforma.

- **Google Forms**<sup>[6]</sup>, permite crear todo tipo de cuestionarios y evaluaciones para realizar un seguimiento sobre un tema determinado. También permite la creación de exámenes. Posee una interfaz sencilla y de creación rápida de estos cuestionarios.

Permite realizar evaluaciones sobre algún tema en concreto, lleva en cuenta todas las estadísticas de cada pregunta que se añade y se muestran en forma de gráficos. Brinda la posibilidad de subir imágenes y videos para los cuestionarios y además permite modificar la interfaz de cada una de las preguntas.

- **Kahoot**<sup>[7]</sup>: Aplicación web que permite la creación de exámenes y cuestionarios online. Entre sus características principales puede mencionarse la posibilidad de realizar un seguimiento en tiempo real, por lo que facilita la evaluación y seguimiento en un momento determinado de un grupo de personas. Tiene una interfaz amigable con una amplia paleta de colores. Es muy utilizada en el ámbito académico por estas características.

## CAPÍTULO 3: TECNOLOGÍA EMPLEADA

En este capítulo, se van a describir las herramientas tecnológicas empleadas en el desarrollo de este proyecto. Se realizará una clasificación por herramientas back-end, front-end y otras.

### 3.1 Herramientas back-end

#### 3.1.1 Java

Java [8] es un lenguaje de programación y una plataforma digital creado en 1995. Guarda relación con otros lenguajes como C y C++. Puede ejecutarse en cualquier máquina virtual que tenga Java sin importar el tipo de arquitectura. Es un lenguaje de programación que cuenta con las siguientes características:

- Orientado a objetos
- Distribuido, ya que cuenta con grandes capacidades de interconexión
- Robusto al buscar problemas en tiempo de compilación y ejecución
- Seguro al prevenir accesos ilegales a la memoria ya que no tiene contemplado la aritmética de punteros como en otros lenguajes.
- Portable, ya que su portabilidad a otros sistemas es sencilla
- Interpretado al poder ejecutar directamente el código objeto, aunque esta característica le hace más lento con respecto a otros lenguajes de programación.

Una de sus mayores peculiaridades es su recolector de basura, ya que en tiempo de ejecución es este el que decide sobre el ciclo de vida de los objetos sobre los que trabaja la aplicación y además evitar posibles fugas liberando la memoria.

#### 3.1.2 Spring

Spring [9] es un framework o marco de trabajo para el lenguaje de programación Java utilizado para desarrollar aplicaciones web principalmente en el lado del servidor.

Su funcionamiento se basa principalmente en el uso de Beans<sup>1</sup> u objetos que se inyectan en las diferentes clases para su uso.

Spring contiene varios módulos para su utilización entre los que destacan los siguientes:

- Contenedor de inversión de control, similar al contenedor Java, administra el ciclo de vida de los Beans de Spring.
- Programación orientada a aspectos
- Acceso a datos y gestión de transacciones. Con algunas partes de Spring se puede gestionar de forma eficiente y concurrente el acceso y manejo de los datos.
- Modelo Vista-Controlador: Es un framework que contempla principalmente el desarrollo de Apis REST utilizando el patrón modelo vista controlador tan conocido.

---

<sup>1</sup> Un Bean es un objeto manejado por el contenedor de Spring y representan objetos de tipo Singleton, es decir, objetos que solo tienen una instancia única y son reutilizados.

- Autenticación y autorización a través de módulos. Brinda la posibilidad de incluir distintas librerías para la gestión de la seguridad en la aplicación.

Entre las tecnologías incluidas en Spring, se pueden citar las siguientes:

1. Spring Boot [10]: Es una herramienta dentro de Spring que simplifica el trabajo al programador ya que facilita la implementación del acceso a los distintos servicios que proporcione la API. Además, permite simular el comportamiento que tendría una aplicación desplegada en un servidor, facilita la implementación de servicios REST y el acceso a los mismos simulando al arrancar el programa que la aplicación está ejecutándose y desplegada sobre un servidor Tomcat.
2. Spring Data [11]: Es una herramienta que facilita el trabajo al programador proporcionándole una interfaz para la persistencia de los datos y una rápida configuración a través de ficheros de propiedades incluidos en la aplicación.
3. Spring Security [12]: Herramienta de la suite de Spring que permite a los programadores securizar la aplicación web a través de JWT (Json Web Token) y proporciona filtros de paso en la entrada de cualquier servicio implementado.

### 3.1.3 Tomcat

Tomcat[10] es un contenedor de aplicaciones con soporte de servlets<sup>2</sup>. Puede funcionar como servidor propio. Se tiene que Spring Boot lleva embebido un servidor de este tipo.

### 3.1.4 JWT

Json Web Token[11] es un estándar abierto basado en JSON para la generación de claves de acceso a las APIs.

El cliente solicita el acceso a un recurso de la API, y si este tiene permisos suficientes, recibe el JWT que contiene la información necesaria para el acceso. El usuario debería almacenar esta clave de alguna forma para enviarla en cada petición.

Siempre que el usuario quiere acceder a una ruta protegida o recurso, el cliente tiene que enviar el JWT, generalmente en el encabezamiento de Authorization.

### 3.1.5 JPA/Hibernate

Java Persistence API[12] es un ORM (Object Relational Mapping) de Java utilizado para la persistencia de los datos. Se encarga de mapear las clases de Java a objetos de base de datos. Estas clases son las conocidas en Java como “POJO” (Plain Old Java Object) y ofrece la posibilidad de construir todo el modelo de datos de la aplicación únicamente cambiando el nombre de una de sus propiedades.

Hibernate es una de las implementaciones de JPA para Java que facilita como se ha comentado el mapeo de objetos a base de datos mediante archivos XML o por anotaciones. Por lo tanto, se puede decir que agiliza la relación entre aplicación y base de datos.

---

<sup>2</sup> Un servlet es una clase en java que se utiliza para ampliar propiedades del servidor

### **3.1.6 Bases de datos MySQL/MariaDB**

MySQL<sup>[13]</sup> es un motor o sistema gestor de base de datos relacional. Contiene licencia pública general y contiene licencia comercial por Oracle. Es de código abierto y uno de los sistemas más utilizados en el mundo.

MariaDB es un sistema de gestión de bases de datos derivado de MySQL. Cuenta con licencia GPL (Generic Public License).

Como servidor de almacenamiento de base de datos se ha utilizado XAMPP que contiene el gestor de base de datos de MySQL, el servidor web Apache y algunas otras utilidades relacionadas con PHP. En la actualidad, XAMPP ha cambiado su base de datos por la de MariaDB.

### **3.1.7 Spring Tools Suite (STS)**

Spring Tools Suite<sup>[14]</sup> es un IDE de programación que se utiliza para programar aplicaciones basadas en Spring con Java. Está derivado de Eclipse, pero es bastante más ligero y viene preparado para levantar fácilmente estas aplicaciones. Contiene un entorno para inyectar otro tipo de servidores que no sea el Tomcat que tiene embebido Spring por defecto y facilita la programación ya que contiene un sistema de búsqueda específica de anotaciones y librerías más sencillo que Eclipse.

STS puede descargarse como un IDE completo o instalarse en versiones de Eclipse.

## **3.2 Herramientas front-end**

### **3.2.1 Android**

Android<sup>[15]</sup> es un sistema operativo que está desarrollado por Google. Se basa principalmente en el kernel de Linux. Está desarrollado principalmente para dispositivos móviles, aunque en la actualidad también está disponible en relojes inteligentes, coches, tabletas...etc.

Para programar en Android el lenguaje que utiliza como base es Java, de ahí la facilidad para pasar de uno a otro. Aunque Android contiene especificaciones propias para las vistas a través de XML, sin embargo, para programar la lógica de las aplicaciones se debe acudir a la sintaxis básica de Java.

En la actualidad se utiliza Kotlin para programar en Android, lenguaje desarrollado por Google para desvincularse de Oracle al pertenecer Java a esta empresa.

### **3.2.2 Retrofit**

Retrofit<sup>[16]</sup> es un cliente o librería de Android que facilita la incorporación de las peticiones REST a las diferentes Apis expuestas en internet. Permite realizar todos los tipos de peticiones existentes GET, PUT, POST, DELETE y PATH, Además, realiza la serialización de los objetos automáticamente, por lo que solo es necesario preocuparse de utilizar el objeto puesto que Retrofit se encargará de transformar obteniéndolo de la API rest. Los manejadores de peticiones son almacenados en interfaces Java en las que lo único que se debe indicar es el tipo de petición a realizar, el endpoint de acceso, y el objeto POJO de respuesta que obtendrá el servicio.

### **3.2.3 Android Studio**

Android Studio<sup>[17]</sup> es un IDE o entorno de desarrollo integrado para Android. Contiene una serie de características entre las que se incluyen un emulador propio para la simulación del dispositivo real, plantillas para diseños, renderizado en tiempo real, visualización rápida de herramientas e integración de los distintos elementos disponibles en el sistema operativo.

## **3.3 Otras herramientas**

### **3.3.1 GitHub**

Como sistema de control de versiones se ha utilizado GitHub<sup>[18]</sup>, que es una plataforma que permite a los desarrolladores trabajar en paralelo y en colaboración. La característica principal de Git es que permite tener múltiples ramas locales totalmente independientes entre sí que posteriormente pueden ser fusionadas, además contiene un entorno gráfico muy intuitivo para la fusión, la recuperación y las distintas operaciones sobre ramas.

### **3.11 Ngrok**

Ngrok<sup>[19]</sup> es una herramienta que permite acceder a una máquina local a cualquier persona en internet con la que compartamos una url generada dinámicamente. Dicho de otra manera, genera un puente entre cliente y servidor, siendo el servidor la máquina en la que se ejecuta esta aplicación.

## CAPÍTULO 4: ESPECIFICACIÓN DE REQUISITOS

En este capítulo se van a describir los casos de uso que se han definido para desarrollar la aplicación. Para ello se han agrupado los casos de uso de acuerdo con el actor que los inicia. Observar que existen algunos casos de uso que son comunes a varios actores, los cuales se presentan aparte.

### 4.1 Actores

En la aplicación se han considerado 3 actores posibles:

1. Profesor: Es aquel que se encarga de gestionar los recursos de aprendizaje que utiliza la aplicación. Así tiene como funciones principales crear, modificar, publicar o eliminar preguntas, repositorios de preguntas y caminos de aprendizaje<sup>3</sup>.
2. Alumno: Es aquel que interactúa con la aplicación para realizar el aprendizaje. Es por ello por lo que su función principal será responder a las preguntas de los caminos de aprendizaje.
3. Administrador: Es aquel que lleva a cabo funciones de mantenimiento de la aplicación.

A continuación, en las siguientes secciones se van a describir los diferentes casos de uso que se han definido para cada actor.

### 4.2 Profesor

En esta sección se van a presentar los casos de uso referidos al actor de tipo profesor, los cuales se encuentran representados en el diagrama de casos de uso de la Figura 1.

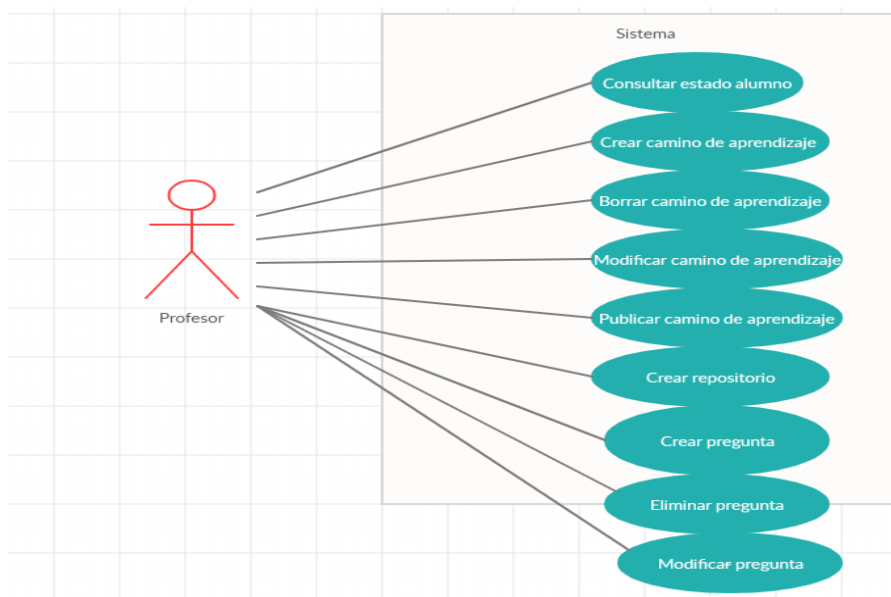


Figura 1: Diagrama de casos de uso del actor Profesor

A continuación, se listan las descripciones de los casos de uso asociados a este actor.

---

<sup>3</sup> Un camino de aprendizaje es una posible secuencia de preguntas que se ha diseñado para un nivel de dificultad dado.

<b>CU-1</b>	<b>CONSULTAR ESTADO ALUMNO</b>							
<b>Objetivos asociados</b>	Consultar el estado de un alumno inscrito							
<b>Entradas</b>	Nick del alumno							
<b>Salidas</b>	Interfaz con los datos del alumno							
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> <li>• Alumno existente en el sistema</li> </ul>							
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre consultar alumnos</td> </tr> <tr> <td>2</td> <td>Introducir Nick del alumno</td> </tr> </tbody> </table>		<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre consultar alumnos	2	Introducir Nick del alumno
<b>PASO</b>	<b>ACCIÓN</b>							
1	Pulsación sobre consultar alumnos							
2	Introducir Nick del alumno							
<b>Postcondición</b>	Salida por pantalla de los datos							
<b>Restricciones</b>	-							
<b>Comentarios</b>	Cuando un profesor consulta el estado de un usuario, aparece información sobre la puntuación que lleva hasta ahora, qué preguntas ha fallado, qué nota ha sacado en cada una de ellas, cuántas veces ha repetido una pregunta, tiempo invertido.							

<b>CU-2</b>	<b>CREAR CAMINO DE APRENDIZAJE</b>							
<b>Objetivos asociados</b>	Crear un camino de aprendizaje con una serie de preguntas (Borrador)							
<b>Entradas</b>	Preguntas asociadas al camino de aprendizaje							
<b>Salidas</b>	Mensaje de éxito							
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> <li>• Camino de aprendizaje no existente en el sistema</li> </ul>							
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre crear C. A</td> </tr> <tr> <td>2</td> <td>Introducir preguntas</td> </tr> </tbody> </table>		<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre crear C. A	2	Introducir preguntas
<b>PASO</b>	<b>ACCIÓN</b>							
1	Pulsación sobre crear C. A							
2	Introducir preguntas							
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito							
<b>Restricciones</b>	-							
<b>Comentarios</b>	El concepto de camino de aprendizaje es el conjunto de preguntas que se le van a realizar a los estudiantes, y los hitos que tendrá un aprendizaje							

<b>CU-3</b>	<b>BORRAR CAMINO DE APRENDIZAJE</b>							
<b>Objetivos asociados</b>	Borrar un camino de aprendizaje del sistema							
<b>Entradas</b>	ID del camino de aprendizaje a borrar							
<b>Salidas</b>	Mensaje de éxito							
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> <li>• Camino de aprendizaje existente en el sistema</li> </ul>							
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre borrar C. A</td> </tr> <tr> <td>2</td> <td>Introducir ID a borrar</td> </tr> </tbody> </table>	<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre borrar C. A	2	Introducir ID a borrar	
<b>PASO</b>	<b>ACCIÓN</b>							
1	Pulsación sobre borrar C. A							
2	Introducir ID a borrar							
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito							
<b>Restricciones</b>	-							
<b>Comentarios</b>	-							

<b>CU-4</b>	<b>PUBLICAR CAMINO DE APRENDIZAJE</b>							
<b>Objetivos asociados</b>	Publicar un camino de aprendizaje en BBDD para ser visible por alumnos							
<b>Entradas</b>	ID del camino a publicar							
<b>Salidas</b>	Mensaje de éxito							
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> <li>• Camino de aprendizaje existente en el sistema</li> </ul>							
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre publicar C. A</td> </tr> <tr> <td>2</td> <td>Introducir ID a publicar</td> </tr> </tbody> </table>	<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre publicar C. A	2	Introducir ID a publicar	
<b>PASO</b>	<b>ACCIÓN</b>							
1	Pulsación sobre publicar C. A							
2	Introducir ID a publicar							
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito							
<b>Restricciones</b>	-							
<b>Comentarios</b>	-							

<b>CU-5</b>	<b>MODIFICAR CAMINO DE APRENDIZAJE</b>	
<b>Objetivos asociados</b>	Modificar un camino de aprendizaje	
<b>Entradas</b>	ID del camino a modificar	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> <li>• Camino de aprendizaje existente en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Pulsación sobre modificar C.A
	2	Introducir ID a modificar
	3	Introducir datos nuevos
	4	Validar
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito	
<b>Restricciones</b>	-	
<b>Comentarios</b>	Se podrá modificar un camino no publicado o en estado borrador	

<b>CU-6</b>	<b>CREAR REPOSITORIO</b>	
<b>Objetivos asociados</b>	Crear en el sistema un repositorio de preguntas	
<b>Entradas</b>	Preguntas a insertar en el sistema con sus respectivas respuestas	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema como profesor</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Pulsación sobre añadir Repositorio
	2	Introducir Preguntas y respuestas
	3	Validar
	4	Mensaje de éxito
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito	
<b>Restricciones</b>	-	
<b>Comentarios</b>	-	

<b>CU-7</b>	<b>CREAR PREGUNTA</b>											
<b>Objetivos asociados</b>	Añadir una pregunta a un repositorio propio del profesor											
<b>Entradas</b>	Pregunta a insertar con sus respuestas											
<b>Salidas</b>	Mensaje de éxito											
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema como profesor</li> </ul>											
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre añadir pregunta</td> </tr> <tr> <td>2</td> <td>Introducir pregunta y respuestas</td> </tr> <tr> <td>3</td> <td>Validar</td> </tr> <tr> <td>4</td> <td>Mensaje de éxito</td> </tr> </tbody> </table>		<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre añadir pregunta	2	Introducir pregunta y respuestas	3	Validar	4	Mensaje de éxito
<b>PASO</b>	<b>ACCIÓN</b>											
1	Pulsación sobre añadir pregunta											
2	Introducir pregunta y respuestas											
3	Validar											
4	Mensaje de éxito											
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito											
<b>Restricciones</b>	-											
<b>Comentarios</b>	=											

<b>CU-8</b>	<b>ELIMINAR PREGUNTA</b>											
<b>Objetivos asociados</b>	Eliminar una pregunta del sistema											
<b>Entradas</b>	ID de la pregunta a modificar											
<b>Salidas</b>	Mensaje de éxito											
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema como profesor</li> <li>• Pregunta existente en el sistema</li> </ul>											
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre eliminar pregunta</td> </tr> <tr> <td>2</td> <td>Introducir ID a eliminar</td> </tr> <tr> <td>3</td> <td>Validar</td> </tr> <tr> <td>4</td> <td>Mensaje de éxito</td> </tr> </tbody> </table>		<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre eliminar pregunta	2	Introducir ID a eliminar	3	Validar	4	Mensaje de éxito
<b>PASO</b>	<b>ACCIÓN</b>											
1	Pulsación sobre eliminar pregunta											
2	Introducir ID a eliminar											
3	Validar											
4	Mensaje de éxito											
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito											
<b>Restricciones</b>	-											
<b>Comentarios</b>	=											

<b>CU-9</b>	<b>MODIFICAR PREGUNTA</b>	
<b>Objetivos asociados</b>	Modificar una pregunta de un repositorio	
<b>Entradas</b>	ID de la pregunta a modificar	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> <li>• Pregunta existente en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Pulsación sobre modificar preg.
	2	Introducir ID a modificar
	3	Introducir datos nuevos
	4	Validar
<b>Postcondición</b>	Salida por pantalla de mensaje de éxito	
<b>Restricciones</b>	-	
<b>Comentarios</b>	-	

### 4.3 Alumno

En esta sección se van a presentar los casos de uso referidos al actor de tipo alumno, los cuales se encuentran representados en el diagrama de la Figura 2.

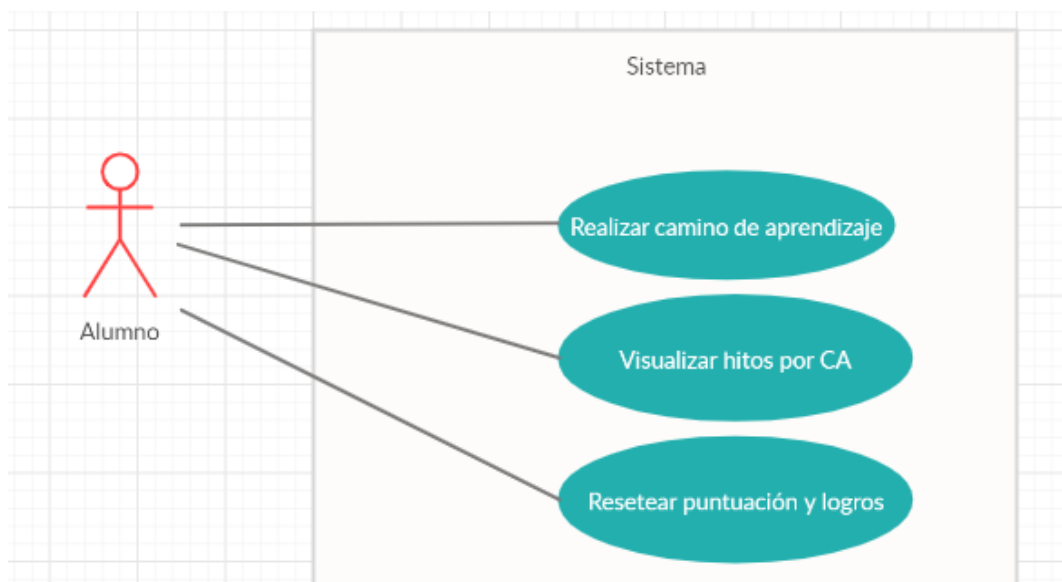


Figura 2: Diagrama de casos de uso del actor Alumno

A continuación, se listan las descripciones de los casos de uso asociados a este actor.

<b>CU-10</b>	<b>REALIZAR C.A</b>	
<b>Objetivos asociados</b>	Realizar una serie de preguntas o camino de aprendizaje publicado por un profesor.	
<b>Entradas</b>	-	
<b>Salidas</b>	Preguntas del camino de aprendizaje	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Usuario autenticado en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Pulsación sobre realizar examen
	2	Introducir respuestas
	3	salida
<b>Postcondición</b>	<ul style="list-style-type: none"> <li>• Examen o camino realizado</li> </ul>	
<b>Restricciones</b>	-	
<b>Comentarios</b>	-	

<b>CU-11</b>	<b>VISUALIZAR HITOS C.A</b>	
<b>Objetivos asociados</b>	Visualización de medallas, hitos...etc. por camino de aprendizaje asociado	
<b>Entradas</b>	-	
<b>Salidas</b>	Interfaz gráfica con los datos	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Usuario autenticado en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Pulsación sobre visualización de hitos
	2	Seleccionar camino de aprendizaje
	3	Mostrado de datos
<b>Postcondición</b>	-	
<b>Restricciones</b>	-	
<b>Comentarios</b>	-	

CU-12	RESETEAR PUNTUACIÓN Y LOGROS									
<b>Objetivos asociados</b>	Reiniciar contador de progreso de juego y estadísticas del usuario									
<b>Entradas</b>	-									
<b>Salidas</b>	Mensaje de éxito									
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Usuario autenticado en el sistema</li> </ul>									
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th>PASO</th> <th>ACCIÓN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre Reiniciar puntuación</td> </tr> <tr> <td>2</td> <td>pregunta de confirmación</td> </tr> <tr> <td>3</td> <td>Mensaje de éxito</td> </tr> </tbody> </table>	PASO	ACCIÓN	1	Pulsación sobre Reiniciar puntuación	2	pregunta de confirmación	3	Mensaje de éxito	
PASO	ACCIÓN									
1	Pulsación sobre Reiniciar puntuación									
2	pregunta de confirmación									
3	Mensaje de éxito									
<b>Postcondición</b>	-									
<b>Restricciones</b>	-									
<b>Comentarios</b>	-									

#### 4.4 Administrador

En esta sección se van a presentar los casos de uso referidos al actor de tipo administrador, los cuales se encuentran representados en el diagrama de la Figura 3.

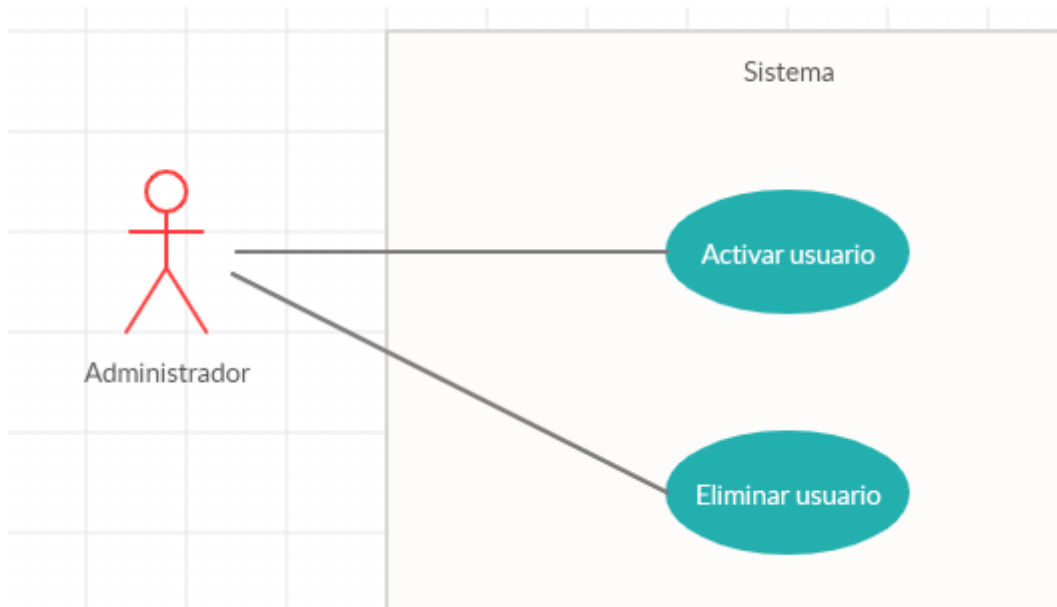


Figura 3: Diagrama de casos de uso del actor administrador

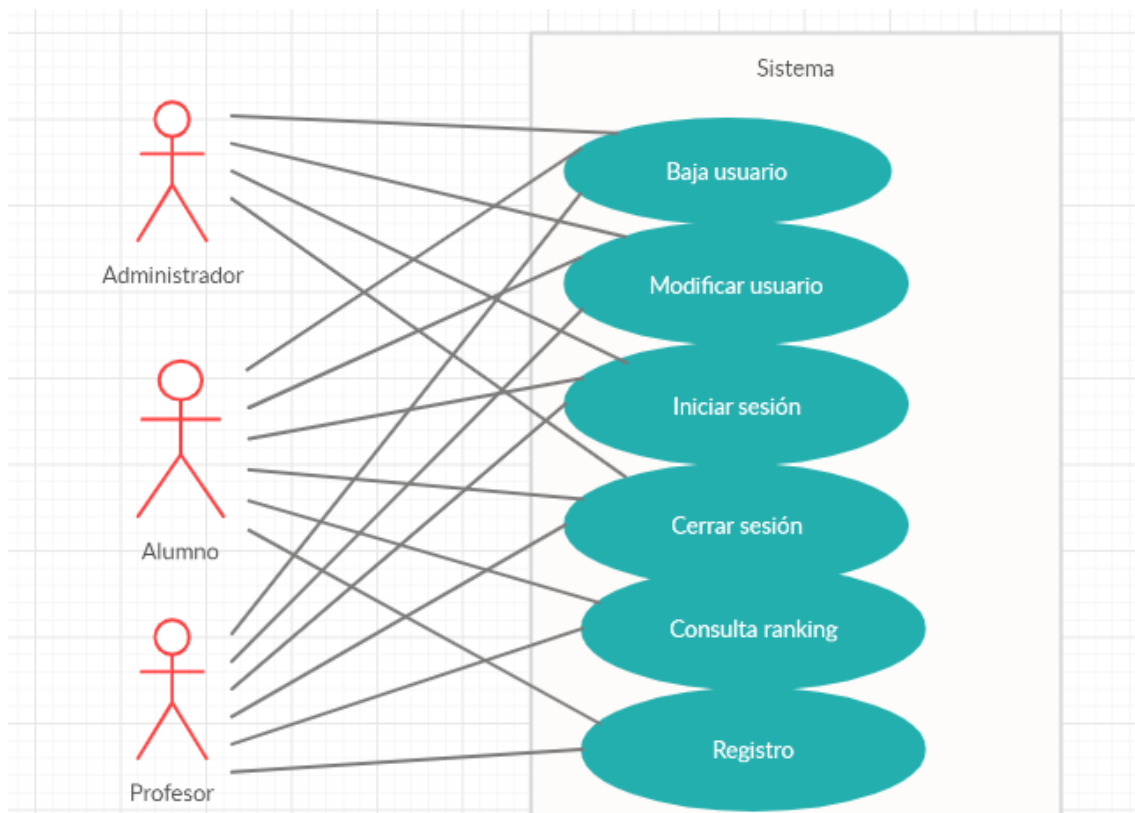
A continuación, se listan las descripciones de los casos de uso asociados a este actor.

<b>CU-13</b>	<b>Activar usuario</b>	
<b>Objetivos asociados</b>	Activar solicitud de alta de usuario en el sistema	
<b>Entradas</b>	-	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>● Estar autenticado como administrador en la app</li> <li>● Usuario no activo en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Entrar en solicitudes
	2	validar
<b>Postcondición</b>	Usuario registrado en el sistema	
<b>Restricciones</b>		
<b>Comentarios</b>	El administrador podrá modificar la solicitud de alta y podrá activar al usuario o bien borrar la solicitud si esta no se ajusta a lo pedido	

<b>CU-14</b>	<b>Eliminar usuarios</b>	
<b>Objetivos asociados</b>	Eliminar usuario del sistema	
<b>Entradas</b>	ID o nick del usuario a eliminar	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>● Estar autenticado como administrador en la app</li> <li>● Usuario activo en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Entrar en Desactivar usuario
	2	Validar y desactivar
<b>Postcondición</b>	Usuario dado de baja en el sistema	
<b>Restricciones</b>	-	
<b>Comentarios</b>	En el caso de que el administrador elimine un profesor, todo el repositorio de preguntas y los caminos de aprendizaje que haya incluido en la app, quedarán eliminados en cascada.	

## 4.5 Casos de uso comunes

En este apartado se describen los casos de uso comunes a los tres actores: profesor, alumno o administrador. (Ver Figura 4)



*Figura 4: Diagrama de casos de uso comunes*

A continuación, se listan los casos de uso mostrados en la Figura 4.

<b>CU-15</b>	<b>Registro</b>									
<b>Objetivos asociados</b>	Registrar en el sistema a un usuario con el rol de profesor o alumno									
<b>Entradas</b>	<ul style="list-style-type: none"> <li>● Nombre</li> <li>● Apellidos</li> <li>● Nick</li> <li>● Fecha de nacimiento</li> <li>● Correo electrónico</li> <li>● Contraseña</li> <li>● Profesor con el que se inscribe (En el caso de ser alumno)</li> </ul>									
<b>Salidas</b>	Mensaje de éxito									
<b>Precondición</b>	<ul style="list-style-type: none"> <li>● Correo electrónico no registrado en el sistema</li> <li>● Nick no registrado en el sistema</li> </ul>									
<b>Secuencia normal</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">PASO</th> <th style="text-align: left;">ACCIÓN</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Seleccionar si es profesor o alumno</td> </tr> <tr> <td>2</td> <td>Introducir datos</td> </tr> <tr> <td>3</td> <td>Validar registro e iniciar app</td> </tr> </tbody> </table>		PASO	ACCIÓN	1	Seleccionar si es profesor o alumno	2	Introducir datos	3	Validar registro e iniciar app
PASO	ACCIÓN									
1	Seleccionar si es profesor o alumno									
2	Introducir datos									
3	Validar registro e iniciar app									
<b>Postcondición</b>	Solicitud de registro enviada al administrador									
<b>Restricciones</b>	<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">PASO</th> <th style="text-align: left;">ACCIÓN</th> </tr> </thead> <tbody> <tr> <td>S-1</td> <td>Algún campo de registro vacío</td> </tr> <tr> <td>S-2</td> <td>Correo e. ya registrado</td> </tr> <tr> <td>S-3</td> <td>Nick ya registrado</td> </tr> </tbody> </table>		PASO	ACCIÓN	S-1	Algún campo de registro vacío	S-2	Correo e. ya registrado	S-3	Nick ya registrado
PASO	ACCIÓN									
S-1	Algún campo de registro vacío									
S-2	Correo e. ya registrado									
S-3	Nick ya registrado									
<b>Comentarios</b>	El registro consiste en rellenar un formulario con los datos del usuario y seleccionando un tipo de usuario: profesor o estudiante. El registro no es inmediato, le llega la petición al administrador y este debe activarlos o borrar la solicitud de registro									

<b>CU-16</b>	<b>BAJA USUARIO</b>	
<b>Objetivos asociados</b>	Dar de baja un usuario en el sistema	
<b>Entradas</b>	<ul style="list-style-type: none"> <li>Nick o email</li> <li>Contraseña</li> </ul>	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>Correo electrónico o Nick registrado en el sistema</li> <li>Sesión iniciada</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Seleccionar opción dar de baja
	2	Introducir datos
	3	Validar baja del usuario y salir
<b>Postcondición</b>	Usuario borrado del sistema	
<b>Restricciones</b>	<b>PASO</b>	<b>ACCIÓN</b>
	S-1	Algún campo de registro vacío
	S-2	Correo e. no registrado
	S-3	Nick no existente
<b>Comentarios</b>		

<b>CU-17</b>	<b>MODIFICAR USUARIO</b>	
<b>Objetivos asociados</b>	Modificar un usuario en el sistema	
<b>Entradas</b>	<ul style="list-style-type: none"> <li>Datos para modificar (nombre, fecha de nacimiento...etc.)</li> </ul>	
<b>Salidas</b>	Mensaje de éxito	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>Sesión iniciada en la app</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Seleccionar opción dar de modificar
	2	Introducir datos
	3	Validar datos
<b>Postcondición</b>	Usuario modificado del sistema	
<b>Restricciones</b>	-	
<b>Comentarios</b>	-	

<b>CU-18</b>	<b>INICIAR SESIÓN</b>									
<b>Objetivos asociados</b>	Iniciar sesión en el sistema									
<b>Entradas</b>	<ul style="list-style-type: none"> <li>• Nick</li> <li>• Contraseña</li> </ul>									
<b>Salidas</b>	-									
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Nick y contraseña existentes en el sistema</li> </ul>									
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Acceder a la app</td> </tr> <tr> <td>2</td> <td>Introducir datos</td> </tr> <tr> <td>3</td> <td>Validar datos y entrar</td> </tr> </tbody> </table>		<b>PASO</b>	<b>ACCIÓN</b>	1	Acceder a la app	2	Introducir datos	3	Validar datos y entrar
<b>PASO</b>	<b>ACCIÓN</b>									
1	Acceder a la app									
2	Introducir datos									
3	Validar datos y entrar									
<b>Postcondición</b>	Acceso del usuario al sistema									
<b>Restricciones</b>	Campos de datos no vacíos									
<b>Comentarios</b>	-									

<b>CU-19</b>	<b>CERRAR SESIÓN</b>							
<b>Objetivos asociados</b>	Cerrar sesión del sistema							
<b>Entradas</b>	-							
<b>Salidas</b>	Mensaje informativo de sesión cerrada							
<b>Precondición</b>	Estar autenticado en el sistema							
<b>Secuencia normal</b>	<table border="1"> <thead> <tr> <th><b>PASO</b></th> <th><b>ACCIÓN</b></th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Pulsación sobre cerrar sesión</td> </tr> <tr> <td>2</td> <td>Salir</td> </tr> </tbody> </table>		<b>PASO</b>	<b>ACCIÓN</b>	1	Pulsación sobre cerrar sesión	2	Salir
<b>PASO</b>	<b>ACCIÓN</b>							
1	Pulsación sobre cerrar sesión							
2	Salir							
<b>Postcondición</b>	Salida del usuario del sistema							
<b>Restricciones</b>	-							
<b>Comentarios</b>	-							

<b>CU-20</b>	<b>CONSULTAR RANKING DE ALUMNOS</b>	
<b>Objetivos asociados</b>	Consultar el ranking con los mejores alumnos	
<b>Entradas</b>	-	
<b>Salidas</b>	Interfaz con el ranking de alumnos	
<b>Precondición</b>	<ul style="list-style-type: none"> <li>• Estar autenticado en el sistema</li> </ul>	
<b>Secuencia normal</b>	<b>PASO</b>	<b>ACCIÓN</b>
	1	Pulsación sobre consultar ranking
	2	Visualización de ranking
<b>Postcondición</b>	Salida por pantalla del ranking	
<b>Restricciones</b>	-	
<b>Comentarios</b>	-	

## CAPITULO 5: MODELO DE DATOS

En este capítulo se va a describir la forma en la que se ha realizado la persistencia de la información del sistema creado.

### 5.1 Modelo E-R

En la figura 5 se muestra el diagrama entidad-relación que se ha diseñado para modelar la información que gestionará la aplicación. En este diagrama se representan las entidades y los atributos definidos para el sistema:

- User, representa la entidad usuario del sistema con su información personal.
- Tema, representa la entidad que hace referencia a un tema determinado de la asignatura.
- Question, representa la entidad pregunta perteneciente a un tema concreto de la asignatura.
- NodoCA, representa un nodo del camino de aprendizaje.
- NodoAlternativo, representa un nodo alternativo del camino de aprendizaje, es un hijo de la entidad NodoCA.
- Logros, representa la entidad que contiene la información de medallas y porcentaje conseguido por los alumnos.
- Answer, representa una entidad con la respuesta a una pregunta de la entidad “Question”.
- Question\_suspended, representa una entidad pregunta la cual se ha fallado en un camino de aprendizaje.
- Estadísticas, representa la entidad que contiene las estadísticas del usuario que realiza un camino de aprendizaje.

Existe una relación entre NodoCA y User, ya que un usuario profesor puede tener varios caminos de aprendizaje, por lo que la relación es de 1 a n. NodoCA tiene una relación de 1 a 1 con Question, ya que cada nodo del camino de aprendizaje tiene asignado una pregunta.

De igual forma se contempla la entidad “Nodo Alternativo” y NodoCA, que están relacionados directamente 1 a 1 ya que, al fallar una pregunta contenida en un nodo del árbol, se mostraría la pregunta asignada al nodo alternativo.

La relación entre User y Tema es de n a n, por lo que para modelarla se extrae una entidad intermedia Logros. Quedando 1 a n la relación entre User y Logros, y 1 a n la relación entre Tema y Logros. Un usuario alumno puede conseguir un nivel o logro en un tema.

La entidad Tema tiene una relación 1 a n con “Question” ya que un tema puede tener varias preguntas asignadas, y a su vez, una pregunta tiene varias respuestas. Es por ello que la entidad Question tiene una relación 1 a n con la entidad Answer.

Por último, se encuentran las entidades “Estadísticas” que representan las estadísticas de un usuario, y que mantiene una relación 1 a n con la entidad User al poder contar un usuario con varias estadísticas. Esta entidad se relaciona con Question\_Suspended en 1 a n ya que las estadísticas tienen una lista de preguntas suspensas, y Estadísticas tendría



## 5.2 Implementación de la base de datos

El modelo E-R definido en la sección anterior, se implementa mediante una base de datos relacional de tipo MariaDB. En la figura 6 se muestra un esquema de las tablas definidas y las relaciones existentes.

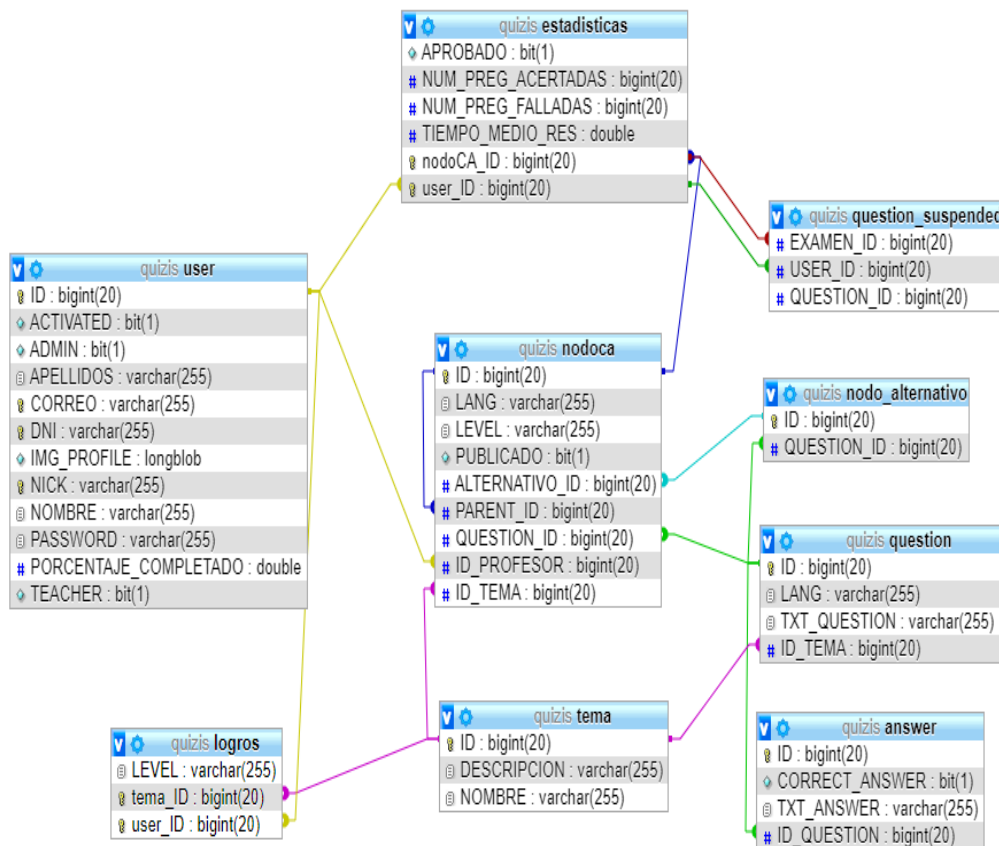


Figura 6: Modelo relacional de la aplicación

A continuación, se explican cada una de las tablas.

### 5.2.1 Tabla User

La tabla User almacena la información del perfil y gestiona los 3 tipos de usuarios presentes en la aplicación.

- Id: identificador del usuario en la tabla
- Nombre: Nombre del usuario
- Apellidos: Apellidos del usuario
- DNI: String que representa el DNI único del usuario
- Nick: Nick elegido por el usuario al crearse la cuenta. Todas las autenticaciones y búsquedas para obtener datos del usuario se realizarán por nick.
- Correo: Correo electrónico del usuario.
- Password: Password para acceso a la cuenta.

- Teacher: Booleano que indica si el usuario que ha accedido a la aplicación es profesor o alumno.
- Activated: Booleano que indica si la cuenta de un usuario profesor está activada o no por el administrador, ya que cualquier persona puede registrarse como profesor, pero será el administrador del sistema el encargado de decidir si tiene acceso en esa cuenta.
- Admin, Booleano que indica a la API si el usuario autenticado es el administrador del sistema.
- Img\_Profile: Archivo blob que representa la imagen de perfil asociada a un usuario.
- Porcentaje\_Completado: Número que indica el porcentaje de juego completado por un alumno.

### 5.2.2 Tabla Tema

La tabla Tema almacena la información de los temas presentes en la aplicación y la descripción de estos.

- Id, identificador del tema en la tabla
- Nombre, Nombre o título del tema
- Descripción: Descripción o subtítulo del tema.

### 5.2.3 Tabla Question

La tabla Question almacena la información del repositorio de preguntas presente en la aplicación.

- Id: Identificador de la pregunta en la tabla.
- Txt\_Question: Enunciado de la pregunta.
- Id\_Tema: Identificar del tema al que está asociada la pregunta. Actúa como clave foránea de la tabla Tema
- Lang: Indica el idioma que se usará para almacenar y recuperar los nodos de los exámenes. Esta propiedad está añadida para dar soporte a varios idiomas en la aplicación.

### 5.2.4 Tabla NodoCA

La tabla NodoCA representa el un nodo del árbol que contiene una pregunta del examen que se mostrará al usuario alumno. Contiene información sobre la pregunta, un puntero al nodo con la pregunta alternativa, puntero al tema que hace referencia la pregunta, y puntero al usuario profesor que ha creado la pregunta.

- Id: Identificador del nodo en la tabla.
- Parent\_Id: Identificador del nodo padre.
- Nivel: Nivel asociado a ese nodo en el camino de aprendizaje (inicial, bronce, plata u oro)
- Publicado: Booleano que indica si ese camino de aprendizaje en el que se encuentra ese nodo está publicado para ser visible al usuario.
- Id\_Tema: Identificador del tema al que hace referencia ese nodo del camino de aprendizaje. Actúa como clave foránea de la tabla Tema.
- Alternativo\_Id: Identificador del nodo alternativo que se mostrará al usuario si se falla la pregunta del NodoCA. Actúa como clave foránea de la tabla NodoAlternativo.

- **Id\_Profesor:** Identificador del usuario al que pertenece el nodo y el camino de aprendizaje. Actúa como clave foránea de la tabla User
- **Question\_Id:** Identificador de la pregunta a la que hace referencia el nodo. Actúa como clave foránea de la tabla Question
- **Lang:** Indica el idioma que se usará para almacenar y recuperar los nodos de los exámenes. Esta propiedad está añadida para dar soporte para varios idiomas en la aplicación

### 5.2.5 Tabla NodoAlternativo

La tabla Nodo alternativo describe la información que representa el “Objeto” definido como pregunta alternativa en el examen, si el alumno falla la pregunta principal. Esta entidad representa un nodo en la estructura de árbol formada para el camino de aprendizaje.

- **Id:** Identificador del nodo alternativo.
- **Question\_Id:** Identificador de la pregunta a la que hace referencia este nodo alternativo. Actúa como clave foránea de la tabla Question.

### 5.2.6 Tabla Logros

En este apartado se describe la información representada en la tabla Logros, que representa el nivel obtenido por un usuario alumno en un determinado tema, por lo que tiene relación con ambas tablas.

- **User\_Id:** Identificador de usuario que actúa como clave foránea de la tabla User
- **Tema\_Id:** Identificador de tema que actúa como clave foránea de la tabla Tema
- **Nivel:** Nivel (inicial, bronce, plata u oro) en el que se encuentra el usuario con id User\_Id para el tema Tema\_Id.

### 5.2.7 Tabla Answer

La tabla Answer representa la información de una respuesta a una determinada pregunta, por lo que contiene un puntero a la pregunta de la que procede.

- **Id:** Identificador de la respuesta en la tabla.
- **Text\_Answer:** Título o descripción de la respuesta.
- **Id\_Question:** Identificador de la pregunta a la que pertenece la respuesta. Actúa como clave foránea de la tabla Question
- **Correct:** Booleano que indica si la respuesta es la correcta para la pregunta a la que hace referencia.

### 5.2.8 Tabla Estadísticas

La tabla Estadísticas contiene la información que ha obtenido un alumno en un determinado examen, es decir, número de respuestas acertadas y falladas, tiempo medio de respuesta...

- **User\_Id:** Identificador del usuario en la tabla usuario. Actúa como clave foránea de la tabla User.

- **Nodoca\_id**, Identificador del examen que ha realizado el alumno. Como la tabla **NodoCA** guarda una estructura de árbol, entonces este identificador almacena el id del nodo padre del camino de aprendizaje.
- **Tiempo\_Medio\_Respuesta**: Tiempo medio que ha tardado el alumno en responder todas las cuestiones del examen.
- **Num\_Preg\_Acertadas**: Número de preguntas que el alumno ha acertado en el examen. Tiene en cuenta tanto las alternativas como las generales.
- **Num\_Preg\_Falladas**: Número de preguntas que el alumno ha fallado en el examen. Tiene en cuenta tanto las alternativas como las generales.
- **Aprobado**: Booleano que indica si el alumno ha aprobado este examen.

### **5.2.9 Tabla Question\_Suspended**

La tabla **Question suspended** contiene la información que representa la información de las preguntas que ha fallado un alumno en un determinado examen, es decir, los ids de estas preguntas.

- **User\_Id**: Identificador del usuario en la tabla **usuario**. Actúa como clave foránea de la tabla **User**.
- **Examen\_id**: Identificador del examen que ha realizado el alumno. Como la tabla **NodoCA** guarda una estructura de árbol, entonces éste identificador almacena el id del nodo padre del camino de aprendizaje.
- **Question\_id**: Identificador de la pregunta que actúa como clave foránea de la tabla **Question**.

# CAPÍTULO 6: ARQUITECTURA DE LA APLICACIÓN

En este capítulo se describe la arquitectura de la aplicación.

## 6.1. Modelo cliente-servidor.

En este sistema se ha utilizado un modelo cliente-servidor, el cual se caracteriza por estar formado por los siguientes elementos:

- Existe un servidor que se encarga de ofrecer servicios a una serie de clientes que los consumen.
- Existe un conjunto de clientes que lanzan peticiones al servidor y esperan una respuesta por parte del servidor

En el servidor se ha implementado una API (interfaz de programación de aplicaciones) de servicios REST<sup>4</sup> que permite acceder a un servicio concreto. Cuando el cliente realiza peticiones a través de HTTP a los servicios que expone la API REST, ésta recupera los datos necesarios de la base de datos, los procesa y devuelve al cliente con la estructura necesaria. El cliente para utilizarlos solo debe conocer el formato y contenido de la respuesta del servicio solicitado. En el esquema de la figura 7 se muestra esquemáticamente la arquitectura del sistema.

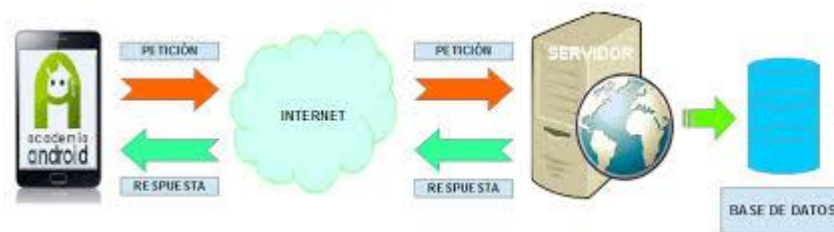


Figura 7: Modelo Cliente-Servidor

## 6.2 Patrones de diseño

En este apartado se explicarán los patrones de diseño utilizados en la implementación del sistema.

### 6.2.1 Capa de cliente

En la capa de cliente se ha utilizado el patrón MVP (Model View Presenter). Es un patrón de diseño que facilita la programación con vistas y mejora el aislamiento entre capas. Sus principales características son las siguientes:

- El modelo de la aplicación contiene las llamadas a los servicios, base de datos u otros datos externos, y realiza la lógica de negocio sobre estos.
- El presentador actúa como mediador entre la vista y el modelo, ya que invoca métodos de ambos y redirige las respuestas del modelo a la vista determinada cargando los atributos necesarios a mostrar en la interfaz.

---

<sup>4</sup> REST es cualquier interfaz entre sistemas que use HTTP para obtener datos o generar operaciones sobre esos datos en algún formato de datos como XML y JSON.

- La vista solo se encarga de mostrar datos y realizar algún tipo de cambio pasivo en las interfaces. Invoca métodos del presentador para saber qué quiere realizar el usuario. Para ello tiene creado un objeto que representa al presentador.

En la figura 8 puede verse una representación de dicho patrón.

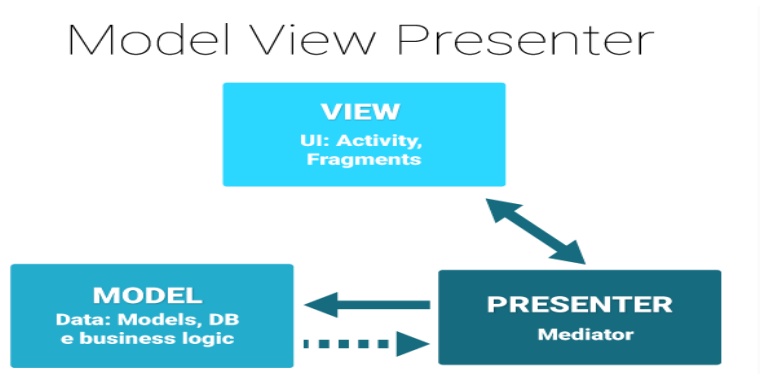


Figura 8: Patrón Model View Presenter

### 6.2.2 Capa de servidor

En la capa de servidor, el patrón utilizado es el MVC o modelo vista controlador, ya que la API REST se ha implementado usando el framework Spring la cual ofrece facilidades para trabajar con este patrón. En Spring, el controlador es el encargado de recibir las peticiones que lanza el cliente y redirigir la llamada al método del modelo (el bean llamado @Service) que trata los datos obtenidos de la BBDD y la lógica de negocio.

El modelo representa los datos o información que maneja la API y realiza toda la lógica de negocio necesaria con estos datos. Posteriormente se devolverán estos datos al controlador que enviará por HTTP en formato JSON al cliente. Todos los Beans de Spring se definen con “@” (el controlador se definiría como @RestController, y el modelo como @Service)

La vista son todos los elementos de la interfaz de usuario (en este caso la aplicación móvil de Android) con los que el usuario interactúa como, por ejemplo: botones, campos de texto...etc.

En la figura 9 puede observarse gráficamente en que consiste el patrón modelo-vista-controlador implementado en Spring.

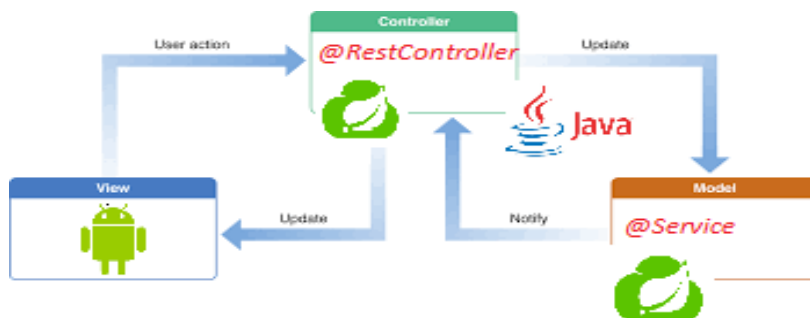


Figura 9: Patrón Modelo-Vista-controlador en Spring



## 7.2 Funcionalidad del cliente

La implementación de la capa de cliente se ha realizado utilizando Android. Entre las razones principales de utilizar este framework de desarrollo se encuentran:

- **Personalización.** Dada su naturaleza libre y abierta, Android permite la mayor personalización de todos los OS que hay en el mercado.
- **Código abierto.**
- **Multitarea.** Android es capaz de gestionar toda la creación de vistas y actividades de una manera sencilla para el programador, todo está delimitado por el ciclo de vida de sus actividades y es fácil seguir el proceso de depuración.
- **Rapidez.** Al estar instalada directamente en el dispositivo, su ejecución es bastante más rápida que si tuviera que precargar e interpretar código en el momento de carga.
- **Diversidad.** La mayoría de las personas hoy en día tienen un smartphone para su uso diario, y cerca del 85% de la población mundial tiene Android como SO en su dispositivo.

En la figura 12 se muestra un diagrama de los módulos funcionales que componen la aplicación. El primer módulo contiene las pantallas de login y registro del usuario y a continuación los 4 últimos módulos representan las 4 pestañas que contiene el menú: Home, Juego, Ayuda y Ranking.

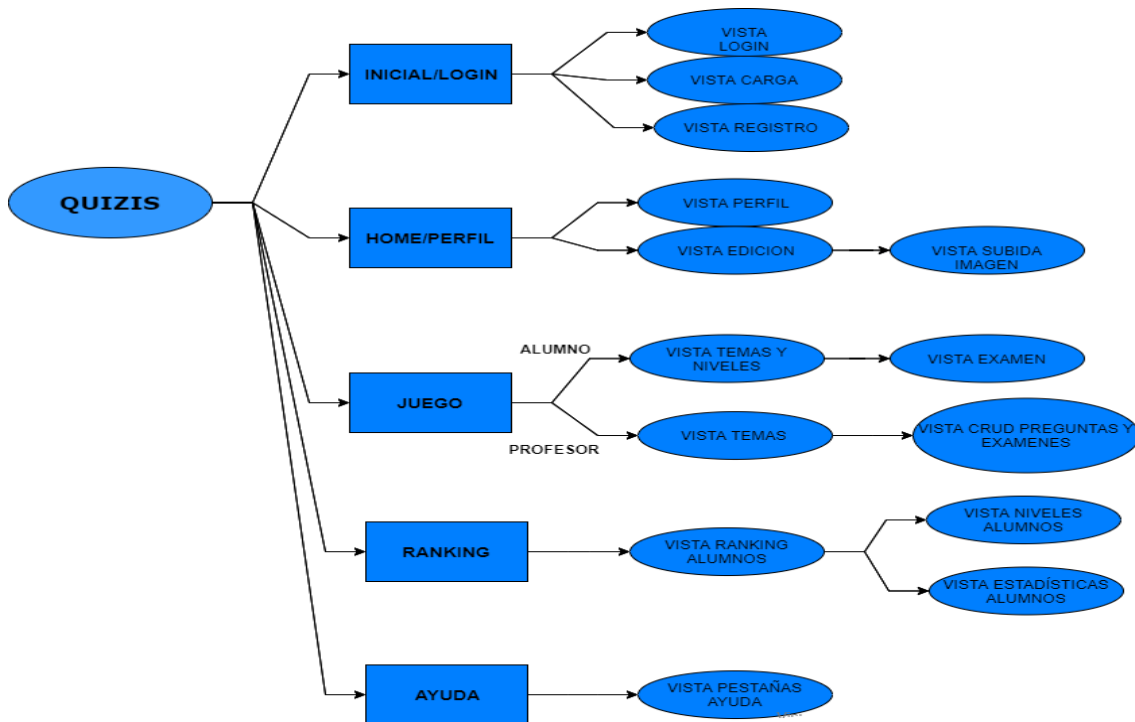


Figura 12: Módulos que componen la aplicación Android

Observar que, para realizar las peticiones, la aplicación tiene para cada módulo una interfaz genérica basada en la librería Retrofit de Android en la que se le incluye la información general sobre cada tipo de petición que se quiera realizar (Tipo de método http, url base, headers, body...etc.). Puede visualizarse en la figura 13 un ejemplo de la interfaz para el módulo usuario que contiene estas peticiones.

```
public interface UserService {

    String API_ROUTE = "/user";

    @GET(API_ROUTE + "{nick}")
    Call<User> getFotoPerfil(@Path("nick") String nick);

    @POST(API_ROUTE + "/picture/{nick}")
    @Multipart
    Call<ResponseBody> uploadImagen(@Path("nick") String nick, @Part MultipartBody.Part image);

    @POST("/login")
    Call<UserDto> doLogin(@Body CredentialsUser user);

    @POST(API_ROUTE)
    Call<ResponseBody> register(@Body User user);

    @PUT(API_ROUTE + "{nick}")
    Call<User> update(@Path("nick")String nick, @Body UserDto user);

    @PUT(API_ROUTE + "/porc/{nick}/{porcentaje}")
    Call<ResponseBody> updatePorc(@Path("nick")String nick, @Path("porcentaje") Double porc);

    @GET(API_ROUTE + "/teachers")
    Call<List<User>> getTeachersNoActivated();

    @GET(API_ROUTE + "/stadistics/{nick}")
```

*Figura 13: Interfaz con algunas peticiones que pueden realizarse a la API sobre usuarios.*

A continuación, se van a describir las funcionalidades implementadas más representativas de la aplicación.

### 7.2.1 Ciclo de vida de una actividad y carga del menú

Todas las actividades o pantallas que pueden ser visualizadas, siguen una estructura similar en su creación y su ciclo de vida pasa por varios estados que pueden verse en la figura 14.

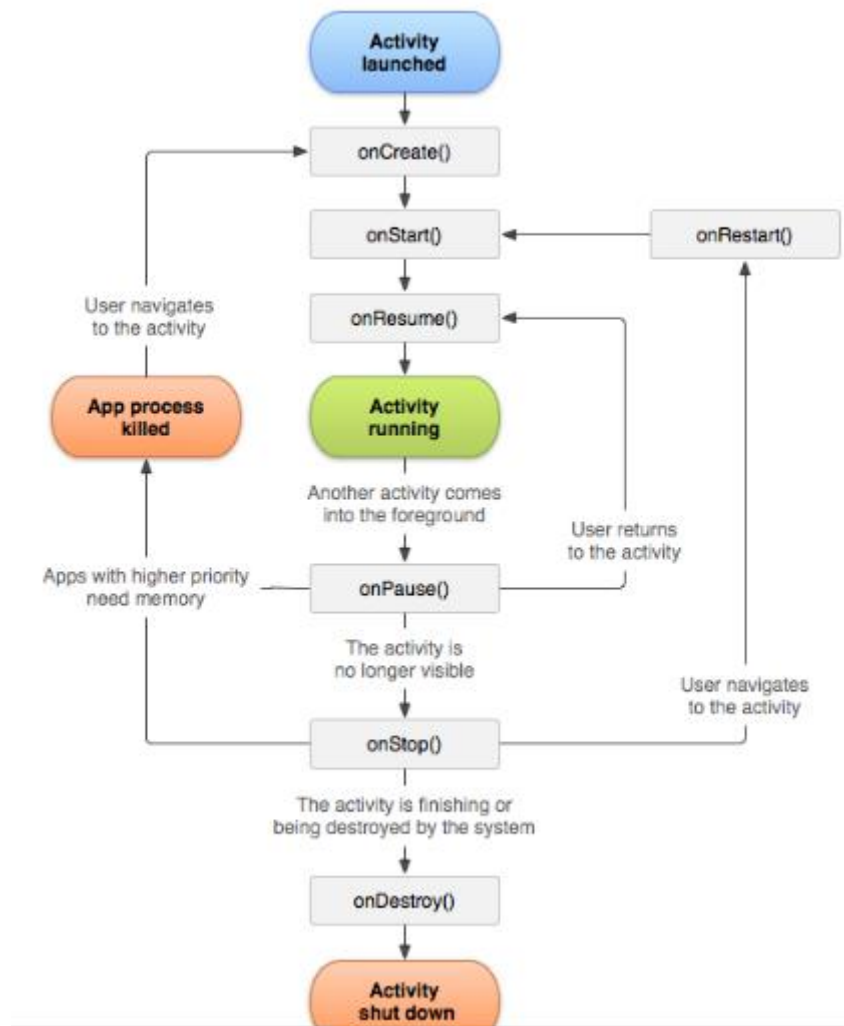


Figura 14: Ciclo de vida de una actividad en Android.

Las actividades de la aplicación se crean en el momento que el programa pasa por `onCreate`. Puede visualizarse la parte de código correspondiente a esta creación en la figura 15, que es similar en todas las actividades.

```

public class LoginActivity extends FormActivity {

    private static final String TAG = "LoginActivity";

    private LoginPresenter loginPresenter;
    private EditText et_nickname, et_password;
    private Button iniciarSesion;
    private TextView tv_link_ayuda;
    private Button registro;
    private String nick;
    private String password;
    private ProgressBar pbMain;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_login);
        btn_sound = MediaPlayer.create(context, R.raw.bump);
        BaseData.setBaseURL(getBaseURL());
        loginPresenter = new LoginPresenter();
        loginPresenter.addView(this);
        //Inicializa los componentes de la pantalla
        initializeView();
        events();
    }

    private void initializeView() {
        et_nickname = findViewById(R.id.nick_login);
        et_password = findViewById(R.id.password_login);
        iniciarSesion = findViewById(R.id.btn_login);
    }
}

```

*Figura 15: Creación inicial de la actividad*

Los pasos que se ejecutan en la mayoría de las actividades al cargarse en la memoria de la aplicación son los siguientes:

1. Se establece la vista correspondiente a la que hará referencia la actividad. En el caso de la figura 15, se instanciará el fichero activity\_login.xml para(R.layout.activity\_login).
2. Se inicializa la variable que hace referencia al sonido que ejecutará la aplicación al pulsar cualquier botón de la vista (btn\_sound).
3. Se inicializa después el presenter correspondiente a la actividad en la que se encuentre la aplicación para poder ejecutar los métodos que a su vez llamarán al modelo encargado de ejecutar la lógica de negocio y lanzar las peticiones a la API REST.
4. Se inicializan los botones, layouts y componentes de la pantalla para poder interactuar con ellos.
5. Finalmente, se invocan los eventos que podrá realizar el usuario para que estos estén activos esperando la interacción de este.

Todas las actividades de la aplicación extienden o heredan de los métodos de una clase padre llamada `FormActivity`, la cual contiene métodos genéricos que se utilizarán a lo largo del ciclo de uso. Se incluyen en esta clase, tres métodos encargados de la creación y visualización del menú general, que será invocado en la carga de cada actividad.

El método llamado “`onOptionsItemSelected`” será invocado con la opción del menú que el usuario quiera visualizar, y se encargará de cargar una nueva actividad si el usuario pulsa en cualquiera de las otras opciones que no estén visibles.

En la figura 16 pueden visualizarse los métodos que se ejecutan para la carga del menú y establecer el aspecto que este tendrá y en la figura 17 se puede ver un ejemplo de cómo muestra la aplicación el menú para el usuario con rol de estudiante, ya que para cada rol se muestra el menú de una forma u otra.

```

@Override
public boolean onCreateOptionsMenu(Menu menu) {
    super.onCreateOptionsMenu(menu);
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.menu, menu);

    //Segundo parametro corresponde con la opción del menú, 0 PERFIL, 1 JUGAR.. Para poner resaltada y en otro caso
    this.assignColorsToItemMenu(menu, this.getActualOption());

    if (getValueBooleanPreferences( clave: "ADMIN" )) {
        menu.findItem(R.id.menu_ranking).setVisible(false);
        menu.findItem(R.id.menu_jugar).setVisible(false);
    }
    if (getValueBooleanPreferences( clave: "TEACHER" ) || getValueBooleanPreferences( clave: "ADMIN" )) {
        menu.findItem(R.id.menu_monedas).setVisible(false);
    }
    return true;
}

protected void setTActionBar() {
    ab = getSupportActionBar();
    ab.setCustomView(R.layout.textview_ab);
    TextView tv = (TextView)ab.getCustomView().findViewById(R.id.tvMenu);
    tv.setText( getValueFloatPreferences( clave: "PORCENTAJE" ).toString());
    //Si el usuario loggeado no es ni profesor ni administrador, se mostrará su porcentaje de juego en la AB
    if (!getValueBooleanPreferences( clave: "TEACHER" ) && !getValueBooleanPreferences( clave: "ADMIN" )) {
        ab.setDisplayShowCustomEnabled(true);
    }
}
}

@Override
public boolean onOptionsItemSelected(MenuItem item) {
    switch (item.getItemId()) {
        case R.id.menu_perfil:
            btn_sound.start();
            if (this.getActualOption() != 1) {
                next( bundle: null, MainActivity.class, destroy: false);
                finish();
            }
            break;
        case R.id.menu_jugar:
            btn_sound.start();
            if (this.getActualOption() != 2) {
                next( bundle: null, ShowTemaActivity.class, destroy: false);
                if (getActualOption() != 1) {
                    finish();
                }
            }
            break;
        case R.id.menu_ranking:
            btn_sound.start();
            if (this.getActualOption() != 3) {
                next( bundle: null, RankingActivity.class, destroy: false);
                if (getActualOption() != 1) {
                    finish();
                }
            }
            break;
        case R.id.menu_ayuda:
            btn_sound.start();
            if (this.getActualOption() != 4) {
                next( bundle: null, HelpActivity.class, destroy: false);
                if (getActualOption() != 1) {
                    finish();
                }
            }
            break;
        default: return true;
    }
    return true;
}
}

```

Figura 16: Métodos de carga del menú en base al rol autenticado

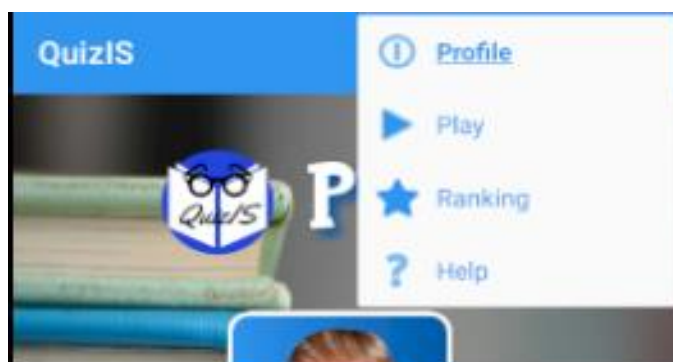


Figura 17: Menú con las distintas opciones disponibles para el usuario

## 7.2.2 Módulo Inicial, Login y Registro

La aplicación se iniciaría con una pantalla de carga o “Splash” en la que se muestra una barra de carga, el título de la aplicación y una frase determinada. La pantalla de login es la primera que muestra la aplicación tras concluir con la pantalla de carga que dura aproximadamente 3 segundos. En este proceso de carga, la aplicación determina el idioma con el que el usuario va a utilizarla, cogiendo por defecto el que tenga configurado en su dispositivo, además de otras configuraciones para la muestra de notificaciones que se verá más adelante. Esta configuración del idioma en la carga inicial puede verse en el fragmento de código de la figura 18.

```
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    // Tenemos una plantilla llamada splash.xml donde mostraremos la información que queramos (logotipo, etc.)
    setContentView(R.layout.splash);
    imagen = (ImageView) findViewById(R.id.iv_splash);
    pb = (ProgressBar) findViewById(R.id.progressBar1);
    tv = (TextView) findViewById(R.id.tv_splash);

    new Handler().postDelayed(() - {
        // Cuando pasen los 3 segundos, pasamos a la actividad principal de la aplicación
        Intent intent = new Intent(packageContext, SplashActivity.this, LoginActivity.class);
        startActivity(intent);
        finish();
    }, DURACION_SPLASH);

    if(Locale.getDefault().getDisplayLanguage().equals("español")) {
        BaseData.setLanguage("esp");
    }
    else if(Locale.getDefault().getDisplayLanguage().equals("English")) {
        BaseData.setLanguage("eng");
    }
}
```

Figura 18: Carga de datos iniciales y configuración del idioma

La aplicación ofrece soporte para inglés y español. Posteriormente será posible incluir otros idiomas de una forma sencilla ya que a través de un fichero de propiedades que contiene todos los textos y mensajes de la aplicación, se puede visualizar en el idioma configurado en el dispositivo. Un ejemplo de estos ficheros con los distintos textos para el idioma español se muestra en la siguiente figura:

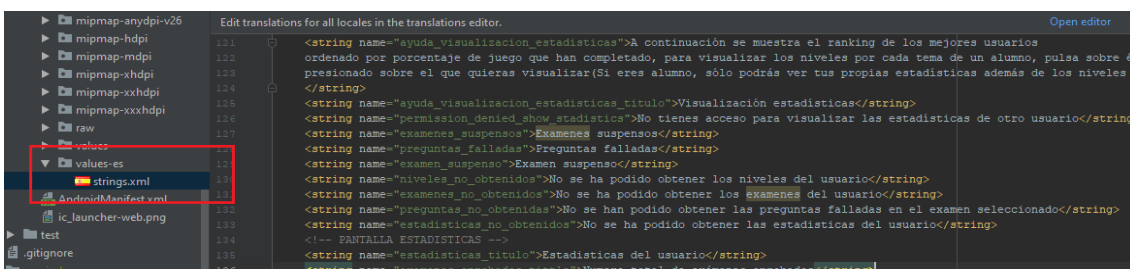
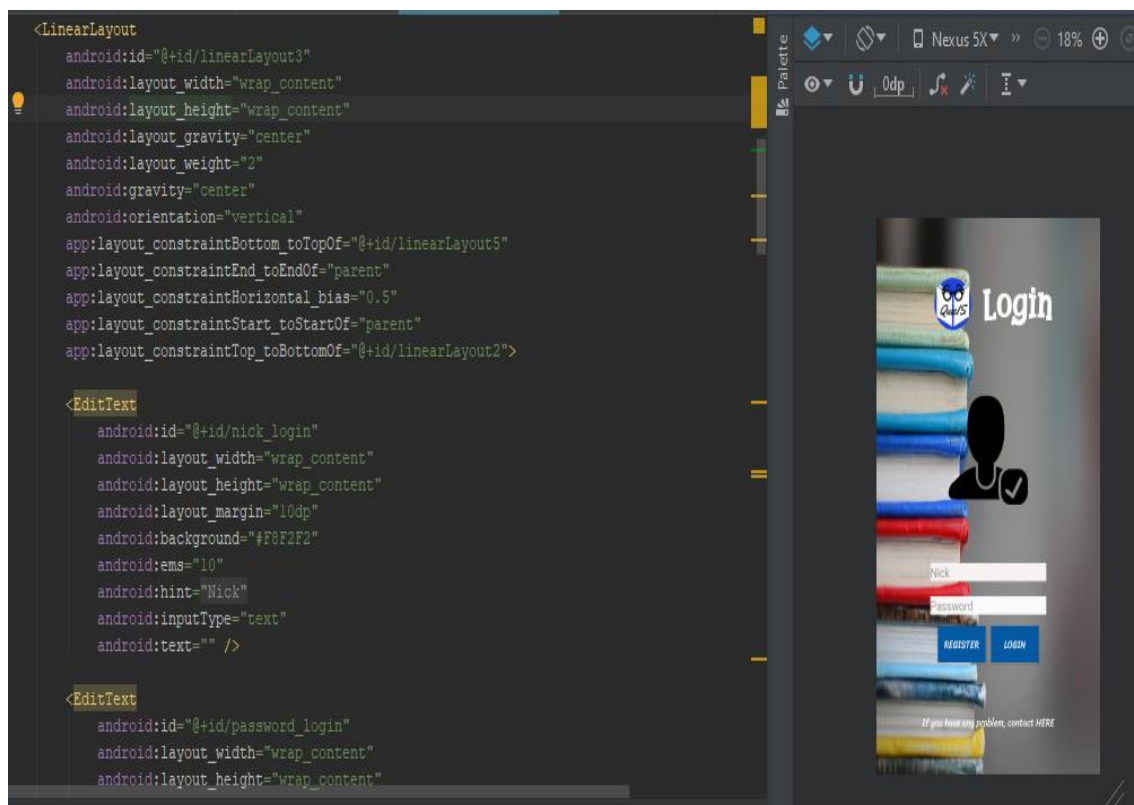


Figura 19: Fichero de propiedades con textos y mensajes para el idioma español.

Para poder utilizar la aplicación más allá de la pantalla de inicio de sesión, es necesario que el usuario esté registrado en el sistema. En esta pantalla se tienen dos botones, uno de ellos estaría destinado a autenticarse y cargar la pantalla del perfil del usuario (siguiente módulo) y el otro botón estaría destinado a un registro del usuario en la aplicación. Esta pantalla sería la denominada “registro” ya que el usuario introduciría

todos los datos solicitados por la aplicación, incluyendo si se quiere registrar como profesor o como alumno, y posteriormente, redirigirá de nuevo a la pantalla de login. Si el usuario se registra como alumno, se podrá iniciar sesión inmediatamente, pero sí indica que es profesor en la pantalla de registro, el usuario debe esperar a que el administrador de la aplicación active la cuenta, ya que el rol de profesor tiene más privilegios y puede visualizar y modificar más contenido que el alumno.

En la figura 20 puede visualizarse la pantalla de login inicial y como estaría estructurado el fichero XML que incorpora los componentes de la pantalla. En la figura 21 puede visualizarse un ejemplo de código de cómo se interactúa ante los eventos del usuario. La pantalla de registro funcionará de forma similar. Cuando se pulsa un botón, se lanza el evento que puede contemplarse en la figura anteriormente citada.



*Figura 20: Fichero XML que incorpora los componentes de la actividad y vista de pantalla*

```

private void events() {
    //Llamar a metodo iniciar sesión (hacer login)
    iniciarSesion.setOnClickListener((v) - {
        btn_sound.start();
        iniciarSesion.setEnabled(false);
        registro.setEnabled(false);
        loginPresenter.doLogin();
    });
    //Hacer registro en el sistema
    registro.setOnClickListener((v) - {
        btn_sound.start();
        next( bundle: null, RegisterActivity.class, destroy: false);
    });

    tv_link_ayuda.setOnClickListener((v) - {
        Intent intent = new Intent(Intent.ACTION_SENDTO); // it's not ACTION_SEND
        intent.putExtra(Intent.EXTRA_SUBJECT, value: "Subject of email");
        intent.putExtra(Intent.EXTRA_TEXT, value: "Body of email");
        intent.setData(Uri.parse("mailto:dsoporte@quizis.com")); // or just "mailto:" for blank
        intent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        startActivity(intent);
    });
}

```

Figura 21: Gestión de eventos en las distintas actividades.

La gestión de la sesión y las credenciales están implementadas basándose en la autenticación del usuario a través de JWT (Json Web Token). El usuario al pulsar el botón de “Login”, lanza una petición a la API REST y obtiene un token de sesión si las credenciales introducidas son válidas. Este token es válido durante un período de tiempo que se quiera establecer en el servidor y se utilizará para realizar todas las peticiones del usuario dentro de la aplicación. El servidor solo permitirá el acceso a las peticiones sin autorización en el header a los servicios de login y registro.

A continuación, se muestra un ejemplo de Token de sesión.

```

"jwt": "Bearer
eyJhbGciOiJIUzUxMiJ9.eyJpYXQiOiJlOTI4MDQ5MzgsIm1zcyI6Imh0dHBzOjI8vd3d3LnF1aXppcy5jb20vIiwic3ViIjoiaWdhcmNpYSIsImV4cCI6MTU4Mjg5MTMzOTM0LjQdDrevXG55aQE30zB0EDEb9EYjNAnH132QG5I0hEZ9Lic6k8AG-D4Fvj2LNE5PgKBXDrB29jstRq35Th2mX8Gg"

```

Figura 22: Ejemplo de Json Web Token o token de sesión del usuario

En todas las peticiones que la aplicación realiza a la API REST, se incluye en el header de forma genérica el JWT obtenido al autenticarse en la misma. Retrofit ofrece la posibilidad de añadir un interceptor que realizará unas determinadas acciones cada vez que se quiera realizar una petición a internet. Se ha implementado este interceptor en la aplicación de forma que introduzca en el header de cualquier petición a la API, el Token de sesión del usuario. Este token de sesión del usuario es cogido de una clase estática llamada “BaseData.class” que contiene variables accesibles desde cualquier punto. De esta forma se podrá comprobar siempre que el usuario está autenticado cuando realice cualquier tipo de petición. La API devolverá la respuesta http Unauthorized 404 cuando

la sesión del usuario haya caducado y de esta forma se cerrará la sesión. El fragmento de código correspondiente al interceptor que añade el token en todos los header de las peticiones lanzadas con Retrofit puede verse en la figura 23.

```
public class ServiceInterceptor implements Interceptor{  
  
    public ServiceInterceptor() {  
    }  
  
    @Override  
    public Response intercept(Interceptor.Chain chain) throws IOException {  
        Request request = chain.request();  
  
        System.out.println("Interceptor");  
  
        if(request.header( name: "No-Authentication") == null){  
            request = request.newBuilder()  
                .addHeader( name: "User-Agent", value: "APP NAME")  
                .addHeader( name: "Authorization", value: "Bearer " + BaseData.getJwtUser())  
                .build();  
        }  
  
        return chain.proceed(request);  
    }  
}
```

*Figura 23: Clase interceptor que añade el token de sesión en todas las peticiones del usuario.*

### **7.2.3 Módulo Home o Perfil.**

Este módulo se centra en la gestión de la información del usuario autenticado. En el caso de ser alumno podrá editar su perfil, cerrar sesión, darse de baja del sistema o reiniciar todos los logros e hitos conseguidos al estado inicial. Si el usuario autenticado es un profesor, podrá realizar lo mencionado anteriormente excepto la parte de reiniciar sus logros. En cambio, si el usuario es el administrador, además de lo anterior, podrá también dar de baja a cualquier usuario introduciendo su Nick y activar cuentas de profesores que soliciten el registro en la aplicación. Para realizar todas las funciones mencionadas anteriormente se dispone de un botón para cada funcionalidad. En toda la aplicación se tiene acceso al menú para navegar por las distintas pestañas que son 4: Perfil, Ranking, Juego y Ayuda. En la figura 24 se puede visualizar como sería la pantalla home del usuario alumno y administrador respectivamente (la del usuario con el rol de profesor es similar a la del alumno sin el botón “RESETEAR PUNTUACION”)

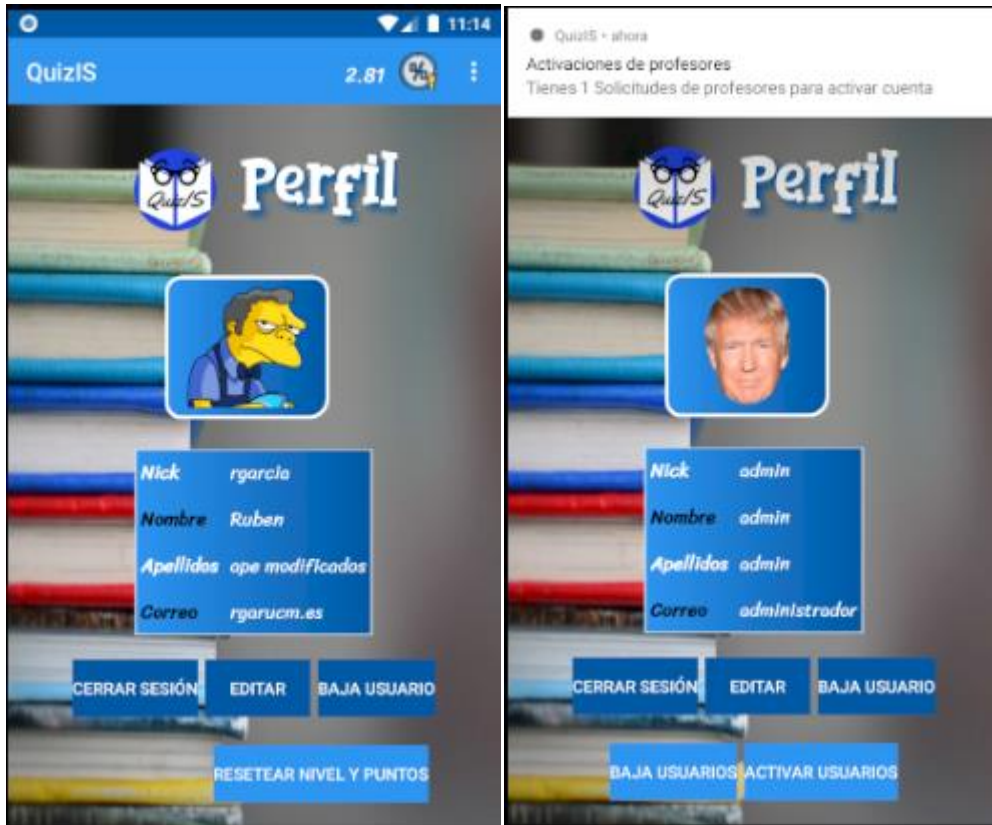


Figura 24: Pantalla inicial de alumno y administrador

Como se puede observar, esta pantalla cuenta con un botón de “ELIMINAR USUARIO”. Este botón al ser pulsado lanzará un evento de la aplicación que mostrará un diálogo al usuario preguntándole si quiere darse de baja de la aplicación. El código correspondiente a este diálogo puede verse en la figura 25.

```
private void showDialogAndSendUserToDelete() {
    AlertDialog.Builder alertDialog = new AlertDialog.Builder( context: MainActivity.this, R.style.AlertDialogCustomForHelp);
    alertDialog.setTitle("Enter the user to delete");
    alertDialog.setMessage("Enter user");

    final EditText input = new EditText( context: MainActivity.this);
    LinearLayout.LayoutParams lp = new LinearLayout.LayoutParams(
        LinearLayout.LayoutParams.MATCH_PARENT,
        LinearLayout.LayoutParams.MATCH_PARENT);
    input.setLayoutParams(lp);
    alertDialog.setView(input);

    alertDialog.setPositiveButton( text: "YES",
        (dialog, which) -> {
            btn_sound.start();
            String nick = input.getText().toString();
            mainPresenter.deleteUser(nick);
        });

    alertDialog.setNegativeButton( text: "NO",
        (dialog, which) -> {
            btn_sound.start();
            dialog.cancel();
        });
    alertDialog.setIcon(R.drawable.img_remove_user);
    alertDialog.show();
}
```

Figura 25: Método que muestra diálogo para darse de baja de la aplicación

Este método tiene un estilo que es común en todos los diálogos de la aplicación, definido en un archivo “.XML” denominado “AlertDialogForCustomHelp”. En la figura 25 puede comprobarse en la primera línea cómo se asigna este estilo al diálogo al hacer “R.style.AlertDialogCustomForHelp”.

Si el usuario autenticado es estudiante o administrador, se tiene un botón que ejecuta la acción determinada para cada tipo de usuario. En el caso de ser alumno, el botón mostrará el texto “Resetear puntos y logros” que se encargará de reiniciar el porcentaje de juego completado del alumno a 0%, y además reiniciará todas sus estadísticas de exámenes suspensos, aprobados, tiempo de respuesta...etc.

En el caso de que el usuario autenticado fuera administrador, primero la aplicación realizará una petición a la API REST para pedir los profesores que han solicitado activar su cuenta y se mostrará una notificación con estas solicitudes. Además, el botón citado mostrará el texto “Activar Usuarios”. El administrador podría seleccionar la notificación o pulsar directamente el botón. Ambos componentes llevarán al administrador a una pantalla donde la aplicación mostrará los profesores que han solicitado activar su cuenta, y el administrador será el encargado de validar esta solicitud para que el profesor pueda autenticarse en la aplicación y publicar exámenes y preguntas.

La acción que ejecutará dicho botón dependiendo del rol del usuario autenticado, se ha implementado como se muestra en la figura 26.

```
next( bundle: null, UpdateUserActivity.class, destroy: false);
});
//Mostrar usuarios pendientes de activación que tiene el administrador
activarUsuario.setOnClickListener((v) - {
    btn_sound.start();
    if (getValueBooleanPreferences( clave: "ADMIN")) {
        if(getListTeachers().size() == 0){
            showToast("You havent pending teachers activation");
        }
        else {
            Intent i = new Intent( packageContext MainActivity.this, ActivationActivity.class);
            i.putExtra( name: "teachers", (Serializable) getListTeachers());
            startActivity(i);
        }
    } else {
        showDialogandResetLevel();
    }
});
//dar de baja al usuario logueado
bajaUsuario.setOnClickListener((v) - {
    btn_sound.start();
    showDialogoAndDeleteUser(getValueStringPreferences( clave: "NICK"));
});
//Dar de baja un usuario introducido en dialog, funcion de administrador
bajaUsuarios_admin.setOnClickListener((v) - {
    btn_sound.start();
    showDialogAndSendUserToDelete();
});
```

Figura 25: Evento del botón que ejecutará una acción u otra por rol autenticado

El código correspondiente encargado de mostrar la notificación al administrador en el proceso de autenticación es el mostrado en la figura 26.

```
private void buildNotification() {
    // Create an explicit intent for an Activity in your app
    Intent notifyIntent = new Intent( packageContext this, ActivationActivity.class);
    notifyIntent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
        | Intent.FLAG_ACTIVITY_CLEAR_TASK);
    notifyIntent.putExtra( name: "teachers", (Serializable) getListTeachers());
    PendingIntent notifyPendingIntent = PendingIntent.getActivity(
        context this, requestCode: 0, notifyIntent, PendingIntent.FLAG_UPDATE_CURRENT
    );

    NotificationCompat.Builder builder = new NotificationCompat.Builder( context this, channelId: "1111")
        .setSmallIcon(R.mipmap.ic_launcher_foreground)
        .setContentTitle("Teacher Activations")
        .setContentText(("You have" + " " + teachersNoactivated.size() + " " + "Teacher requests to activate account")
        .setPriority(NotificationCompat.PRIORITY_HIGH)
        .setContentIntent(notifyPendingIntent)
        .setAutoCancel(true);

    NotificationManagerCompat notificationManager = NotificationManagerCompat.from(this);

    // notificationId is a unique int for each notification that you must define
    notificationManager.notify( id: 1111, builder.build());
}
```

*Figura 26: Código de construcción de notificación de usuarios por activar para el administrador*

Como se puede comprobar, si el usuario autenticado es administrador, al pulsar sobre el botón, se lanzaría el evento que ejecutará una nueva actividad (ActivationActivity), donde se mostrará la lista de solicitudes de profesores por activar. Puede visualizarse esta pantalla en la figura 27.



*Figura 27: Pantalla de activaciones para administrador*

Si el usuario autenticado es alumno, al pulsar sobre el botón mencionado, se mostrará un mensaje de aviso de que va a reiniciar toda su puntuación y el progreso de juego (método “showDialogAndResetLevel” de la figura 25). Puede visualizarse esta información en la figura 28.



*Figura 28: Pulsación sobre reiniciar puntuación autenticado como alumno*

Se puede comprobar cómo se tienen otros dos botones comunes para cualquier rol, que son el de cerrar sesión y editar perfil. El botón de cerrar sesión se encargará de restaurar el fichero “shared\_preferences” de Android, que es un fichero temporal que utiliza la aplicación y contiene información relevante sobre el usuario autenticado para tratarse en tiempo de ejecución. Este fichero, cada vez que el usuario cierre sesión, se vaciará eliminando la información de este con sus preferencias de visualización, y volverá a cargarse con la información del nuevo usuario autenticado. En la figura 29 pueden visualizarse los métodos de carga de información o variables y borrado en el fichero mencionado.

```

public void putUserInSharedPreferences(UserDto user) {
    putStringInSharedPreferences( campo: "NAME", user.getUser().getNombre());
    putStringInSharedPreferences( campo: "APELLIDOS", user.getUser().getApellidos());
    putStringInSharedPreferences( campo: "CORREO", user.getUser().getCorreo());
    putStringInSharedPreferences( campo: "NICK", user.getUser().getNick());
    putBooleanInSharedPreferences( campo: "TEACHER", user.getUser().getTeacher());
    putBooleanInSharedPreferences( campo: "ADMIN", user.getUser().getAdmin());
    putStringInSharedPreferences( campo: "JWT", user.getJwt());
    if(!user.getUser().getTeacher() && !user.getUser().getAdmin()){
        putFloatInSharedPreferences( campo: "PORCENTAJE", user.getUser().getPorcentaje());
    }
}

protected void removeDataPreferences() {
    PreferenceManager.getDefaultSharedPreferences( context: FormActivity.this).
        edit().clear().apply();
}

```

*Figura 29: Métodos de carga y borrado de información del usuario en fichero “shared\_preferences.xml”*

Si se pulsa sobre el botón de cerrar sesión, se mostrará al usuario un diálogo donde deberá decidir si realmente quiere salir de la aplicación. En cambio, si el usuario pulsa sobre el botón de “Editar Perfil”, la aplicación cargará una nueva pantalla de visualización para que el usuario rellene los datos que quiere actualizar de su perfil. Dentro de esta pantalla se tiene también un botón que permite al usuario actualizar su imagen de perfil, si se pulsa sobre él, se llevará al usuario a una nueva pantalla para que seleccione una imagen a subir al servidor y pueda actualizar su perfil. Este contenido puede visualizarse en las imágenes 30, 31 y 32 respectivamente.

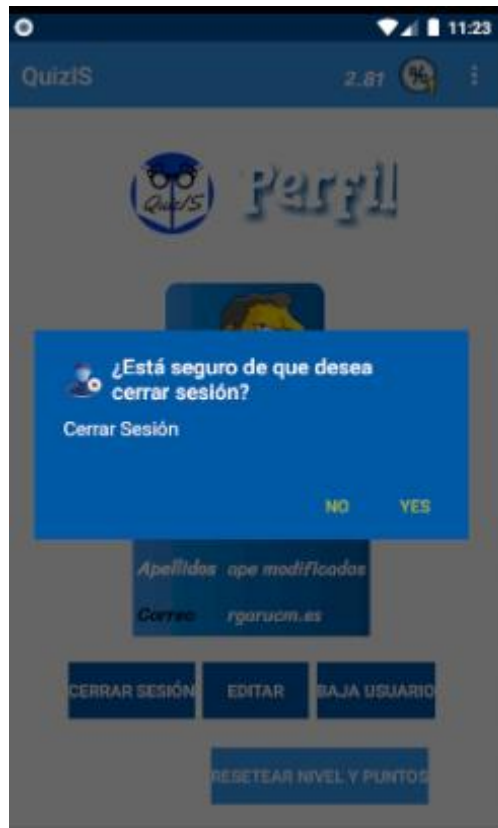


Figura 30: Pantalla al realizar pulsación del usuario sobre cerrar sesión



Figura 31: Pantalla al realizar pulsación del usuario sobre editar perfil



*Figura 32: Pantalla al realizar pulsación del usuario sobre subir imagen (Botón de la figura 31)*

#### **7.2.4 Módulo Ayuda.**

En este módulo o pestaña del menú, se permite al usuario consultar una ayuda para utilizar la aplicación, dependiendo del rol con el que haya accedido. Se mostrarán una serie de ventanas que indican toda la funcionalidad de la aplicación, y lo que se puede ir haciendo en cada pantalla. Si el usuario registrado es profesor se mostrarán unas imágenes de ayuda y si es alumno, otras. Para pasar a la siguiente imagen de ayuda, el usuario dispone de dos botones, uno para avanzar a la siguiente pestaña de ayuda, y otro para volver. En la figura 33 se puede visualizar un ejemplo de la pantalla inicial de ayuda que se le muestra a un alumno, y en la figura 34 se puede visualizar el código correspondiente a la construcción de las imágenes que se introducirán en el componente que muestra la imagen en la respectiva actividad, denominado “imagen\_help”.

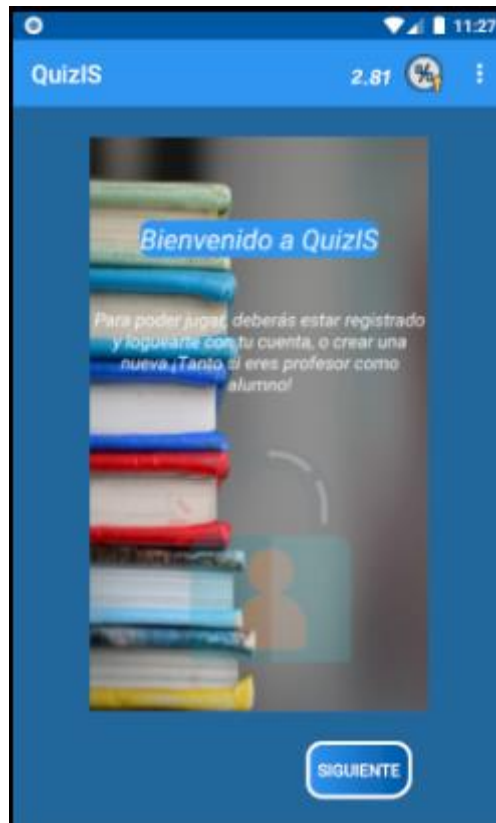


Figura 33: Pantalla inicial de ayuda al usuario

```

private void updateFoto() {
    String mDrawableName = "";
    if (getValueBooleanPreferences( clave: "TEACHER" )) {
        mDrawableName = URL_BASE_PROFESOR + ventanaActual;
    }
    else if (!getValueBooleanPreferences( clave: "ADMIN" )) {
        mDrawableName = URL_BASE_ALUMNO + ventanaActual;
    }

    int resID = getResources().getIdentifier(mDrawableName, defType: "drawable", getPackageName());
    imagen_help.setImageResource(resID);

    if (ventanaActual == numImágenesEnAndroid) {
        btn_ant.setVisibility(View.VISIBLE);
        btn_sig.setText("FINISH");
    }

    if (ventanaActual > 1 && ventanaActual < numImágenesEnAndroid) {
        btn_ant.setVisibility(View.VISIBLE);
        btn_sig.setText("NEXT");
    }

    if (ventanaActual == 1) {
        btn_sig.setText("NEXT");
        btn_ant.setVisibility(View.INVISIBLE);
    }
}

```

Figura 34: Inyección de imagen correspondiente en componente "imagen\_help" y botones de anterior y siguiente.

El método “updateFoto” se encargará de mostrar el botón “Siguiente” y “Anterior” en la ayuda dependiendo de cuantas imágenes de ayuda tenga para cargar la aplicación. Cada vez que el usuario pulsa siguiente, se construye la vista con la imagen y los botones correspondientes hasta que se llega a la imagen final, donde se mostrará sólo el botón de finalizar. Este módulo de la aplicación es, por mencionarse de alguna forma, un tutorial de uso.

### **7.2.5 Módulo Ranking.**

Este módulo se centra en la visualización de información acerca de los estudiantes registrados en la aplicación. La opción del menú correspondiente para mostrar estos datos es “Ranking”, y una vez es seleccionada, se cargará la actividad correspondiente. La inicialización de esta actividad se centra en la obtención de una lista de la API REST que contiene todos los alumnos registrados con la información del porcentaje de juego que ha completado cada uno. Esta petición, que es similar en concepto al resto de peticiones que realiza la aplicación, donde un método presente en el modelo se encarga de invocar la interfaz que contiene el path del servicio que se quiere invocar, y posteriormente a través de otra interfaz, devolver los datos o una respuesta de error. Puede visualizarse en la figura 35 el método encargado de invocar la interfaz con la URL del servicio al que realizar la petición. Tras obtener la lista de los alumnos con su porcentaje de juego, a través de un RecyclerView<sup>5</sup>, la aplicación se encarga de distribuir cada alumno en este contenedor y dibujar sus datos en la vista. Puede visualizarse el componente que corresponde a cada alumno en la figura 36. Todas las vistas que contiene la aplicación en forma de lista están basadas en este componente mencionado, en el que se podrá inyectar cada objeto a visualizar en la lista a través de otro fichero XML. Cada RecyclerView tiene definido un adaptador encargado de inyectar y tratar la lógica de cada componente en esta lista, se puede visualizar una parte del tratamiento de la información del estudiante en el adaptador de “estudiantes” denominado AdapterRVStudents de la aplicación en la figura 37.

---

<sup>5</sup> RecyclerView puede definirse como un contenedor Android que contiene una lista de componentes.

```
public void getStudents(final RankingCallback rankingCallback) {
    Call<List<User>> call = userService.getStudents();

    call.enqueue(new Callback<List<User>>() {
        @Override
        public void onResponse(Call<List<User>> call, Response<List<User>> response) {
            if (response.isSuccessful()) {
                rankingCallback.onReturnListSuccess(response.body());
            } else if (response.code() == 403) {
                rankingCallback.quitAppJwtExpired();
            } else {
                try {
                    rankingCallback.onReturnListError(response.errorBody().string());
                } catch (IOException e) {}
                e.printStackTrace();
            }
        }

        @Override
        public void onFailure(Call<List<User>> call, Throwable t) {
            String json = "Error ";
            try {
                json = new StringBuffer().append(t.getMessage()).toString();
            } catch (NullPointerException e) {}
            rankingCallback.onReturnListError(json);
        }
    });
}
```

Figura 35: Método encargado de realizar petición a la API y de obtener los datos

```
<RelativeLayout
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_marginTop="3dp"
    android:layout_marginLeft="40dp"
    android:layout_marginBottom="7dp"
    android:layout_weight="1">

    <TextView
        android:id="@+id/nombre_student"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:hint="JOSE"
        android:textColor="#0A0A0A"
        android:textSize="18sp"
        android:textStyle="bold"
        android:fontFamily="@font/griracha"/>

    <TextView
        android:id="@+id/apellidos_student"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/nombre_student"
        android:hint="GARCIA LOPEZ"
        android:textColor="#0A0A0A"
        android:textSize="18sp"
        android:textStyle="bold"
        android:fontFamily="@font/griracha"/>
```

Figura 36: Componente con información del usuario inyectado en RecyclerView

```

@Override
public int getItemCount() { return students.size(); }

public class ViewHolderDatos extends RecyclerView.ViewHolder implements View.OnClickListener, View.OnLongClickListener{
    TextView textView1;
    TextView textView2;
    TextView textView3;
    TextView textView4;
    ImageView imagenStudent;

    public ViewHolderDatos(@NonNull View itemView) {
        super(itemView);
        itemView.setOnClickListener(this);
        itemView.setOnLongClickListener(this);
        textView1 = (TextView) itemView.findViewById(R.id.nombre_student);
        textView2 = (TextView) itemView.findViewById(R.id.apellidos_student);
        textView3 = (TextView) itemView.findViewById(R.id.porcentaje_student);
        textView4 = (TextView) itemView.findViewById(R.id.correo_student);
        imagenStudent = (ImageView) itemView.findViewById(R.id.imagenStudent);
    }

    public void asignarDatos(User user) {
        textView1.setText(user.getNombre());
        textView2.setText(user.getApellidos());
        textView3.setText(user.getPorcentaje().toString() + "% de juego completado");
        textView4.setText(user.getCorreo());
        imagenStudent.setImageBitmap(StringToBitMap(user.getFoto()));
    }
}

```

*Figura 37: Parte de la información tratada en el Adapter perteneciente al RecyclerView de estudiantes.*

En la figura 38 puede visualizarse la interfaz gráfica resultante de una lista de ejemplo con varios estudiantes registrados en la aplicación:

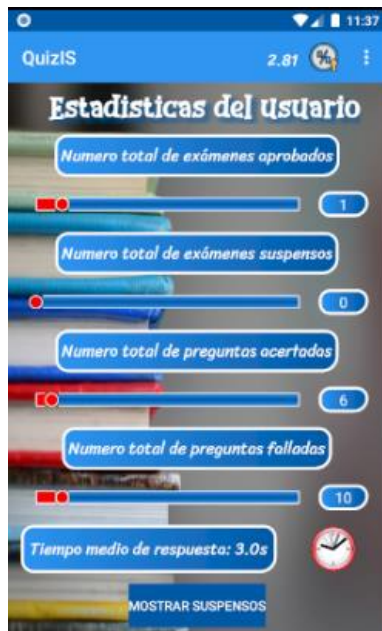


*Figura 38: Pantalla de ranking alumnos*

A partir de este momento, se tienen dos opciones a realizar por el usuario:

1. Pulsación corta sobre un alumno: realizándose una pulsación rápida sobre uno de los alumnos de la lista, la aplicación llevará a una nueva actividad donde podrá visualizarse a través de otro RecyclerView, el nivel o logro del alumno seleccionado por cada tema existente en la aplicación. Esta visualización es accesible para cualquier tipo de rol de usuario autenticado en la aplicación, por lo que todos los alumnos y profesores pueden visualizar los logros obtenidos por todos los alumnos.
2. Pulsación prolongada sobre un alumno: En el caso en el que se deje pulsado sobre un alumno en la lista, se cargará una nueva actividad donde la aplicación realizará una llamada a la API REST pidiendo las estadísticas del alumno sobre el que se ha pulsado. Estas estadísticas se muestran en forma de diagrama de barras y están compuestas por el número de exámenes que ha aprobado o suspendido el alumno, número de preguntas acertadas y falladas, tiempo medio de respuesta en todos los exámenes realizados...etc.

Esta información está restringida únicamente a los profesores registrados en la aplicación y al propio alumno, de tal forma que cualquier profesor podrá visualizar las estadísticas de cualquier alumno, pero sólo el alumno podrá visualizar sus propias estadísticas y no las del resto de compañeros. Dentro de esta pantalla de estadísticas, se contempla un botón que redirige a una nueva carga de actividad para mostrar los exámenes completos que ha suspendido el alumno junto con las preguntas falladas en ese examen ya que anteriormente sólo se podría visualizar un resumen general de las estadísticas. La interfaz gráfica del resumen de las estadísticas puede visualizarse en la figura 39, así mismo, la lógica de código implementada para dar permiso de visualización respondiendo a eventos de click puede visualizarse en la figura 40.



*Figura 39: Pantalla general de estadísticas del alumno*

```

private void events(AdapterRVStudents adaptador) {
    adaptador.setOnItemClickListener(new AdapterRVStudents.ClickListener() {
        @Override
        public void onItemClick(final int position, View v) {
            btn_sound.start();
            Intent i = new Intent( packageContext: RankingActivity.this, RankingExpandedActivity.class);
            i.putExtra( name: "nick", listStudents.get(position).getNick());
            startActivity(i);
        }

        @Override
        public void onItemLongClick(int position, View v) {
            btn_sound.start();
            Intent i = new Intent( packageContext: RankingActivity.this, TeacherRankingStatisticsStudentActivity.class);
            i.putExtra( name: "nickAlumno", listStudents.get(position).getNick());
            //Si eres profesor, tienes acceso a todas las estadísticas de todos los usuarios mostrados en el ranking
            if(getValueBooleanPreferences( clave: "TEACHER")) {
                startActivity(i);
            }
            //Como eres alumno, solo se te permitirá acceder a tus estadísticas
            else{
                //Si el usuario logueado pulsa sobre sus estadísticas...
                if(getValueStringPreferences( clave: "NICK").equals(listStudents.get(position).getNick())){
                    startActivity(i);
                }
            }
            else{
                showToast("You do not have access to view another users statistics");
            }
        }
    }
}

```

*Figura 40: Lógica de código implementada para pulsación sencilla o pulsación prolongada*

## 7.2.6 Módulo Juego.

Este módulo es el encargado de llevar principalmente el control del juego como tal. El alumno tendrá la posibilidad de realizar exámenes para aumentar su progreso de juego y mejorar sus estadísticas, y el profesor podrá añadir, eliminar o modificar exámenes y preguntas.

La pantalla inicial que se presenta para ambos es distinta, en el caso del alumno se muestra una pantalla similar a la mencionada en apartados anteriores de los niveles del alumno, donde se puede visualizar en forma de lista todos los temas disponibles en la aplicación para realizar exámenes junto con el nivel (bronce, plata u oro) en ese tema.

En cambio, para el profesor solamente se muestra la lista con todos los temas disponibles. Puede visualizarse esta interfaz en las figuras 41 y 42 respectivamente.



Figura 41: Pestaña de Juego para el alumno



*Figura 42: Pestaña de Juego para el profesor*

Posteriormente se podrá presionar sobre el tema en concreto del juego y si el usuario registrado es un alumno, la aplicación realizará la carga de la actividad correspondiente al desarrollo del juego para un alumno, que es la de resolver un examen.

Si el usuario es profesor, se cargará otra actividad diferente en la que el usuario podría seleccionar que acción desea realizar a través de unos botones (Crear, modificar o eliminar exámenes y preguntas)

El código donde se decide si se muestra una actividad u otra dependiendo del tipo de usuario que presione sobre el tema en concreto puede visualizarse en la figura 43.

```

private void events(AdapterRVTemas adaptador) {
    adaptador.setOnItemClickListener(new AdapterRVTemas.ClickListener() {
        @Override
        public void onItemClick(int position, View v) {
            btn_sound.start();
            putLongInSharedPreferences( campo: "TEMA", Long.valueOf(position + 1));
            if (getValueBooleanPreferences( clave: "TEACHER")) {
                next( bundle: null, TeacherGameActivity.class, destroy: false);
            } else {
                String nivelDelTemaSeleccionado = niveles.get(position).getNivel();
                if (nivelDelTemaSeleccionado.equals("oro")) {
                    showToast("You can not level up, reset your level if you want to t...");
                } else {
                    Intent i = new Intent( packageContext: ShowTemasActivity.this, StudentGameActivity.class);
                    i.putExtra( name: "NIVEL", nivelDelTemaSeleccionado);
                    showDialogoTemasStudent(i);
                }
            }
        }
    });

    @Override
    public void onItemLongClick(int position, View v) { btn_sound.start(); }
}
}

```

Figura 43: Evento para decisión de lanzar actividad de profesor o alumno en el juego

### 7.2.6.1 Alumno.

Si el usuario que presiona sobre el tema es un alumno, la aplicación mostrará un diálogo donde se le preguntará al alumno si quiere realizar un examen determinado para subir al nivel siguiente al que posee. Puede visualizarse esta interfaz en la figura 44.

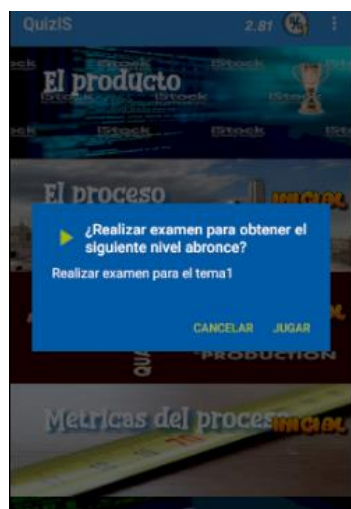
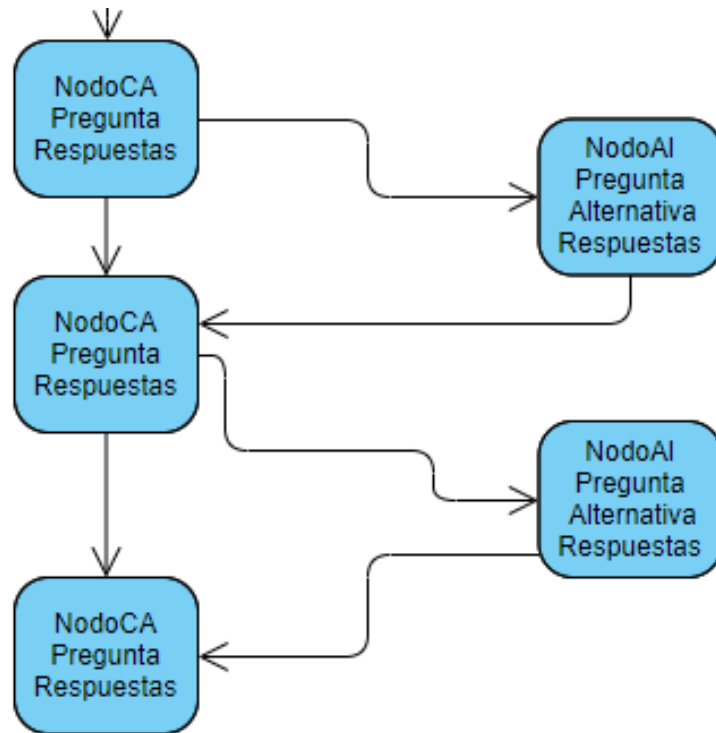


Figura 44: Pulsación del alumno sobre un tema es pestaña juego

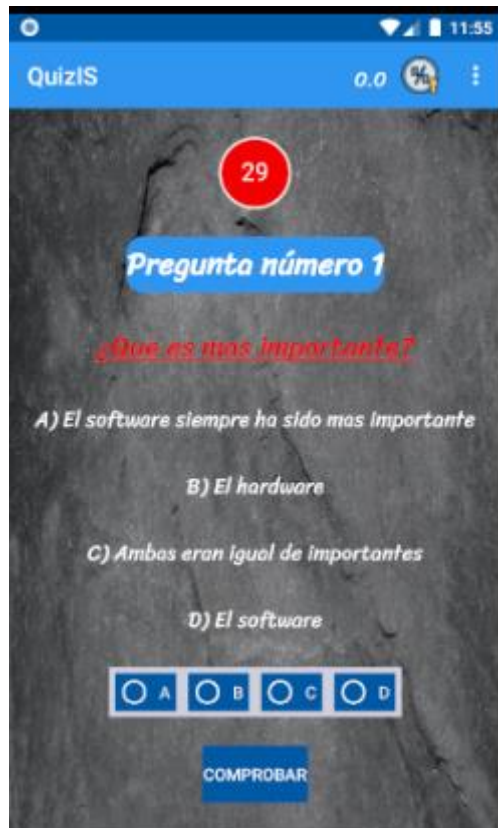
Una vez el usuario ha seleccionado que quiere comenzar su examen, la aplicación carga una nueva actividad realizando una petición a la API para que devuelva un examen o camino de aprendizaje para un alumno que se obtendrá en forma de árbol. Este árbol tiene la estructura que puede visualizarse en la figura 45.



*Figura 45: Estructura del árbol que contiene el examen*

La estructura del árbol está compuesta principalmente por un nodo raíz del que parte el examen que contiene la pregunta inicial y las respuestas. Cada nodo del árbol consta de un objeto que contiene la pregunta, otro con las respuestas y sus dos hijos, que serían el Nodo Alternativo con su pregunta correspondiente y sus respuestas, y el otro el nodo siguiente del examen. Este árbol terminaría en el nodo 10 que únicamente tendría como hijo un nodo alternativo por si se falla la pregunta y terminaría.

La actividad encargada de construir constantemente las vistas necesarias de las preguntas que se le muestran al alumno consta de un temporizador de 30 segundos que se le ofrecen al alumno para responder antes de pasar a la siguiente cuestión, la pregunta, y las 4 opciones de las que deberá elegir una que es la correcta. Puede visualizarse la interfaz gráfica que se mostrará al alumno en la figura 46.



*Figura 46: Pantalla de pregunta de examen para el alumno*

Esta pantalla cuenta con el botón de comprobación que se encarga principalmente de comprobar si la respuesta que ha introducido el alumno es la correcta, y construir la siguiente pregunta, así como reiniciar el contador de nuevo a 30 segundos. Puede construirse una pregunta principal o la alternativa a ésta, que dará una menor puntuación al alumno ya que esta es mostrada si se falla la pregunta principal. El algoritmo de juego lleva un contador para saber la pregunta actual a mostrar, y un booleano que indica si la pregunta a mostrar es alternativa o principal.

El código correspondiente a la construcción de la pregunta en la actividad puede visualizarse en la figura 47 y el correspondiente al chequeo de la respuesta en la figura 48. El diagrama secuencial de desarrollo de juego para un alumno puede visualizarse en la figura 49.

```

private void checkCorrectAndAddPunt() {
    //numeroOfAnswerCorrectInRG Obtiene primero el numero de la respuesta correcta en el examen, si es la "a" obtiene el 1, si es la "b" obtiene
    //getNumberofAnswerCorrectInQuiz Obtiene el numero asociado al radiobutton Seleccionado, si esta seleccionado el primero, obtiene un 1..etc
    if (getNumberofAnswerCorrectInRG() == getNumberofAnswerCorrectInQuiz()) {
        puntuacionTotal += 1;
        numPreguntAcer++;
    } else { //Ha fallado la pregunta y debemos mostrar la alternativa
        showShortToast("You failed!");
        numPregFall++;
        questionFall = new Question();
        questionFall.setId(getNodoActual(preguntaActual).getQuestion().getId());
        questionsSuspended.add(questionFall);
        alternativa = true;
    }
}

private void checkCorrectAlternativeAndAddPunt() {
    if (getNumberofAnswerCorrectInRG() == getNumberofAnswerCorrectInQuizAlternative()) {
        puntuacionTotal += 0.5;
        numPreguntAcer++;
    } else {
        questionFall = new Question();
        questionFall.setId(getNodoActual(preguntaActual).getAlternativo().getQuestion().getId());
        questionsSuspended.add(questionFall);
        numPregFall++;
    }
}
}

```

*Figura 47: Construcción de la pregunta que será mostrada al usuario (Alternativa o principal)*

```

private void buildQuestionAct() { //Si alternativa viene a true, debemos mostrar la pregunta alternativa
    NodeCA nodo = getNodoActual(preguntaActual);
    int preguntaAct = this.preguntaActual + 1;
    if (!alternativa) {
        tituloPreg.setText("Question number" + " " + preguntaAct);
        preg.setText(nodo.getQuestion().getTextQuestion());
        res1.setText(nodo.getQuestion().getAnswers().get(0).getTextAnswer());
        res2.setText(nodo.getQuestion().getAnswers().get(1).getTextAnswer());
        res3.setText(nodo.getQuestion().getAnswers().get(2).getTextAnswer());
        res4.setText(nodo.getQuestion().getAnswers().get(3).getTextAnswer());
    } else {
        tituloPreg.setText("Question number" + " " + preguntaAct + "(Alt)");
        preg.setText(nodo.getAlternativo().getQuestion().getTextQuestion());
        res1.setText(nodo.getAlternativo().getQuestion().getAnswers().get(0).getTextAnswer());
        res2.setText(nodo.getAlternativo().getQuestion().getAnswers().get(1).getTextAnswer());
        res3.setText(nodo.getAlternativo().getQuestion().getAnswers().get(2).getTextAnswer());
        res4.setText(nodo.getAlternativo().getQuestion().getAnswers().get(3).getTextAnswer());
    }
    if (preguntaActual == getNpreguntasExamen() - 1) {
        siguiente.setText("FINISH");
    }
    sound_clock.start();
}
}

```

*Figura 48: Métodos de chequeo de respuesta correcta alternativa y principal*

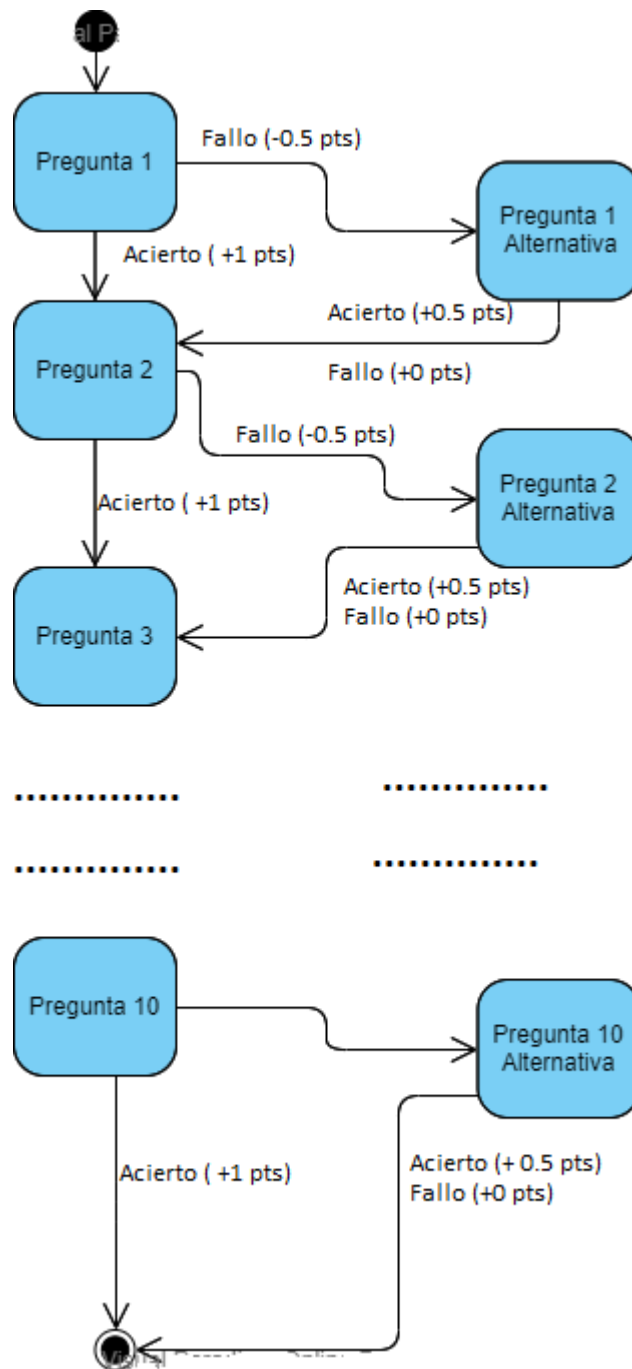


Figura 49: Diagrama de la secuencia de desarrollo del juego

### 7.2.5.2 Profesor.

En el caso en el que el usuario autenticado fuera un profesor, al realizar la pulsación sobre un tema, la aplicación cargará la actividad que contiene los botones para crear, modificar y actualizar exámenes y preguntas en el repositorio. Puede visualizarse esta interfaz en la figura 50.



*Figura 50: Pantalla de juego del profesor al pulsar sobre un tema*

Si el usuario pulsa sobre el botón de “Crear Pregunta”, se cargará una nueva actividad donde se pedirá al usuario que introduzca la pregunta a guardar, las respuestas, y que indique cual es la correcta. El fichero correspondiente con una parte del código de los componentes que contiene esta vista puede visualizarse en la figura 51.

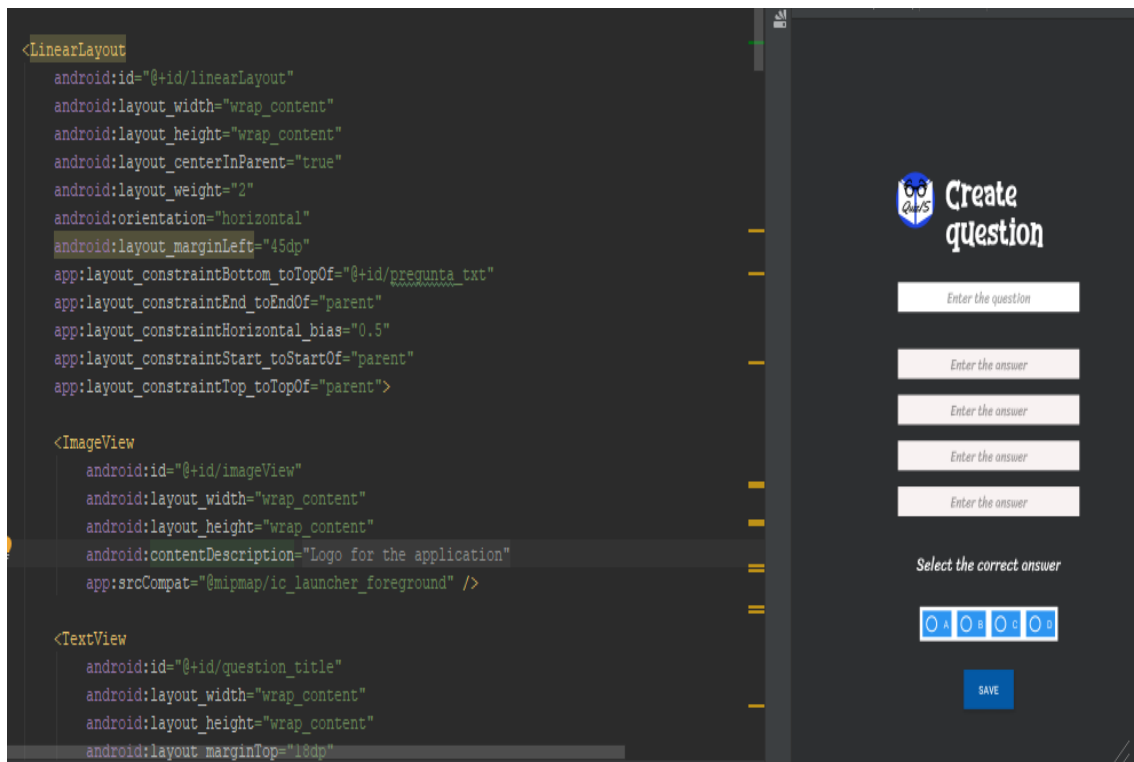


Figura 51: Código con los componentes para guardar una pregunta

Cuando el usuario ha rellenado todos estos campos y pulsa sobre el botón “Guardar”, la aplicación construye el objeto pregunta correspondiente con sus respuestas y enviará a la API REST la petición (similar a lo que se ha explicado en secciones anteriores) para almacenar estos datos y poder utilizarlos posteriormente.

El código de construcción de la pregunta y respuestas tras la pulsación en el botón de guardado puede visualizarse en la figura 52.

```

public void inicializaDatosQuestion() {
    question = new Question();
    Tema tema = new Tema();
    tema.setId(getValueLongPreferences( clave: "TEMA"));
    question.setTema(tema);
    question.setTextQuestion(pregunta.getText().toString());
    question.setLanguage(BaseData.getLanguage());
    //Construye las respuestas y las setea a la pregunta
    question.setAnswers(buildAllAnswers());
}

private List<Answer> buildAllAnswers() {
    List<Answer> answers = new ArrayList<>();
    answers.add(buildAnswer(respuesta1, (RadioButton) findViewById(R.id.rb_a)));
    answers.add(buildAnswer(respuesta2, (RadioButton) findViewById(R.id.rb_b)));
    answers.add(buildAnswer(respuesta3, (RadioButton) findViewById(R.id.rb_c)));
    answers.add(buildAnswer(respuesta4, (RadioButton) findViewById(R.id.rb_d)));
    return answers;
}

```

*Figura 52: Construcción del objeto pregunta con sus respuestas*

Para la implementación de la actualización de pregunta se ha utilizado algo muy similar al guardado.

Para borrar una pregunta del repositorio, cuando el usuario pulsa sobre el determinado botón, esta carga una nueva actividad donde se le muestra al usuario una lista con todas las preguntas que contiene el repositorio hasta el momento. Este proceso de creación y visualización de la lista de preguntas es similar al que se implementó con el componente RecyclerView y Adaptadores del apartado 7.2.4 del módulo Ranking. Si el usuario intenta eliminar una pregunta que está siendo usada en un examen por otro profesor o por él mismo, la API lanzará una excepción con un mensaje determinado que será capturado por la aplicación y no dejará eliminar esta pregunta, mostrando un Toast de error (Figura 53). El código correspondiente a la captura del mensaje en respuesta de la API cuando la pregunta está siendo utilizada puede verse en la figura 54.



Figura 53: Toast al intentar eliminar una pregunta en uso

```

public void dropQuestion(Long id, final QuestionCallback questionCallback) {
    Call<ResponseBody> call = questionService.deleteQuestion(id);

    call.enqueue(new Callback<ResponseBody>() {
        @Override
        public void onResponse(Call<ResponseBody> call, Response<ResponseBody> response) {
            if (response.isSuccessful()) {
                try {
                    questionCallback.onQuestionDeleteSuccess(response.body().string());
                } catch (IOException e) {
                    questionCallback.onQuestionDeleteError(e.getMessage());
                }
            } else if (response.code() == 403) {
                questionCallback.quitAppJwtExpired();
            } else {
                try {
                    questionCallback.onQuestionDeleteError(response.errorBody().string());
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    });
}

@Override
public void onFailure(Call<ResponseBody> call, Throwable t) {
    String json = "Error ";
    try {
        json = new StringBuffer().append(t.getMessage()).toString();
    } catch (NullPointerException e) {
    }
}

```

Figura 54: Código que lanza petición a la API para borrar una pregunta y captura mensaje de error redirigiendo el mensaje al método “OnQuestionDeleteError” de la interfaz correspondiente.

Si el usuario quiere, puede añadir varias preguntas en la misma interfaz al repositorio pulsando en el botón de crear repositorio, donde se le pide que introduzca el número de preguntas que desea guardar y se le mostrará en cada pregunta una interfaz similar a la del guardado de pregunta menciona anteriormente.

En la parte de caminos de aprendizaje o exámenes, el usuario con rol de profesor puede realizar las mismas acciones que sobre la pregunta y, además, podrá publicar un camino de aprendizaje. Un camino de aprendizaje puede crearse, pero cuando este se crea por primera vez, se guarda en base de datos como no publicado, por lo que no será visible aún de manera que puedan realizarlo los alumnos y pueda ser modificado las veces que se quiera hasta que se publique. Para ello, el usuario dispone del botón “PublicarCA” en el que se mostrarán los exámenes que el profesor autenticado haya creado, pero no tenga publicados. Para mostrar estos exámenes la aplicación realiza nuevamente una petición a la API REST a través de la interfaz concreta y mostrará los datos a través de una lista que contiene un resumen del examen, y si nuevamente se pulsa sobre este resumen, ampliará en una nueva ventana con su actividad correspondiente todas las preguntas del examen para que pueda visualizarse completamente. Las ventanas correspondientes pueden visualizarse en las figuras 55 y 56.

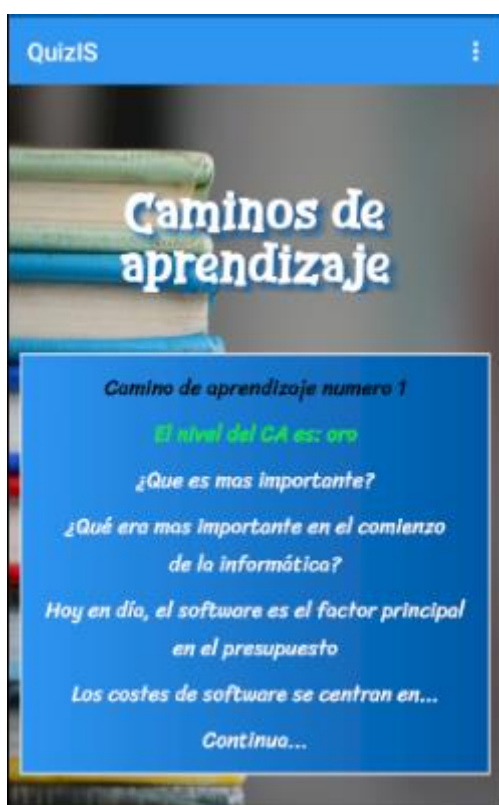


Figura 55: Resumen de caminos de aprendizaje no publicados



Figura 56: Camino de aprendizaje expandido

Si el usuario pulsa sobre el botón de “CrearCA” quiere decir que desea crear un examen. En este caso, la aplicación cargará la actividad correspondiente que se encarga de realizar una petición a la API para obtener nuevamente todas las preguntas almacenadas en el repositorio para el tema que ha seleccionado el usuario anteriormente. Estas preguntas se muestran en forma de lista y el profesor debe seleccionar una a una las preguntas que quiere ir incluyendo en su examen. Primero se le solicita la pregunta principal, está se iluminará de color verde, y posteriormente se le solicita la pregunta alternativa a esta. Una vez ha seleccionado estas dos, se elimina la pregunta inicial de la lista que tiene con el repositorio de preguntas para no incluir la misma dos veces en las principales. En el momento en el que el usuario va seleccionando las preguntas del examen, se va construyendo el árbol correspondiente para poder ser enviado a la API una vez se termine de crear. El código correspondiente a la construcción del árbol puede visualizarse en la figura 57.

```

public void onItemClick(int position, View v) {
    btn_sound.start();
    //Si no es actualizar ni es guardar, queremos borrar la pregunta
    if (!getIntent().getExtras().getBoolean( key: "update") && !getIntent().getExtras().getBoolean( key: "saveCA")) {
        showDialogoAndDeleteQuestion(listQuestions.get(position).getId());
    } else if (getIntent().getExtras().getBoolean( key: "saveCA")) {
        //Si entramos aqui es porque queremos guardar un CA
        if (pulsaciones == 0) {
            questionselected = listQuestions.get(position);
            posicionQuestion = position;
            View elemento = recycler.findViewHolderForAdapterPosition(posicionQuestion).itemView;
            elemento.setBackgroundColor(getResources().getColor(R.color.green));
            adaptador.setPos(posicionQuestion);
            pulsaciones++;
        } else {
            alter = new NodoAlternativo();
            if(questionselected.getId() == listQuestions.get(position).getId()){
                showToast("You cannot select this question as random");
            }
            else {
                alter.setQuestion(listQuestions.get(position));
                checkEndTree(questionselected, alter);
                showShortToast("Question " + contPreguntasAñadidas + " added to exam");
                pulsaciones = 01;
                deleteQuestionFromArray(questionselected);
                adaptador.setPos(55555);
                writeLV(listQuestions);
            }
        }
    }
}

```

Figura 57: Eventos para la construcción del árbol que contiene el examen a guardar

En código puede observarse que se espera a que el usuario pulse sobre un elemento de la lista de preguntas. Una vez se ha pulsado, esta pregunta se almacena en un objeto de este tipo. Y cuando se pulsa la siguiente vez es cuando el método redirige a otro denominado “checkEndTree” que es el encargado de construir el nodo actual del árbol con la pregunta seleccionada y la alternativa, además de comprobar si se ha terminado de crear el examen. También se lleva un contador que indica el número de pulsaciones del usuario para saber si está añadiendo una pregunta alternativa o principal. Una vez se ha construido el nodo correspondiente y chequeado si se ha terminado de crear el examen, se ejecuta primero el método que elimina la pregunta de la lista de preguntas actual(deleteQuestionFromArray) y posteriormente se ejecuta el método writeLV, que vuelve a dibujar la lista de preguntas, pero esta vez, sin la que se ha añadido al examen.

El código correspondiente al método de construcción del nodo del árbol junto con el chequeo de si ha terminado de crear el examen y la representación de las preguntas en la lista puede visualizarse en la figura 58 y 59 respectivamente.

```

private void checkEndTree(Question question, NodoAlternativo alter) {
    if (contPreguntasAñadidas < 10) {
        contPreguntasAñadidas++;
        node.setAlternativo(alter);
        node.setNivel(nivelSeleccionado);
        node.setPublicado(false);
        node.setQuestion(question);
        node.setNickTeacher(getValueStringPreferences( clave: "NICK"));
        node.setTema(getLongPreferences( key: "TEMA"));
        node.setLanguage(BaseData.getLanguage());
        nodes.add(node);
        node = new NodeCA();
    }
    if (contPreguntasAñadidas == 10) {
        //Esto es para preveer que haya toques en pantalla muy rapidos y se seleccionen mas preguntas
        //de las 10 que hay, fallaría la aplicación
        getWindow().setFlags(WindowManager.LayoutParams.FLAG_NOT_TOUCHABLE,
            WindowManager.LayoutParams.FLAG_NOT_TOUCHABLE);
        teacherCAPresenter.saveCA(buildTree());
    }
}
}

```

Figura 58: Construcción del nodo actual con la pregunta principal y alternativa, y construcción del árbol completo para su posterior envío a la API.

```

public void writeLV(List<Question> questions) {
    if(questions.size() == 0){
        showDialogQuestionsEmpty("Not exists questions", "Do not you feel sad to see that there is no question cr...");
    }
    else if (getIntent().getExtras().getBoolean( key: "saveCA") && this.numPreguntasRepositorio < 11){
        showToast("The repository havent ll question or more");
        finish();
    }
    else {
        recycler = findViewById(R.id.recicler_view_ranking);
        recycler.setLayoutManager(new LinearLayoutManager( context: this, LinearLayoutManager.VERTICAL, reverseLayout: false));
        listQuestions = new ArrayList<>();
        this.setQuestionList(questions);
        AdapterRVQuestions adaptador = new AdapterRVQuestions(questions);
        recycler.setAdapter(adaptador);
        events(adaptador);
    }
}
}

```

Figura 59: Método de dibujado en la vista de preguntas.

Para la modificación de un examen, se debe pulsar sobre el botón de modificar y la aplicación realizará dos peticiones a la API, una en la primera actividad para obtener los exámenes del profesor que está autenticado y no han sido publicados aún (ya que una vez que el examen ha sido publicado no puede modificarse), y otra en la actividad siguiente para obtener todas las preguntas del repositorio. De los exámenes obtenidos, se seleccionará para modificar y pasar a la siguiente actividad el que el usuario haya pulsado.

El código que realiza las dos peticiones al comienzo de la actividad puede visualizarse en las figuras 60 y 61 respectivamente

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_ranking); //Reutilizo la view del xml de lista de alumnos en ranking
    textTitulo = ((TextView) findViewById(R.id.titulo_lv_ranking));
    textTitulo.setText("Exams");
    this.setSupportActionBar();
    btn_sound = MediaPlayer.create( context: this, R.raw.bump);
    teacherCAPresenter = new TeacherCAPresenter();
    teacherCAPresenter.addView( view2: this);
    if (!getIntent().getExtras().getBoolean( key: "publicCA") && !getIntent().getExtras().getBoolean( key: "updateCA")) {
        teacherCAPresenter.getAllCA(getValueLongPreferences( clave: "TEMA"), getValueStringPreferences( clave: "NICK"), BaseData.getLanguage());
    } else {
        teacherCAPresenter.getAllCANoPublished(getValueLongPreferences( clave: "TEMA"), getValueStringPreferences( clave: "NICK"), BaseData.getLanguage());
    }
}
```

Figura 60: Código de carga de actividad donde se piden los exámenes de un profesor que no estén publicados (getAllCANoPublished).

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_update_ca);
    this.setSupportActionBar();
    btn_sound = MediaPlayer.create( context: this, R.raw.bump);
    teacherCAPresenter = new TeacherCAPresenter();
    teacherCAPresenter.addView( view4: this);
    initializeViews();
    pulsaciones = 0;
    questionList = new ArrayList<>();
    alternativoList = new ArrayList<>();
    teacherCAPresenter.getAllQuestions(getLongPreferences( key: "TEMA"), BaseData.getLanguage());
    showDialogQuestionadd("Modify exam", "To modify a learning path, first select the question yo...");
}
```

Figura 61: Código de carga de actividad que realiza petición para obtener todas las preguntas del repositorio

El proceso es similar al de creación, aunque en este caso, se muestra al usuario dos RecyclerView en lugar de 1. Uno que contiene el examen seleccionado, y otro que contiene todas las preguntas del repositorio para que el usuario seleccione del examen la pregunta que desea quitar, y del repositorio de preguntas la que desea añadir en su lugar. La aplicación primero pregunta al usuario cual es la pregunta que quiere modificar de la que ha seleccionado, la principal o la alternativa, de manera que cuando se deba construir de nuevo el árbol a persistir, se sustituya en el lugar adecuado la pregunta. El proceso visual para modificar un examen puede verse en las figuras 62 y 63 respectivamente.



Figura 62: Pantalla de modificación de un CA, arriba el examen y debajo las preguntas

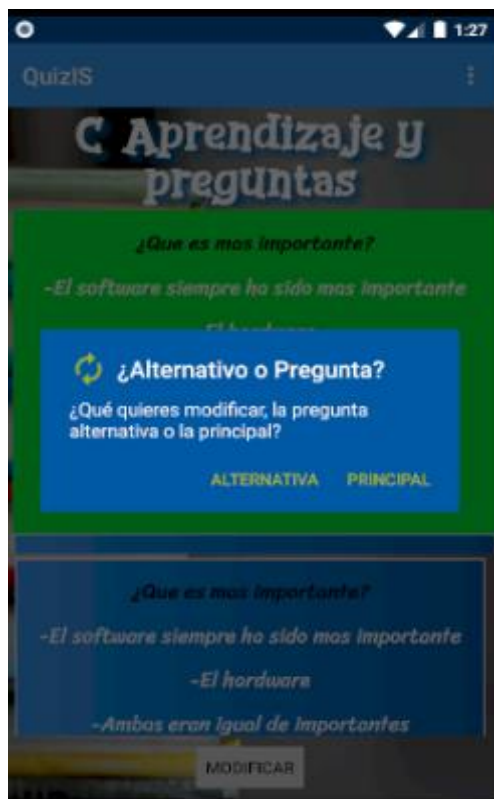


Figura 63: Selección de pregunta a modificar en el CA

En el caso de que el usuario quiera eliminar un examen, de igual forma se le mostrarán todos (los publicados y no publicados) ya que en este caso se podría eliminar un examen que esté publicado y afectaría directamente a las estadísticas del usuario que hubiera conseguido aprobar el mismo.

### **7.3 API REST.**

En este apartado se especificará el diseño de la API REST implementada. Como en el apartado anterior, las funcionalidades serán agrupadas por módulos. La decisión de escoger Spring como framework para el desarrollo de la API se ha tomado principalmente por las siguientes características:

- Rapidez en el desarrollo de microservicios
- Facilidad de securizar la aplicación rápidamente con una de las partes de Spring como es Spring Security
- Posibilidad de escalar la aplicación de manera sencilla
- Extensa documentación que proporciona.

En todas las fases de desarrollo se ha utilizado el servidor de aplicaciones Tomcat que ofrece Spring embebido de forma interna ya que brinda una gran facilidad para desplegar la aplicación y levantar el servidor. Sin embargo, también se ha generado el archivo “.war” y la base de datos MySQL para desplegarse en servidores online ya que es de gran utilidad para realizar pruebas sobre un servidor real (no obliga a tener siempre arrancado la máquina en local y facilita la posterior evaluación de la aplicación). En la figura 64 se muestra el diagrama de clases de la aplicación para el módulo Usuario. Es similar para el resto de los módulos y se puede comprobar que como punto de entrada a la API está el controlador que da el acceso a los diferentes servicios disponibles. Como la aplicación está implementada para soportar múltiples idiomas, en algunos de estos controladores que contienen el path de acceso a los servicios, se debe indicar el idioma en el que se quiere que se busquen y traten los datos en base de datos. Por ejemplo, cuando el usuario realiza una petición de un examen, la API recibe en que idioma necesita ese examen y se encarga de buscarlo. El controlador tiene inyectado un bean de Spring del service o modelo adecuado para poder invocar el método correspondiente. Este service está representado con una interfaz que será implementada por su clase correspondiente y realizará la lógica de negocio necesaria para el tratamiento de la información. La clase \*ServiceImpl, tiene inyectado un bean de Spring del repositorio concreto de JPA. En este caso el bean es UserRepository que contiene las operaciones básicas CRUD ya implementadas a través de la interfaz de JPA. Además, sobre el repositorio se pueden añadir queries en JPQL para realizar operaciones sobre la BD.

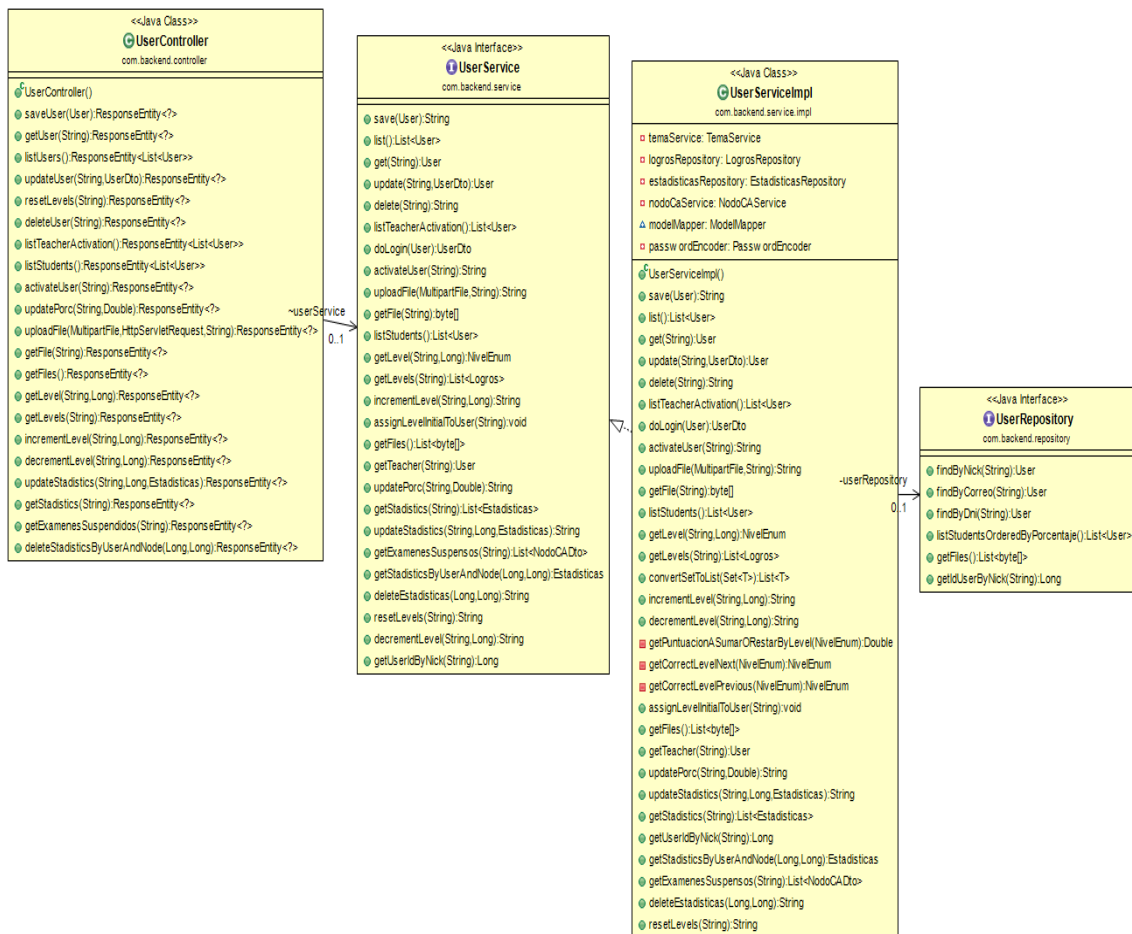


Figura 64: Diagrama de clases de la API para el módulo Usuario

Para definir propiedades de Spring en el manejo de los Beans, configuraciones internas y la conexión con la base de datos, se utiliza un fichero de propiedades denominado “application.properties”. En la figura 65 puede verse este fichero con dichas sentencias: se indica el usuario y contraseña de acceso a base de datos, la url donde estará accesible, propiedades de generación y creación de las distintas tablas a través del ORM, y también la propiedad de inicialización de los datos (ya que puede darse el caso al realizar pruebas de que se quiera introducir unos datos iniciales para el inicio de sesión). Se ha añadido un script que introduce datos de exámenes, usuarios y preguntas para facilitar las pruebas de la aplicación. Cambiando la propiedad de este fichero “spring.datasource.initialization-mode=never” a “always” y la propiedad “spring.jpa.hibernate.ddl-auto=none” a “create”, al arrancar el servidor, Hibernate se encargará de eliminar todas las tablas que haya en la base de datos, generar un esquema nuevo, e introducir los datos facilitados en el script llamado “data-oracle.sql”.

```

#Conexión con BBDD, datasource y JPA
spring.datasource.url=jdbc:mysql://localhost:3306/quizis?serverTimezone=UTC
spring.datasource.username=root
spring.datasource.password=root
spring.jpa.database-platform=org.hibernate.dialect.MySQL57Dialect
#spring.datasource.platform=data-oracle.sql
spring.datasource.data=classpath:data-oracle.sql
#Propiedad para ejecutar script de insercion de datos
spring.datasource.initialization-mode=never
#Propiedad para crear esquema de bd
spring.jpa.hibernate.ddl-auto=none
spring.jpa.show-sql=true
logging.level.org.hibernate.SQL=debug
spring.servlet.multipart.enabled=true
spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl

```

*Figura 65: Fichero propiedades de Spring*

El script de inicialización de los datos se encuentra en el directorio del proyecto src/main/resources junto con el application.properties indicado anteriormente. En la figura 66 se muestra parte del script que se encarga de ejecutar las sentencias necesarias introducir los datos en la base de datos.

```

@RestController
@RequestMapping(value = "/user")
public class UserController {

    @Autowired
    UserService userService;

    // Añadir un usuario nuevo
    @PostMapping(produces = MediaType.APPLICATION_JSON_UTF8_VALUE, consumes = MediaType.APPLICATION_JSON_UTF8_VALUE)
    public ResponseEntity<?> saveUser(@RequestBody User user) throws Exception {
        try {
            String resultado = (userService.save(user));
            if (!user.isAdmin() && !user.getTeacher()) {
                // De esta forma le asignamos en la tabla logros el usuario con todos los temas disponibles
                // Solo si es alumno, para que tenga relación creada entre ellos y solo haga falta ir actualizando el nivel
                userService.assignLevelInitialToUser(user.getNick());
            }
            return new ResponseEntity<String>(resultado, HttpStatus.OK);
        } catch (Exception e) {
            return new ResponseEntity<>(e.getMessage(), HttpStatus.BAD_REQUEST);
        }
    }
}

```

*Figura 66: Script de inicialización de datos*

En el fragmento de código de la figura 67 puede observarse el controlador o punto de entrada a la API con alguno de los servicios presentes en el módulo usuario.

```

INSERT INTO `TEMA` (`id`, `descripcion`, `nombre`) VALUES
(1, 'Tema 1', 'El producto'),
(2, 'Tema 2', 'El proceso'),
(3, 'Tema 3', 'Gestion de proyectos'),
(4, 'Tema 4', 'Metricas del proceso'),
(5, 'Tema 5', 'Estimacion de software'),
(6, 'Tema 6', 'Analisis de requisitos'),
(7, 'Tema 7', 'Gestion del riesgo'),
(8, 'Tema 8', 'Planificacion temporal y PP'),
(9, 'Tema 9', 'Garantia de calidad'),
(10, 'Tema 10', 'Gestion de la configuracion'),
(11, 'Tema 11', 'El modelo de objetos'),
(12, 'Tema 12', 'UML'),
(13, 'Tema 13', 'El proceso unificado de desarrollo de software'),
(14, 'Tema 14', 'La arquitectura multicapa'),
(15, 'Tema 15', 'Pruebas orientadas a objetos'),
(16, 'Tema 16', 'Patrones de diseño');

INSERT INTO `USER` (`id`, `activated`, `admin`, `apellidos`, `correo`, `dni`, `img_profile`, `nick`, `nombre`, `password`, `po
(1, b'1', b'0', 'Garcia López', 'jgarcia@prueba.es', '02233445W', 0x89504e470d0ala0a0000000d4948445200000dc000001610806000000
INSERT INTO `USER` (`id`, `activated`, `admin`, `apellidos`, `correo`, `dni`, `img_profile`, `nick`, `nombre`, `password`, `po
(2, b'0', b'0', 'Alvarez Perez', 'palvarez@ucm.es', '03344550F', 0x89504e470d0ala0a0000000d4948445200000bd0000012c08060000009
INSERT INTO `USER` (`id`, `activated`, `admin`, `apellidos`, `correo`, `dni`, `img_profile`, `nick`, `nombre`, `password`, `po
(3, b'0', b'0', 'ape modificados', 'rgarucm.es', '02551089D', 0x89504e470d0ala0a0000000d49484452000000f0000000f008060000003e55e
INSERT INTO `USER` (`id`, `activated`, `admin`, `apellidos`, `correo`, `dni`, `img_profile`, `nick`, `nombre`, `password`, `po
(4, b'0', b'0', 'Recipe Santos', 'arturo@recio.es', '78692340', 0x89504e470d0ala0a0000000d4948445200000124000002260806000000fac
INSERT INTO `USER` (`id`, `activated`, `admin`, `apellidos`, `correo`, `dni`, `img_profile`, `nick`, `nombre`, `password`, `po
(5, b'0', b'1', 'admin', 'administrador', 'admin', 0xffd8ffe000104a46494600010100000100010000ffdb0084000404040404040505040606

INSERT INTO `LOGROS` (`level`, `tema_id`, `user_id`) VALUES

```

*Figura 67: Controlador o punto de entrada en la API*

A continuación, se definen los distintos módulos junto con los servicios que se exponen en cada uno de ellos en forma de tablas.

### 7.3.1 Módulo Usuario

Este módulo contiene todos los servicios relacionados con el perfil del usuario, login y registro, todos los logros e hitos correspondientes a los alumnos y las estadísticas de estos. La URL base sobre la que estarían montados estos servicios sería “/user”.

La descripción de los endpoint disponibles puede visualizarse en la tabla 1:

ENDPOINT	MÉTODO HTTP	HEADER	BODY	RESPUESTA	DESCRIPCIÓN
/	POST	-	objeto user en formato Json	Código de error (200 o 400) con msg.	Inserta un usuario en BBDD.
/nick	GET	Jwt usuario	-	Código de error(200 o 400) con objeto user en body	Obtiene un usuario de la BBDD
/	GET	Jwt usuario	-	Código de error (200 o 400) con usuarios	Obtiene lista de usuarios registrados
/nick	PUT	Jwt usuario	objeto user en formato Json	Código de error (200 o 400) con msg	Actualiza un usuario en BBDD
/level/nick/reset	PUT	Jwt usuario	-	Código de error con msg.	Actualiza estadísticas usuario
/nick	DELETE	Jwt usuario	-	Código de error con msg	Elimina un usuario de la BBDD
/teachers	GET	Jwt usuario	-	Código de error con msg y objetos usuarios(profesores) en Json	Obtiene todos los profesores que tienen solicitud enviada de registro y no aprobada por el administrador.
/students	GET	Jwt usuario	-	msg de error y objetos usuarios(alumnos) en Json	Obtiene todos los usuarios alumnos ordenados por porcentaje
/activate/nick	PUT	Jwt usuario	-	msg con código de error	Activa un usuario profesor.
/porc/nick/porcentaje	PUT	Jwt usuario	-	msg con código de error	Actualiza el porcentaje de juego completado de un usuario

/picture/{nick}	POST	Jwt usuario	imagen de perfil	msg con código de error	Actualiza la imagen de perfil de un usuario.
/pictures	GET	Jwt usuario	-	imágenes de los usuarios	Obtiene las imágenes de perfil de los alumnos.
/level/{user}/{IdTema}	GET	Jwt usuario	-	msg de error junto con el nivel del usuario	Obtiene el nivel de un usuario alumno en un determinado tema.
/levels/{user}	GET	Jwt usuario	-	msg de error, junto con array de niveles por tema	Obtiene todos los niveles de un usuario por tema.
/level/increment/{nick}/{idTema}	PUT	Jwt usuario	-	msg con código de respuesta	Incrementa el nivel de un usuario en un determinado tema
/level/decrement/{nick}/{idTema}	PUT	Jwt usuario	-	msg con código de respuesta	Decrementa el nivel de un usuario en un determinado tema
/stadistics/{nick}/{nodoCA}	PUT	Jwt usuario	objeto estadísticas de usuario	msg con código de respuesta.	Actualiza las estadísticas de un usuario en base al nivel que se encuentre.
/stadistics/{user}	GET	Jwt usuario	-	msg con código de respuesta, si es ok, array de estadísticas de usuario.	Obtiene Todas las estadísticas del usuario para todos los niveles
/exam/suspended/{user}	GET	Jwt usuario	-	msg con código de respuesta, si es ok, array de objetos examen	Obtiene los exámenes suspensos de un alumno
/idUsuario/{idNodoCa}	DELETE	Jwt usuario	-	msg con código de respuesta	Eliminación de las estadísticas de un usuario

*Tabla 1: Endpoint disponibles en el módulo usuario*

Un ejemplo de petición y respuesta para pedir un usuario concreto con un cliente de peticiones lanzada a la API puede visualizarse en la figura 68.

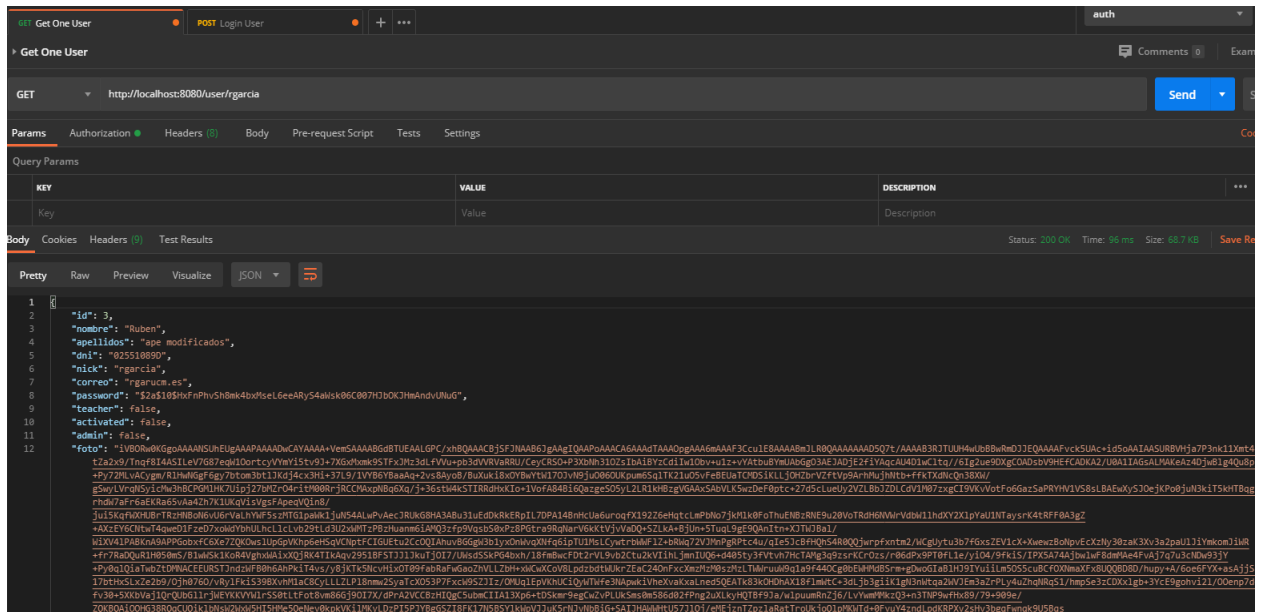


Figura 68: Petición a la API del usuario rgarcia

### 7.3.2 Módulo Question

Este módulo contiene todos los servicios relacionados con las preguntas del repositorio común para todos los profesores. La URL base sobre la que estarán montados estos servicios será “/question”. La descripción de los endpoint disponibles puede visualizarse en la tabla 2:

ENDPOINT	MÉTODO HTTP	HEADER	BODY	RESPUESTA	DESCRIPCIÓN
/	POST	JWT usuario	Objeto Question en formato JSON	Código de error (200 o 400) con msg.	Inserta una pregunta en BBDD.
/questions/{idTema}	GET	JWT usuario	-	Código de error (200 o 400) con objeto array de preguntas	Obtener todas las preguntas para un determinado tema
/questions/{idTema}/language	GET	JWT usuario	-	Código de error (200 o 400) con objeto array de preguntas	Obtener todas las preguntas para un determinado tema y lenguaje
/{id}	GET	JWT usuario	-	Código de error (200 o 400) con msg y en caso de ok, objeto pregunta.	Obtener una pregunta
/	PUT	JWT usuario	objeto pregunta en formato Json	Código de error con msg.	Actualizar todos los datos de una pregunta
/{id}	DELETE	JWT usuario	-	Código de error con msg	Elimina una pregunta con sus respuestas de la BBDD
/suspended/{userNick}/nodoCa	GET	JWT usuario	-	Código de error con msg y array de objeto pregunta	Obtener una lista de preguntas falladas en un examen

Tabla 2: Endpoint disponibles en el módulo Question

### 7.3.3 Módulo Tema

Este módulo contiene todos los servicios relacionados con los temas del repositorio común para todos los usuarios. La URL base sobre la que estarán montados estos servicios será “/tema”. La descripción de los endpoint disponibles puede visualizarse en la tabla 3:

ENDPOINT	MÉTODO HTTP	HEADER	BODY	RESPUESTA	DESCRIPCIÓN
/	GET	JWT usuario	-	código de error (200 o 400) con msg y en caso de ok, array de objeto tema	Obtener todos los temas
/id	GET	JWT usuario	-	código de error (200 o 400) con msg y en caso de ok, objeto tema	Obtener un tema

*Tabla 3: Endpoint disponibles en el módulo Tema*

### 7.3.4 Módulo NodoCA

Este módulo contiene todos los servicios relacionados con los nodos que en conjunto representarán un examen completo, es decir, el árbol correspondiente a un camino de aprendizaje creado por un profesor. La URL base sobre la que estarán montados estos servicios será “/nodoca”. La descripción de los endpoint disponibles puede visualizarse en la tabla 4:

ENDPOINT	MÉTODO HTTP	HEADER	BODY	RESPUESTA	DESCRIPCIÓN
/	POST	JWT usuario	objeto nodoca formato Json	código de error con msg.	Inserta un nodo en BBDD.
/saveCA	GET	JWT usuario	array de objetos nodoca en forma de árbol.	código de error con msg.	Añade un CA completo, es decir, guarda un examen en BBDD.
/getCA/{nick}/{tema} /{nivel}/{language}	GET	JWT usuario	-	código de error (200 o 400) con objeto array de preguntas	Obtener CA completo aleatorio para un tema determinado en un determinado nivel (inicial, bronce, plata, oro) y lenguaje que esté publicado.
/list/nopublished/{tema}/{teacher}/{language}	GET	JWT usuario	-	código de error con msg y en caso de ok, array de objeto nodoca completo (estructura de árbol).	Obtener todos los CA(Exámenes) no publicados de un profesor en un determinado tema
/list/{tema}/{teacher}/{language}	GET	JWT usuario	-	código de error con msg y en caso de ok, array de objeto nodoca completo (estructura de árbol).	Obtener todos los CA de un profesor en un determinado tema

/ {id}	GET	JWT usuario	-	código de error con msg y en caso de ok, objeto nodoCA sin hijos	Obtener un nodo del CA
/public/{idParent}	PUT	JWT usuario	-	Código de error con msg	Publicar un CA o examen (Parámetro id del nodo padre)
/ {idParent}	DELETE	JWT usuario	-	Código de error con msg	Eliminar un examen de la BBDD.
/original/{idParent}/{idNodoASust}/{idQuestionAn}	PUT	JWT usuario	-	Código de error con msg	Actualizar un CA modificando Pregunta original {idQuestionASust} id de la pregunta que queremos sustituir, {idQuestionAn} id de la pregunta a añadir
/alt/{idParent}/{idNodoASust}/{idQuestionAn}	PUT	JWT usuario	-	Código de error con msg	Actualizar un CA modificando Pregunta alternativa, {idQuestionASust} id de la pregunta que queremos sustituir, {idQuestionAn} id de la pregunta a añadir

Tabla 4: Endpoint disponibles en el módulo NodoCA

### 7.3.5 Seguridad de la API

Aunque ya se ha mencionado anteriormente, la API REST está securizada a través del módulo de Spring Security con JWT. El usuario que intente acceder a ella sólo tendrá acceso sin login a dos de los servicios expuestos: registrarse y autenticarse. En la autenticación, si el usuario introduce correctamente sus credenciales, la API generará a través de security un JWT o token de sesión que contiene de forma cifrada la información del usuario además de la información de la sesión junto con la duración. La aplicación consta de dos módulos principales encargados de gestionar esta seguridad, uno de ellos es el módulo de autenticación que se encarga de autenticar si las credenciales son correctas y dar el paso al siguiente módulo, el de autorización, encargado de devolver la autorización al usuario en JWT.

En la figura 69 puede verse un resumen de las validaciones que se aplicarían para la autenticación entre las que estarían la comprobación de si el usuario es profesor, y si está activada su cuenta para saber qué tipo de respuesta devolver.

```
@Override
protected void successfulAuthentication(HttpServletRequest request, HttpServletResponse response, FilterChain chain,
Authentication auth) throws IOException, ServletException {

    com.backend.model.JwtUser user = (JwtUser) auth.getPrincipal();
    String token = Jwts.builder().setIssuedAt(new Date()).setIssuer(Constant.ISSUER_INFO).setSubject(user.getUsername())
        .setExpiration(new Date(System.currentTimeMillis() + Constant.TOKEN_EXPIRATION_TIME))
        .signWith(SignatureAlgorithm.HS512, Constant.SUPER_SECRET_KEY).compact();

    User usr = userRepository.findByNick(user.getUsername());
    if(usr.getTeacher() && !usr.getActivated()) {
        response.setStatus(HttpServletResponse.SC_BAD_REQUEST);
        response.setContentType("application/json");
        response.getWriter().write("Esperando aprobación del usuario por parte del administrador");
        response.getWriter().flush();
        response.getWriter().close();
        return;
    }
    else {
        UserDto userDto = new UserDto();
        userDto.setJwt(Constant.TOKEN_BEARER_PREFIX + " " + token);
        usr.setEstadisticas(null);
        usr.setCamino_aprendizaje(null);
        usr.setFoto(null);
        usr.setLogros(null);
        userDto.setUser(usr);

        String employeeJsonString = this.gson.toJson(userDto);

        response.setStatus(HttpServletResponse.SC_OK);
        response.setContentType("application/json");
        response.getWriter().write(employeeJsonString);
        response.getWriter().flush();
    }
}
```

Figura 69: Validaciones para autenticación de login en la API

El método utilizado en la aplicación encargado de generar el JWT de la sesión del usuario puede verse en la figura 70.

```
public static String getJWTToken(String username) {
    List<GrantedAuthority> grantedAuthorities = AuthorityUtils.commaSeparatedStringToAuthorityList("ROLE_USER");
    String token = Jwts.builder().setId("quizisJWT").setSubject(username)
        .claim("authorities",
            grantedAuthorities.stream().map(GrantedAuthority::getAuthority)
                .collect(Collectors.toList()))
        .setIssuedAt(new Date(System.currentTimeMillis()))
        .setExpiration(new Date(System.currentTimeMillis() + 86400000))
        .signWith(SignatureAlgorithm.HS512, Constant.SUPER_SECRET_KEY.getBytes()).compact();
    return "Bearer " + token;
}
}
```

---

*Figura 70: Método de generación de token de sesión o JWT*

## **CAPITULO 8: EVALUACIÓN**

En este capítulo se describirán los resultados de una evaluación de la usabilidad que se ha realizado en la aplicación implementada. Se han evaluado los roles de administrador, alumno o profesor, y en base al tipo, se ha mostrado un tipo de preguntas u otro.

### **8.1 Diseño de la evaluación**

Para realizar la encuesta se han considerado personas procedentes de la docencia, alumnos y personas ajenas a la universidad. Se han realizado un total de 12 encuestas, de las cuales 6 son con el rol de alumno, 4 como profesor y 2 como administrador. Las preguntas realizadas en la evaluación se miden usando una escala de Likert entre 0 y 5 donde 0 sería el menos satisfecho y 5 el que más.

La evaluación se ha realizado usando Google Forms. En cada bloque de la encuesta se puede visualizar los pasos a seguir antes de realizarse la pregunta de valoración al usuario. La clasificación de los bloques de preguntas es la siguiente:

- El primer bloque de preguntas muestra preguntas generales relacionadas con la edad, relación con la universidad, y sexo (ver figura 71)
- El segundo bloque muestra preguntas relacionadas con el rol del usuario. En este punto el usuario debe seguir los pasos marcados antes de valorar las preguntas. (Ver figura 72)
- El tercer bloque de preguntas consta de preguntas sobre la evaluación global de la aplicación y propuestas de mejoras y cambios donde el usuario puede dar libremente su opinión. (Ver figura 73)

# QUIZIS

## Valoración de QuizIS

Introduce tus datos

\*Obligatorio

Introduce tu nombre \*

Tu respuesta

Introduce tu edad \*

Tu respuesta

Sexo \*

Hombre

Mujer

¿Estás relacionado con la universidad? \*

Sí, soy profesor

Si, soy alumno

No

¿Vas a responder como profesor, alumno o administrador? \*

Elige

Figura 71: Bloque inicial de preguntas

**Ranking alumnos**  
 Pulsa sobre el menú y accede a la opción de Ranking de alumnos, podrás visualizar el ranking de los mejores alumnos por porcentaje de juego completado. Además si pulsas sobre un alumno del ranking, podrás acceder a una pantalla que muestra los niveles que tiene en cada tema, y si dejas pulsado sobre tu usuario dentro del ranking, podrás visualizar tus estadísticas (exámenes suspensos, aprobados, preguntas falladas, tiempo medio de respuesta en los exámenes...etc).

Indica tu grado de satisfacción con la pantalla de ranking de alumnos \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con las pantallas de estadísticas del alumno \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con el modo de visualización de los exámenes y preguntas suspensas \*

0 1 2 3 4 5

Muy malo       Muy bueno

*Figura 72: Pasos a seguir para poder responder y preguntas específicas*

**Valoración de QuizIS**

Danos tu opinión

Indica tu grado de satisfacción con los colores e interfaz de la aplicación. \*

0    1    2    3    4    5

Muy malo                            Muy bueno

Indica tu grado de satisfacción con las animaciones de la aplicación \*

0    1    2    3    4    5

Muy malo                            Muy bueno

Indica tu grado de satisfacción con la facilidad de uso de la aplicación. \*

0    1    2    3    4    5

Muy malo                            Muy bueno

Indica tu grado de satisfacción general \*

0    1    2    3    4    5

Muy malo                            Muy bueno

*Figura 73: Evaluación global de la aplicación*

## 8.2 Resultados de la evaluación

En la figura 74 puede visualizarse la vinculación de los encuestados con el mundo de la docencia. El resultado está repartido al ser un 50% personas ajenas a la universidad y un 50% el total de personas que estudian e imparten o han impartido clase.

¿Estás relacionado con la universidad?  
12 respuestas

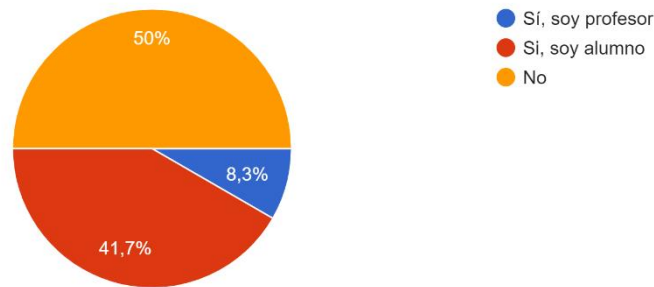


Figura 74: Porcentaje de vinculación al mundo universitario

En la figura 75 se puede visualizar como el 50% de los encuestados decidió realizar la encuesta con el rol de alumno, y el otro 50% se lo reparten entre profesor y administrador.

¿Vas a responder como profesor, alumno o administrador?  
12 respuestas

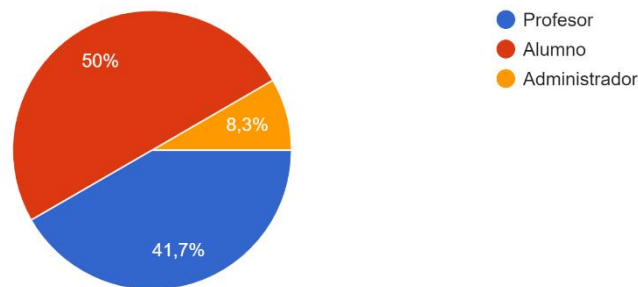


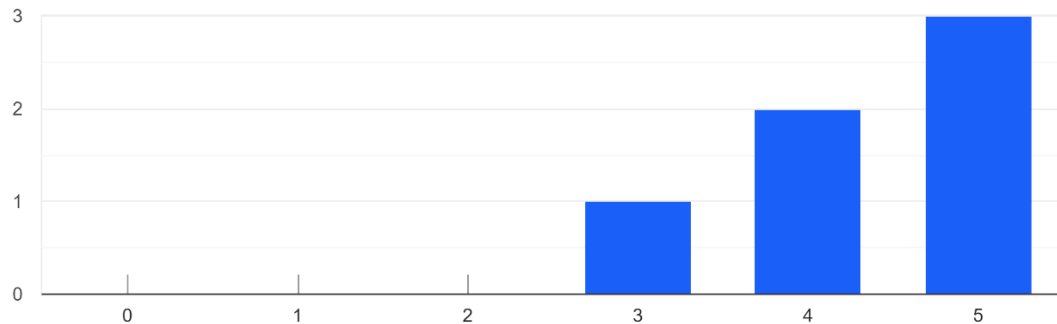
Figura 75: Porcentaje en base al rol elegido para elaborar la encuesta

### 8.2.1 Preguntas referidas al rol alumno.

De todos los usuarios que respondieron como alumno, el resultado de la evaluación sobre cómo realizar una edición de perfil, subir una imagen nueva y la facilidad del proceso puede visualizarse en la figura 76. Todos los resultados de evaluación estaban por encima de los 3 puntos.

Indica tu satisfacción con la facilidad y usabilidad del proceso realizado

6 respuestas

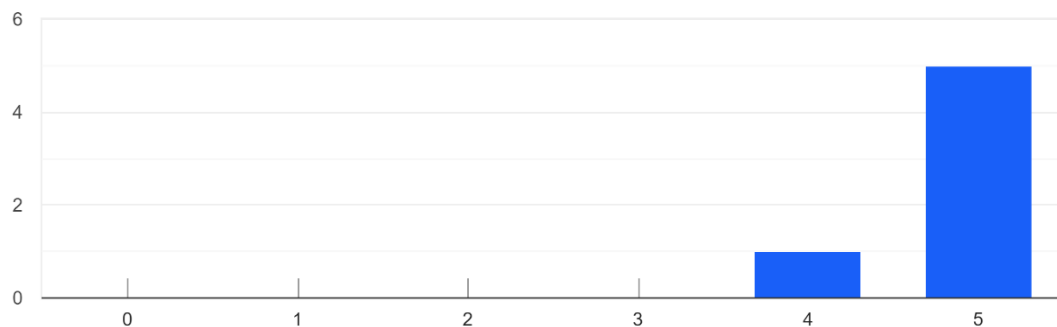


*Figura 76: Evaluación por parte del rol alumno del proceso de editar perfil e imagen y facilidad del proceso.*

La satisfacción con la pantalla del juego que muestra los temas para el rol alumno puede visualizarse en la figura 77 situándose esta entre los 4 y 5 puntos para todos los casos.

Indica tu grado de satisfacción con la pantalla que muestra los temas y niveles

6 respuestas



*Figura 77: Valoración de la pantalla de juego con la interfaz de los temas para el rol alumno.*

En las figuras 78 puede visualizarse la evaluación para el proceso de subir una imagen de perfil. Hay un alto porcentaje que ha valorado el proceso con 2 o 3 puntos mientras que el resto de las puntuaciones se sitúa por encima de 3.

Indica tu satisfacción con la pantalla de editar perfil y añadir una nueva imagen de perfil

6 respuestas

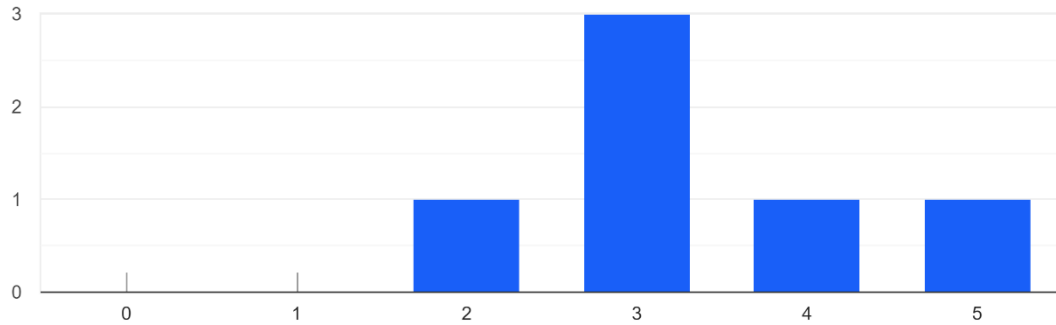


Figura 78: Valoración del proceso de subir una imagen nueva de perfil para el rol alumno.

En cuanto al proceso de realizar un examen, visualizar las estadísticas del alumno y cerrar sesión (ver figura 79), el resultado es positivo al votar el 100% de los encuestados una puntuación de 5 en la escala de Likert. (ver figura 80)

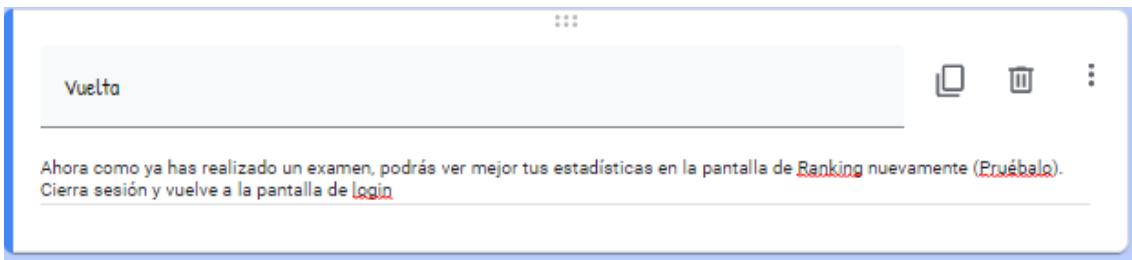


Figura 79: Proceso pedido en la evaluación (Realizar un examen, y cerrar sesión)

Indica tu grado de satisfacción con este proceso

6 respuestas

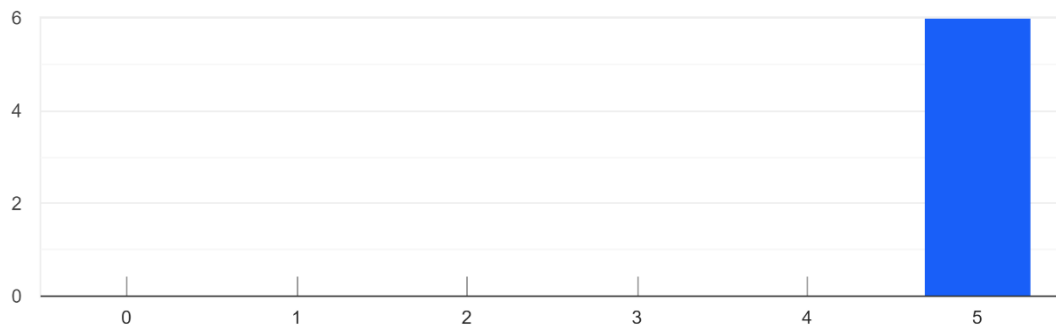
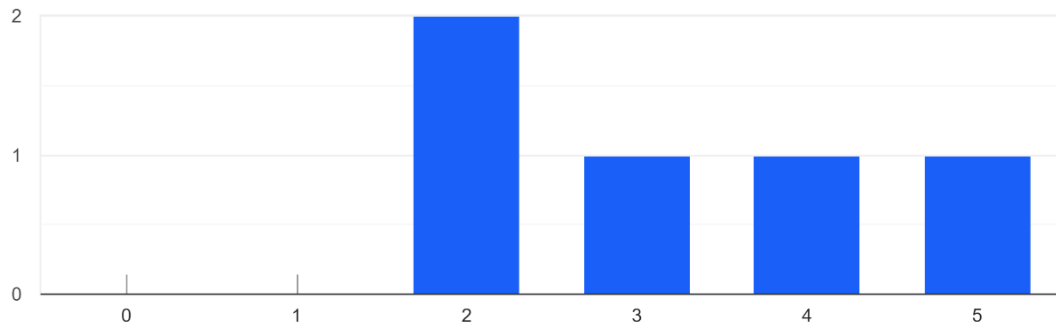


Figura 80: Valoración del proceso de realizar un examen y cerrar sesión.

### 8.2.2 Preguntas referidas al rol profesor.

En la evaluación de los usuarios como profesor, La satisfacción con el proceso de editar el perfil y subir una imagen puede visualizarse en la figura 81. En los datos mostrados la mayoría de la puntuación se sitúa en 2 puntos, aunque el resto de las puntuaciones son positivas estando por encima de 3.

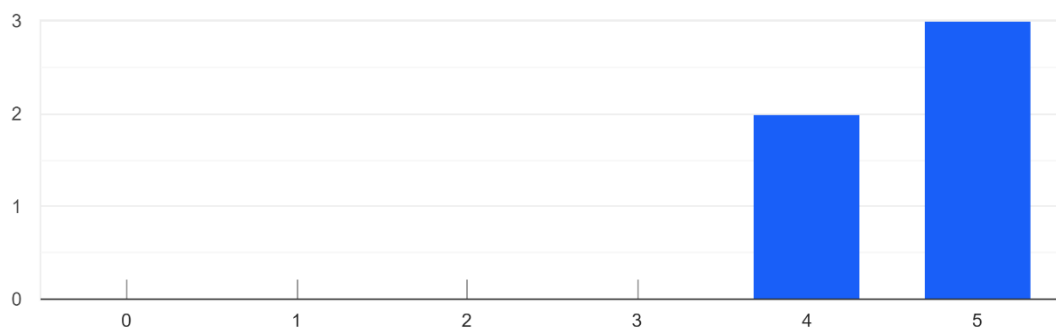
Indica tu satisfacción con la pantalla de editar perfil y añadir una nueva imagen de perfil  
5 respuestas



*Figura 81: Valoración del proceso de subir una imagen nueva de perfil para el rol profesor.*

La evaluación sobre pantalla del juego que muestra los temas puede visualizarse en la figura 82. Se puede comprobar que la valoración general de todos los encuestados está en 4 o 5 puntos.

Indica tu grado de satisfacción con la pantalla que muestra los temas  
5 respuestas

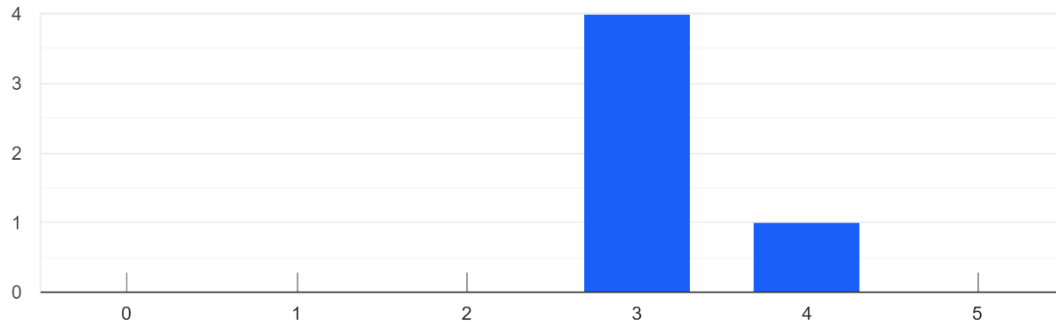


*Figura 82: Valoración de la pantalla de juego con la interfaz de los temas para el rol profesor.*

El grado de satisfacción con la facilidad para crear un examen o camino de aprendizaje y la de actualizar una pregunta del repositorio común puede visualizarse en las figuras 83 y 84 respectivamente.

Indica tu grado de satisfacción con la facilidad en la creación de un examen

5 respuestas

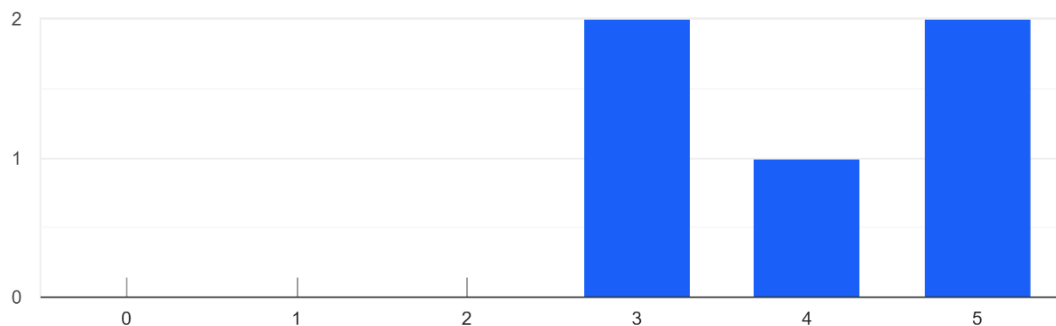


*Figura 83: Valoración del proceso de crear un camino de aprendizaje o examen.*

La mayoría de puntuación se encuentra en los 3 puntos y el resto por encima.

Indica tu grado de satisfacción con la pantalla de actualizar pregunta

5 respuestas

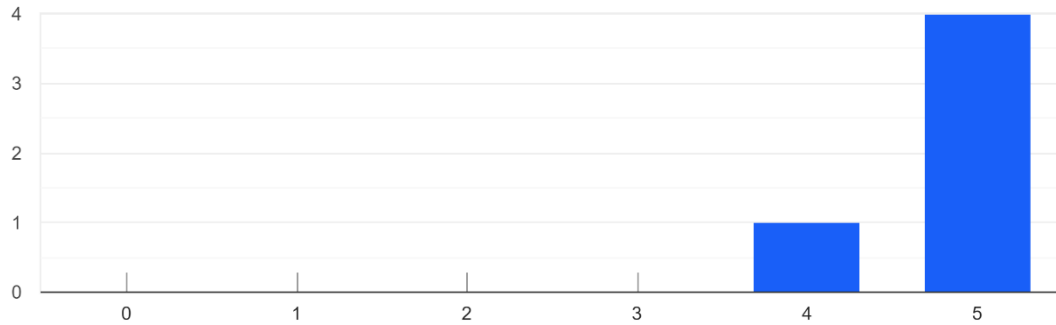


*Figura 84: Valoración del proceso de actualizar una pregunta para los profesores.*

Todas las puntuaciones se encuentran por encima de los 3 puntos, siendo 3 y 5 la gran mayoría.

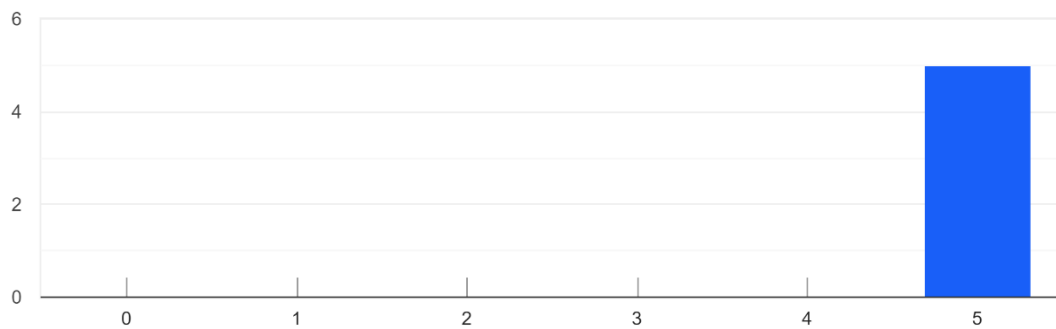
La satisfacción con las pantallas de publicación y eliminación de un examen para los usuarios que realizan la encuesta como profesores puede visualizarse en las figuras 85 y 86 respectivamente. Puede comprobarse que la puntuación está por encima de 4 en todos los casos siendo bastante positiva.

Indica tu grado de satisfacción con la pantalla de publicar camino de aprendizaje o examen  
5 respuestas



*Figura 85: Valoración de la pantalla de publicación de exámenes*

Indica tu grado de satisfacción con la pantalla de eliminar un camino de aprendizaje o examen  
5 respuestas



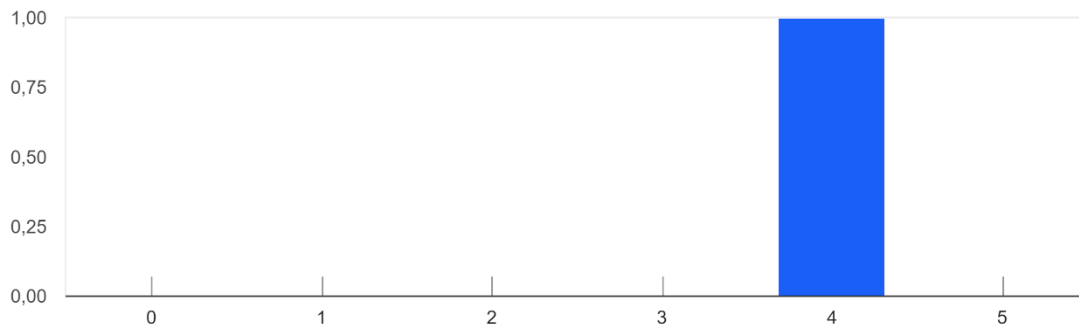
*Figura 86: Valoración de la pantalla de borrado de exámenes*

### **8.2.3 Preguntas referidas al rol administrador.**

Para el usuario que realiza la encuesta como administrador, en la evaluación del proceso de eliminar a algún usuario y la interfaz mostrada, el resultado se muestra en la figura 87. Todas las valoraciones han sido positivas con 4 puntos.

Indica tu satisfacción con la facilidad y usabilidad del proceso realizado

1 respuesta



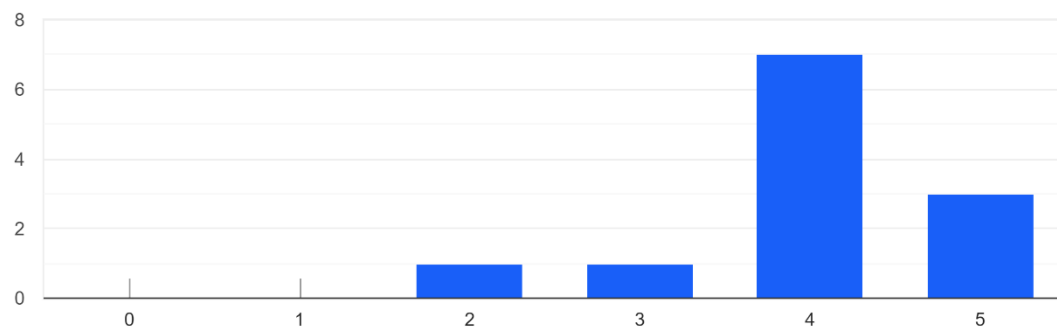
*Figura 87: Valoración del proceso e interfaz en la eliminación de un usuario por parte de rol administrador.*

### 8.2.4 Valoraciones generales

Por último, la valoración general con los colores de la interfaz y de la aplicación puede visualizarse en la figura 88. Se puede comprobar que hay multitud de opiniones, situándose la mayoría en 4 y 5 puntos, aunque también se ha votado con 2 y 3 puntos en algunos casos.

Indica tu grado de satisfacción con los colores e interfaz de la aplicación.

12 respuestas

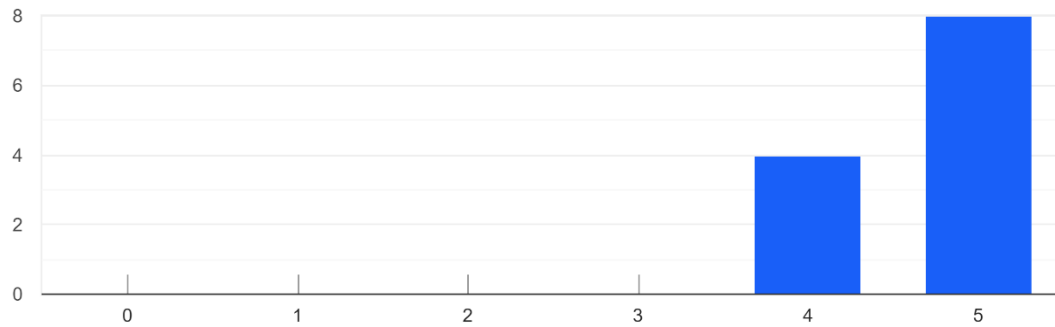


*Figura 88: Valoración general sobre los colores e interfaz de la aplicación*

A continuación, en la figura 89 se muestra el grado de satisfacción con la facilidad de uso de la aplicación, de forma que el 33% puntúa con un valor de 4 y el 66% puntúa con un valor de 5 puntos. En la figura 90 se muestra el grado de satisfacción general de todos los usuarios.

Indica tu grado de satisfacción con la facilidad de uso de la aplicación.

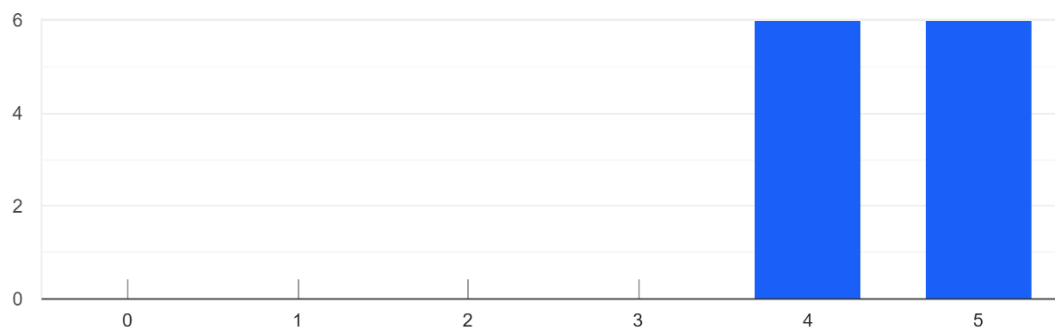
12 respuestas



*Figura 89: Valoración general sobre la facilidad de uso de la aplicación*

Indica tu grado de satisfacción general

12 respuestas



*Figura 90: Valoración general sobre la aplicación*

Algunos ejemplos de mejoras que proponen los usuarios sobre la aplicación pueden visualizarse en la figura 91.

---

## ¿Qué mejorarías?

8 respuestas

Poder ver la pantalla de forma horizontal y quitar el sonido de fondo al pulsar.

Eliminaría el color rojo de las preguntas

Algunos colores y diseños

Buenas, mejoraría el tema en los cuestionarios en las respuestas deberías poner algo al lado para poder diferenciar cual es a,b,c por ejemplo por que las definiciones están una encima de la otra pero sin ninguna manera de poder diferenciar cual es cual, otra cosa los colores de las letras deberías cambiarlo a algún tono que se diferencie mejor.

El sonido del tiempo en las preguntas de exámenes, el color y tipo de letra de algún texto y la corrección en al escribir el correo y demás en la página de registro

La visualización de algunos textos no es del todo correcta en algunos casos y modificar examen es algo engorroso

La interfaz, los colores

*Figura 91: Mejoras planteadas por los usuarios*

En el anexo 3, se puede consultar el estudio completo.

## **CAPÍTULO 9: CONCLUSIONES Y TRABAJO FUTURO**

### **9.1 Conclusiones**

Se ha conseguido desarrollar una aplicación que permite realizar un seguimiento del trabajo de los alumnos a través de la participación en el juego, entre los puntos que se han terminado, cabe destacar la implementación de una aplicación que permite a los profesores enseñar conceptos de IS a los alumnos mediante gamificación, ya que tiene la opción de publicar el contenido que más se adecúe a su asignatura. También permite otorgar porcentajes a los alumnos en función de la participación en el juego. Los profesores además pueden mantener y construir un repositorio común de preguntas y exámenes de la asignatura.

Los alumnos por otra parte pueden realizar exámenes o caminos de aprendizaje y conseguir medallas para aumentar el progreso del juego.

Con respecto a los alumnos que no dominen correctamente el idioma, la aplicación también da soporte para los idiomas inglés y español, por lo que podrán adaptarse correctamente al seguimiento de la asignatura.

### **9.2 Trabajo futuro**

La aplicación se podría mejorar de la siguiente manera:

- Ampliación del espectro de idiomas, ya que actualmente la aplicación ofrece soporte para inglés y español únicamente, pero podría ampliarse para el resto.
- Incluir un foro de dudas en un apartado del menú que sería de gran utilidad para que los usuarios pudieran responderse dudas entre ellos y redactar opiniones.
- Incluir un chat entre profesores y alumnos con el objetivo de favorecer la comunicación entre usuarios y la resolución de dudas.
- Ampliación de las asignaturas para jugar.
- Manejar otro tipo de exámenes mostrados: mostrar imágenes, dibujos, reordenación de palabras...etc. en el desarrollo de un examen en lugar de mostrar siempre preguntas de tipo texto.

## **CHAPTER 9: CONCLUSIONS AND FUTURE WORK**

### **9.1 Conclusions**

It has been possible to develop an application that allows to monitor the work of the students through participation in the game, among the points that have been completed, it is worth highlighting the implementation of an application that allows teachers to teach IS concepts to Students through gamification, since they have the option of publishing the content that best suits their subject. It also allows percentages to be awarded to students based on participation in the game. Teachers can also maintain and build a common repository of questions and exams for the subject.

On the other hand, students can take exams or learning paths and get medals to increase the progress of the game. With respect to students who do not correctly master the language, the application also supports the English and Spanish languages, so they will be able to adapt correctly to the monitoring of the subject

### **9.2 Future work**

The application could be improved in the following way:

- Expansion of the language spectrum, since currently the application offers support for English and Spanish only, but could be expanded for the rest.
- Include a forum of doubts in a section of the menu that would be very useful so that users could answer each other's doubts and write opinions.
- Include a chat between teachers and students to promote communication between users and the resolution of doubts.
- Extension of the subjects to play.
- Handle other types of exams shown: show images, drawings, rearrangement of words ... etc. in the development of an exam instead of always showing text-type questions.

## REFERENCIAS

[1] Vesselinov, R., & Grego, J. (2012). Duolingo effectiveness study. City University of New York, USA, 28.

[2] <https://mars.apprenderer.sm/>, revisado: 05/06/2020

[3] <https://como-funciona.com/oxford-learners-quick-grammar/>, revisado: 05/06/2020

[4] <https://www.todotest.com/>, revisado: 05/06/2020

[5] BALTIERRA, Eric Alvarez, et al. El verdadero impacto de los LMS en el aprendizaje de los alumnos contra el uso de los Métodos tradicionales. Revista Electrónica de Investigación e Innovación Educativa-REIIE ISSN: 2448-556X, 2019, vol. 4, no 2, p. 8-15.

<https://www.easy-lms.com/features/mobile-lms/item10250>, revisado: 05/06/2020

[6] GUTIÉRREZ, Juan Manuel Alavez. Google forms como instrumento de medición de conocimientos. Revista Vinculando, 2017.

[7] RODRÍGUEZ-FERNÁNDEZ, Leticia. Smartphones y aprendizaje: el uso de universitaria. Revista Mediterránea de Comunicación/Mediterranean Journal of Commu 8, no 1, p. 181-189.

[8] DEITEL, Harvey M.; DEITEL, Paul J. Como programar en Java. Pearson educación, 2003.

[9] JOHNSON, Rod, et al. The spring framework–reference documentation. interface, 2004, vol. 21, p. 27.

[10] GOODWILL, James. Deploying Web applications to Tomcat. En Apache Jakarta Tomcat. Apress, Berkeley, CA, 2002. p. 19-34.

[11] JONES, Michael; CAMPBELL, Brain; MORTIMORE, Chuck. JSON Web Token (JWT) profile for OAuth 2.0 client authentication and authorization Grants. May-2015.[Online]. Available: <https://tools.ietf.org/html/rfc7523>, 2015.

[12] KING, Gavin, et al. Hibernate reference documentation. 2004. <http://prdownloads.sourceforge.net/hibernate>, 2010.

[13] BARTHOLOMEW, Daniel. Mariadb vs. MYSQL. Dostopano, 2012, vol. 7, no 10, p. 2014.

[14] CHIANG, Geoff. Instant Spring Tool Suite. Packt Publishing Ltd, 2013.

[15] BÁEZ, Manuel, et al. Introducción a android. EME Madrid, España, 2012, vol. 121.

[16] <https://medium.com/contraslashsas/retrofit-para-android-desde-0-1c8be830a1af>, revisado: 05/06/2020

[17] STUDIO, Android. Android Studio. The Official IDE for Android, 2017.

[18] DABBISH, Laura, et al. Social coding in GitHub: transparency and collaboration in an open software repository. En Proceedings of the ACM 2012 conference on computer supported cooperative work. 2012. p. 1277-1286.

[19] DIAMANTARAS, Ilias. Interfaz de usuario de voz para sistemas de automatización. 2019.

## ANEXO

### Anexo I. Guía de uso

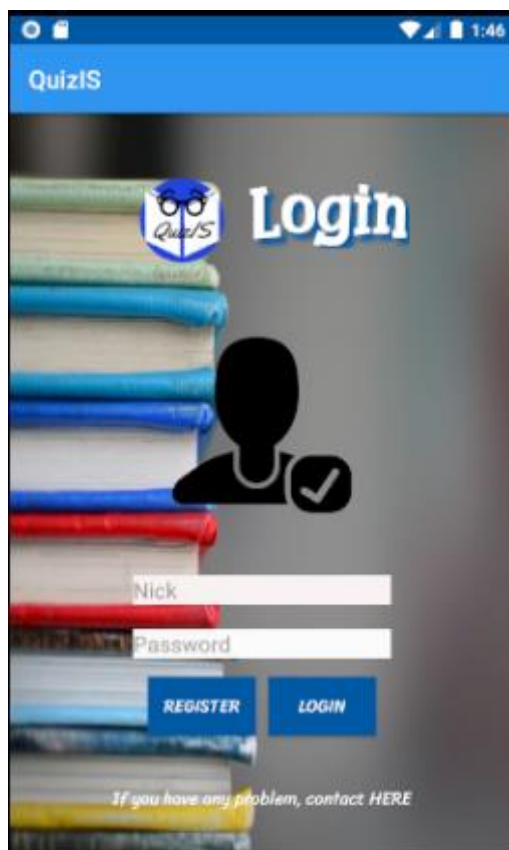
En este apartado se incluye una guía de uso en la que se explicará los distintos apartados y funcionalidades que contiene la aplicación. Ya que existen 3 tipos de usuarios, se explicarán por un lado las funcionalidades comunes y por otro las particulares de cada uno.

#### 1. Funcionalidades comunes

A continuación, se exponen las funcionalidades comunes para los 3 perfiles de usuario.

##### 1.1 Iniciar y cerrar sesión

Para poder iniciar sesión es necesario que los usuarios estén registrados previamente en la aplicación. La pantalla principal de login donde se deben introducir los datos del usuario es la siguiente (Ver figura 92).



*Figura 92: Pantalla inicial de login*

Tras iniciar sesión, la pantalla que se muestra al usuario autenticado es la de Home o Perfil, para cerrar sesión y volver a la pantalla principal sólo habría que pulsar sobre el botón cerrar sesión. (Ver figura 93)



*Figura 93: Pantalla home del usuario y botón de cerrar sesión*

## **1.2 Editar perfil**

Para editar la información que aparece del perfil del usuario lo único que debe realizar es pulsar sobre el botón editar y se redirigirá al usuario a una nueva vista que se corresponde con la de la figura 94



*Figura 94: Pantalla de edición de perfil del usuario*

Como se puede comprobar, se tienen 2 botones, el botón “Actualizar” actualizará la información del usuario con los campos que haya rellenado. Los obligatorios sería únicamente el de Password, que solicita la contraseña actual del usuario. Si se quisiera cambiar la contraseña, habría que completar el campo “Password”, y los dos siguientes, “New Password” y “Confirmar Password” con la nueva contraseña que quiere el usuario.

Si se pulsa sobre subir imagen, aparecerá una nueva vista para que el usuario actualice su imagen de perfil (ver figura 95).



*Figura 95: Actualizar imagen de perfil*

Las restricciones de la imagen a seleccionar son que esta debe tener como máximo 1Mb de tamaño y estar en la memoria interna del dispositivo. El usuario debería pulsar sobre “Subir” una vez ha seleccionado la imagen deseada.

### **1.3 Baja usuario**

Cualquier usuario puede darse de baja de la aplicación. Debe pulsar sobre el botón de la pantalla principal “Baja usuario” (ver figura 96). La aplicación preguntará al usuario si realmente quiere darse de baja total de la aplicación (Figura 97) y ejecutará la acción indicada.



Figura 96: Pantalla home del usuario con pulsación sobre “Baja usuario”

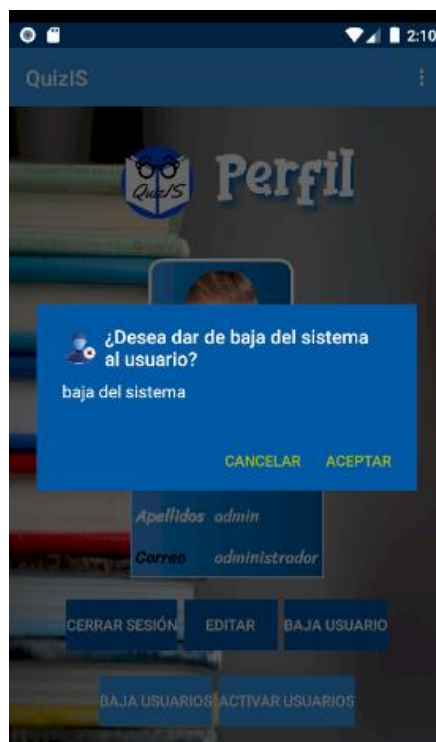


Figura 97: Aviso sobre la baja del usuario

## 1.4 Consulta Ranking y Registro

Estas dos funcionalidades solo están disponibles para los usuarios con roles de profesor y alumno.

Para consultar el ranking de los mejores alumnos, el usuario deberá pulsar sobre el menú situado en la parte superior a la derecha (Ver figura 98)



Figura 98: Posición del menú.

A continuación, la aplicación mostrara un panel de ayuda (Ver figura 99) ya que aparte de visualizar el ranking, se puede ver más información de los usuarios depende del rol con el que se esté autenticado. El usuario deberá pulsar en “Aceptar” y aparecerá una vista con el ranking de los mejores alumnos, el usuario podrá desplazar la lista hacia abajo para ver todos aquellos que no se vean a simple vista. La información que se puede visualizar en este punto es el Nombre, Nick, Email y porcentaje de juego completado. Se puede visualizar esta información en la figura 100.

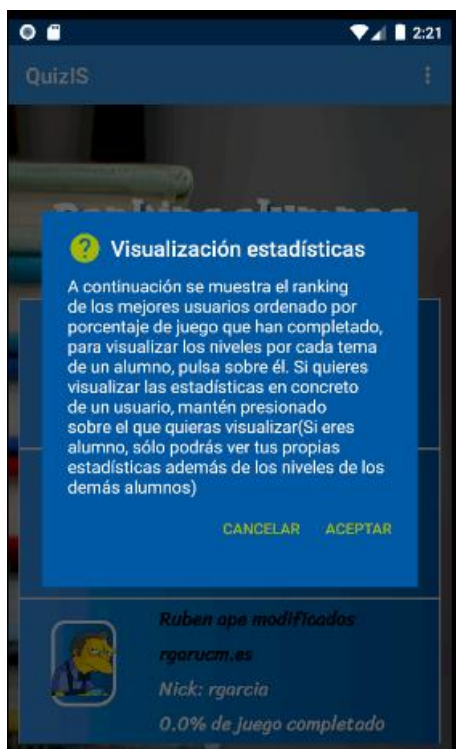
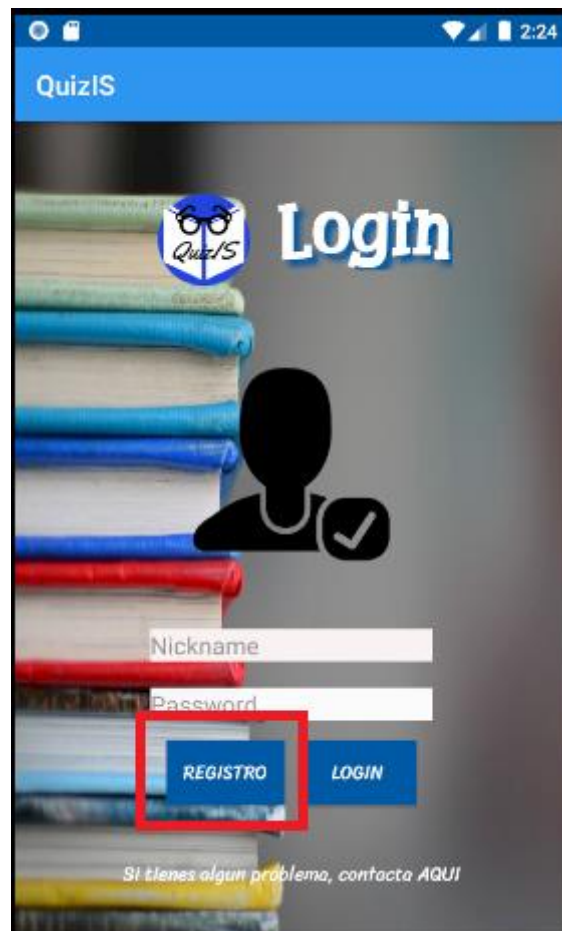


Figura 99: Ayuda sobre el ranking de alumnos



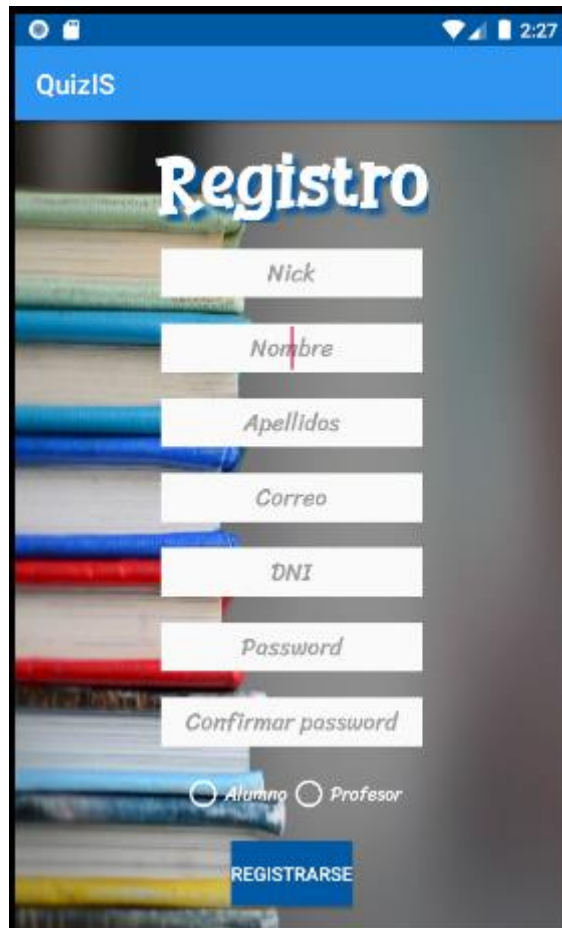
Figura 100: Información del ranking de los mejores alumnos que forman parte del juego

Para que un usuario se registre, debe estar en la pantalla inicial de login descrita en puntos anteriores y pulsar sobre el botón registro. (Figura 101)



*Figura 101: Pulsación sobre registro en la pantalla inicial*

A continuación, deberá introducir sus datos para el registro y pulsar sobre el botón “Registrarse”. El usuario debe seleccionar si se quiere registrar como alumno o profesor, en el caso de no seleccionarlo, se registrará automáticamente como alumno. (Ver figura 102)



*Figura 102: Pantalla para registrarse en la aplicación*

## **2. Funcionalidades Administrador**

A continuación, se exponen las funcionalidades específicas para el usuario administrador.

### **2.1 Activar usuarios**

Para que el administrador pueda dar de baja a cualquier usuario, en su pantalla de Home, debe pinchar sobre el botón “Baja Usuarios” situado en la parte inferior a la izquierda. (Ver figura 103)



*Figura 103: Botón de administrador para dar de baja a cualquier usuario*

Como se puede visualizar en la figura 103, la aplicación muestra en forma de notificación si el administrador tiene alguna cuenta de profesor por activar, es posible activar una cuenta de profesor a través del botón “Activar Usuarios” situado al lado de “Baja usuarios” o directamente pinchando sobre la notificación que muestra la aplicación. Cada vez que el administrador inicia sesión, se comprueba si el administrador tiene cuentas pendientes de activar. El resultado de pinchar sobre “Activar Usuarios” o sobre la notificación, es una nueva pantalla con la información de los profesores. El usuario solo tendría que pinchar sobre la cuenta del profesor que quiera activar, y se le mostrara un mensaje preguntando si quiere realmente activar esa cuenta, ya que, si pulsa sobre la opción “Aceptar”, el usuario que se haya registrado con esa cuenta tendrá acceso a todo el repositorio de preguntas y exámenes. Esta interfaz con las cuentas por activar y el mensaje de aviso puede visualizarse en la figura 104 y 105 respectivamente.



Figura 104: Pantalla de activación de profesores

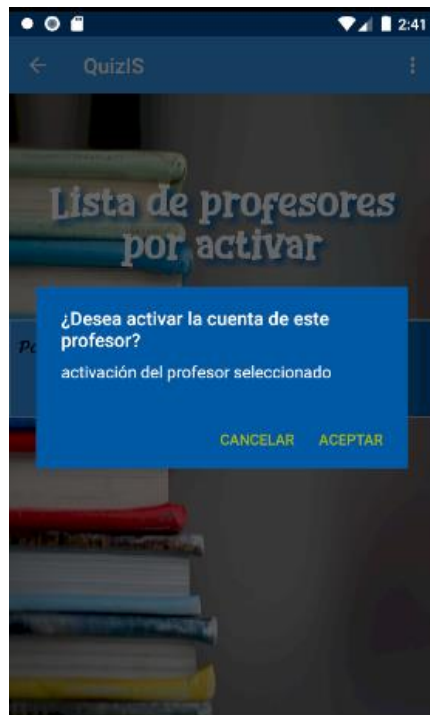


Figura 105: Aviso sobre pulsación para activar una cuenta de profesor

### 3. Funcionalidades Alumno

A continuación, se exponen las funcionalidades específicas para el usuario alumno.

#### 3.1 Visualizar hitos por camino de aprendizaje

En este apartado se contempla la posibilidad de que el usuario pueda ver sus estadísticas de los exámenes que ha suspendido, aprobado, tiempo medio de respuesta, nivel que tiene en cada tema...etc.

Para que el usuario pueda visualizar los niveles que tiene por tema y sus estadísticas debe ir a la pantalla de Ranking del menú. En la interfaz que se le muestra con todos los alumnos, el usuario podrá pinchar sobre su propio perfil o el de cualquier otro alumno y verá información de los niveles que tiene en cada tema. (Ver figura 106)

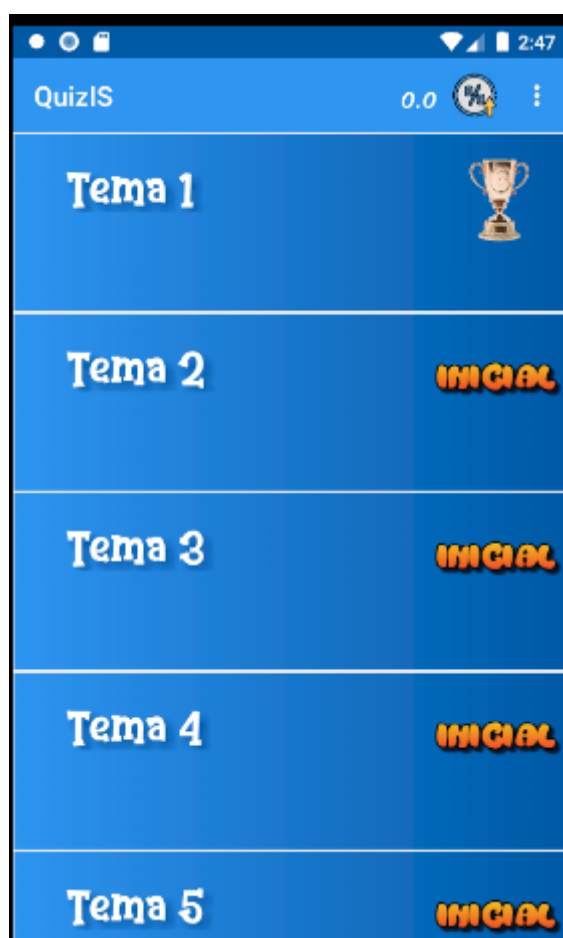
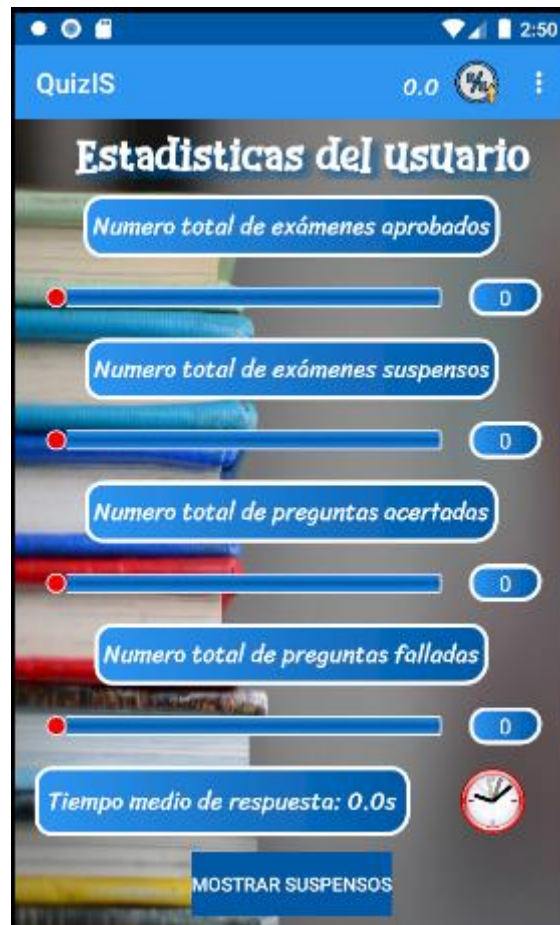


Figura 106: Interfaz mostrada al pulsar sobre un alumno en la pantalla de Ranking

El usuario también tiene la posibilidad de ver las estadísticas como se comentó anteriormente, pero en este caso, solo tiene permiso para ver las suyas propias y no la del resto de compañeros. Si el usuario mantiene pulsada la pantalla sobre su perfil en la pantalla de ranking, accederá a sus estadísticas propias. (Ver figura 107)



*Figura 107: Pantalla con estadísticas del usuario autenticado*

Si el usuario pulsa ahora sobre “Mostrar suspensos”, aparecería una pantalla con un resumen con los exámenes que ha suspendido (Ver figura 108) y pulsando sobre el examen en concreto, podría visualizar el examen completo y además las preguntas sobre las que ha fallado (Ver figura 109).

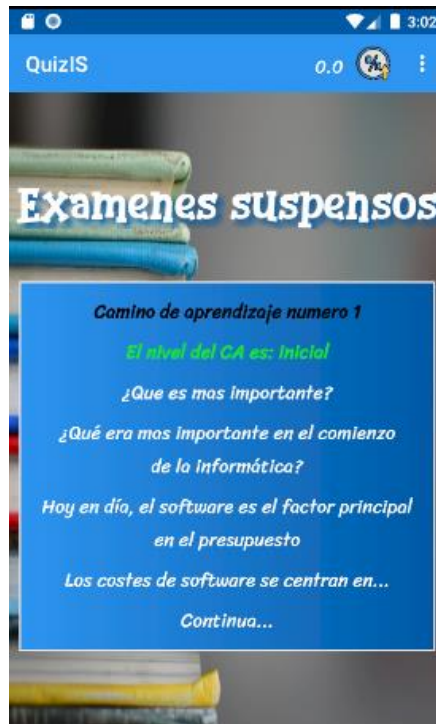


Figura 108: resumen de exámenes suspensos del usuario



Figura 109: Examen suspenso y preguntas falladas

### 3.2 Resetear puntuación y logros

El usuario puede reiniciar su contador de porcentaje de juego completado, y además borrar todas sus estadísticas. Para ello, en la pantalla Home, deberá pulsar sobre el botón “Resetear puntuación y logros”. Tras esta acción, reiniciará el juego completo y se encontrará en el nivel inicial de todos los temas, así como un 0% de juego completado, no tendrá exámenes suspensos ni estadísticas.

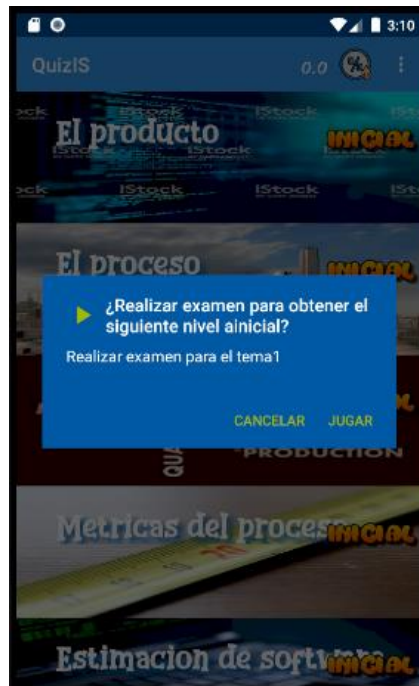
### 3.3 Realizar Camino de aprendizaje

Cuando el usuario quiera subir de nivel y probar sus conocimientos, así como mejorar sus estadísticas, deberá acudir a la pestaña del menú “Juego” donde se mostrará una interfaz con todos los temas disponibles y el nivel que tiene en cada uno de ellos. (Ver figura 110)



Figura 110: Pestaña juego del alumno

Si el usuario pulsa sobre cualquiera de los temas disponibles, recibirá una pregunta sobre si quiere realizar un examen para obtener el siguiente nivel en ese tema en concreto. (Ver figura 111)



*Figura 111: Pregunta mostrada al pulsar sobre un tema por parte del usuario alumno*

En el momento que se pulse en “Jugar”, el alumno comenzará un examen del tema seleccionado, si es que hay algún profesor que ha publicado exámenes para ese tema y ese nivel, en caso contrario volverá a la pantalla inicial y será informado de que aún no existen exámenes en el repositorio para obtener el siguiente nivel.

Si es el caso en el que, si hay exámenes disponibles, comenzará el juego (Ver figura 112).

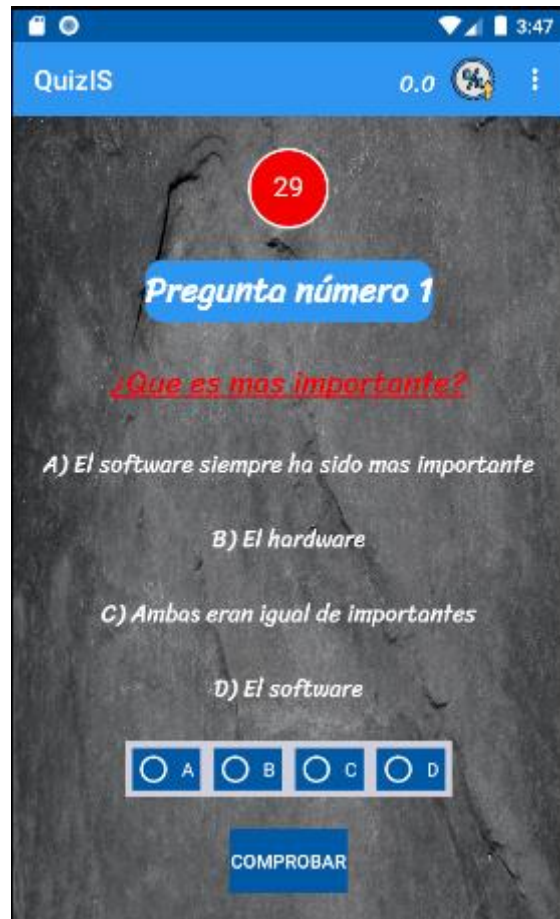


Figura 112: Comienzo del examen con la pregunta 1

### 2.3.1 Desarrollo del juego

Tiene 30 segundos para responder cada pregunta, si no, la aplicación pasara a la siguiente y se contará como fallo. El árbol de preguntas es, se muestra la pregunta número 1, si el alumno falla, pierde 0.5 puntos que le otorgaría una respuesta correcta, y se le mostraría una pregunta alternativa, que, en caso de ser acertada, sumaría 0,5 puntos y en caso de volver a fallar, 0 puntos. Este proceso continua hasta la pregunta 10. Por lo que el alumno podría tener que responder a 20 preguntas en total si falla en todos los casos. Cuando recibe la pregunta principal, si la falla, es informado de que ha fallado en la respuesta (Ver figura 113). Cuando acaba el examen, se le informa mediante un sonido acorde a la puntuación, si ha superado o no el examen. (Ver figura 114) y la aplicación sumaría el porcentaje de juego correspondiente en el caso de ser aprobado. Si el usuario abandona el examen, descenderá un nivel al actual que tenga en ese momento. (Ver figura 115)

Los 4 niveles del juego son inicial, bronce, plata u oro y cada examen cuenta un porcentaje de juego completado dependiendo de si se está realizando un examen para un determinado nivel, es decir, el nivel oro completo, compone un 45% de juego completado, el nivel plata un 35% y el nivel bronce un 20%. Si el usuario está en nivel inicial y realiza un examen del tema 1 para subir al nivel bronce, como son 16 temas los que hay disponibles, si aprueba, subirá un porcentaje de juego completado de  $20 / 16 = 1.25 \%$ . De igual forma se calcula para el nivel oro y plata.

Los criterios para pasar al siguiente nivel son los siguientes:

- ➔ Si el usuario tenía nivel inicial y obtiene una puntuación  $\geq 5$  en el examen, pasaría al siguiente nivel (bronce)
- ➔ Si el usuario tenía nivel bronce y obtiene puntuación  $\geq 7$  en el examen, pasaría al siguiente nivel (plata)
- ➔ Si el usuario tenía nivel plata y obtiene puntuación  $\geq 8$  en el examen, pasaría al siguiente nivel (oro)

El juego termina cuando el usuario alcanza el nivel oro en todos los temas. Las preguntas mostradas al usuario van acorde a un tema y un nivel.

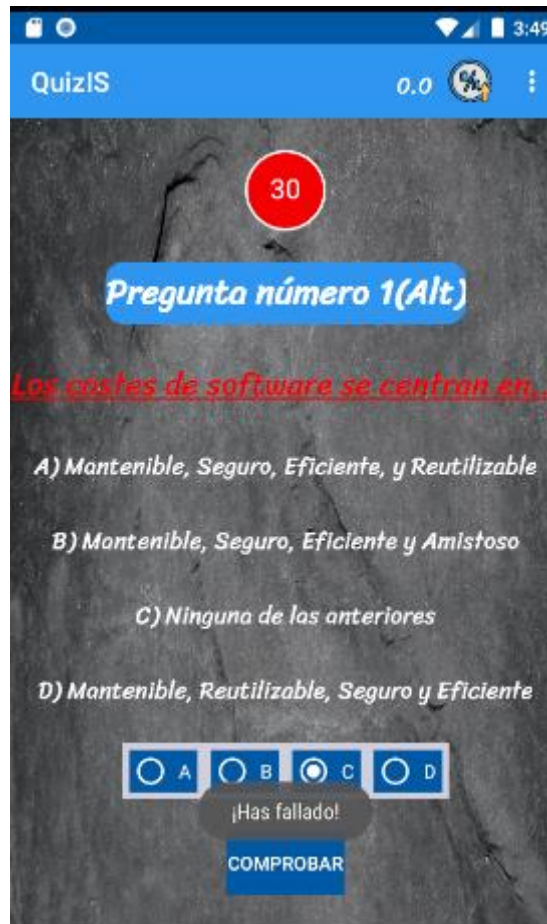


Figura 113: Toast<sup>6</sup> con fallo de pregunta y mostrada pregunta alternativa

<sup>6</sup> Toast: Mensaje que se muestra en pantalla durante unos segundos.



Figura 114: Resultado del examen al finalizarlo

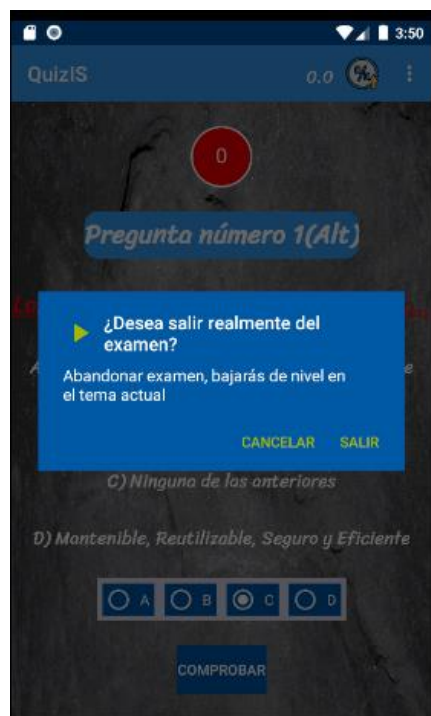


Figura 115: Pulsación en el botón de retroceder para abandonar examen

## 4. Funcionalidades Profesor

A continuación, se exponen las funcionalidades específicas para el usuario profesor.

### 4.1 Consultar estado alumno

En este apartado, el profesor podrá visualizar el estado del alumno. Se corresponde con la visualización del Ranking y, además, la información correspondiente a todas las estadísticas de cualquier alumno. El proceso que se ha de seguir es el mismo que en el expuesto en el punto 3.1 de las funcionalidades del alumno. La única diferencia de este apartado es que el profesor puede visualizar la información y estadísticas de cualquier alumno registrado en el sistema, en cambio, el alumno solo podía visualizar las suyas propias.

### 4.2 Crear pregunta y Repositorio

Cuando el usuario profesor accede a la pestaña del menú llamada “Juego”, podría visualizar una interfaz similar a la del alumno, pero esta vez sin niveles en cada tema. (Ver figura 116)



Figura 116: Interfaz de juego para profesor

Una vez pulsa sobre cualquiera de los temas, podrá visualizar una interfaz como el de la figura 117, que se corresponde con una pantalla de gestión donde podrá añadir preguntas al repositorio común, modificar preguntas ya añadidas, borrar preguntas del repositorio, crear, modificar y publicar exámenes...etc.



*Figura 117: Pantalla de gestión del profesor*

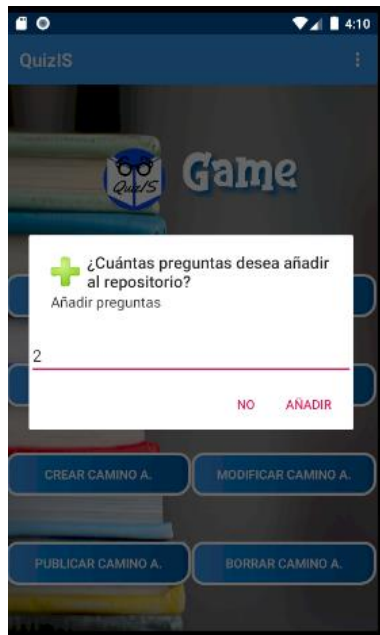
Al pulsar sobre el botón “Crear pregunta”, la aplicación cargaría una nueva pantalla donde se le piden los datos a introducir de la pregunta a guardar en el repositorio. (Ver figura 118)



*Figura 118: Pantalla para añadir una pregunta al repositorio común*

Una vez el profesor pulse sobre “Guardar”, esta pregunta será almacenada en el repositorio común de preguntas de todos los profesores y podrá ser modificada, borrada o utilizada para algún examen por cualquier profesor.

Para crear un “Repositorio” o conjunto de preguntas sin tener que estar pulsando continuamente en “Crear pregunta” existe el botón “Crear repositorio” que puede visualizarse en la figura 117, donde preguntará al usuario que cuantas preguntas desea añadir al repositorio (Ver figura 119) y el proceso seguiría el mismo que el de la figura 117 tantas veces como preguntas haya introducido el usuario.



*Figura 119: Pulsación sobre crear repositorio e introducir número de preguntas que va a añadir el profesor al repositorio común.*

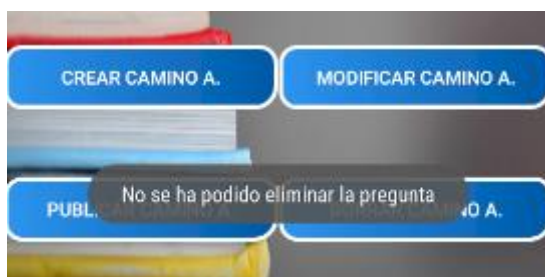
### **4.3 Borrar y Modificar pregunta**

En el caso en el que el usuario quiera eliminar una pregunta, pulsaría sobre el botón “Borrar pregunta”. La aplicación cargaría una pantalla con todas las preguntas disponibles en el repositorio común de profesores. (Ver figura 120)



*Figura 120: Repositorio común de preguntas*

Si el usuario pulsa sobre alguna de las preguntas, se preguntará si quiere eliminar esta pregunta, y confirmando, se eliminará definitivamente del repositorio. En el caso en el que la pregunta esté siendo usada en algún examen, esta no podrá eliminarse y mostrará un mensaje informativo al usuario. (Ver figura 121)

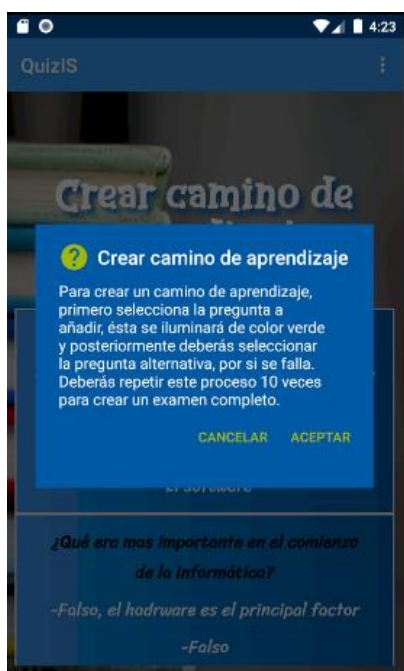


*Figura 121: Mensaje informativo al intentar eliminar una pregunta usada en un examen por otro profesor.*

Para modificar una pregunta, el procedimiento es similar al de crear una pregunta, la aplicación mostraría primero una pantalla igual que la de la figura “Repositorio de preguntas” y tras seleccionar una de las preguntas, mostraría otra pantalla similar al de la figura “Crear pregunta”

#### **4.4 Crear y modificar camino de aprendizaje o examen**

Cuando el usuario profesor quiera crear un examen, deberá pulsar sobre “Crear CA”. Esta opción le mostrará primero una pregunta para que el usuario indique a que nivel va destinado el examen (bronce, plata u oro) y posteriormente un aviso con la ayuda para que sepa crear el examen (Ver figura 122), Después el conjunto de preguntas del repositorio (Ver figura 123).



*Figura 122: Ayuda para crear un examen*



*Figura 123: Pantalla de creación de examen*

El proceso para crear un examen es el siguiente:

1. El usuario elige la pregunta y esta se ilumina en verde (Figura 123)
2. El usuario elige la pregunta alternativa (Por si falla la general)
3. Se añade la pregunta al árbol
4. Se sigue este procedimiento 10 veces hasta completar el examen

El repositorio debe contar con más de 11 preguntas para poder crear un examen y, además, el usuario puede abandonar este proceso en cualquier momento. Será avisado antes de abandonar por si realmente no quisiera hacerlo.

Para modificar un examen, este no tiene que haber sido publicado aun, por lo que cuando el usuario pulsa sobre “Modificar CA”, la aplicación cargará una pantalla donde se le muestra un resumen de los exámenes que aún no están publicados (Ver figura 124). Si pulsa sobre uno de estos, se cargaría la pantalla de modificación del examen. Primero mostraría una pestaña de ayuda para saber cómo realizar el proceso (Ver figura 125), y posteriormente se mostraría el examen a modificar y debajo el repositorio de preguntas (Ver figura 126). Cuando el usuario selecciona una de las preguntas del examen, la aplicación preguntará si quiere sustituir la pregunta principal o la alternativa, y posteriormente el usuario deberá seleccionar una pregunta del repositorio para sustituir la anterior. Finalmente debería pulsar sobre el botón de “Modificar”.

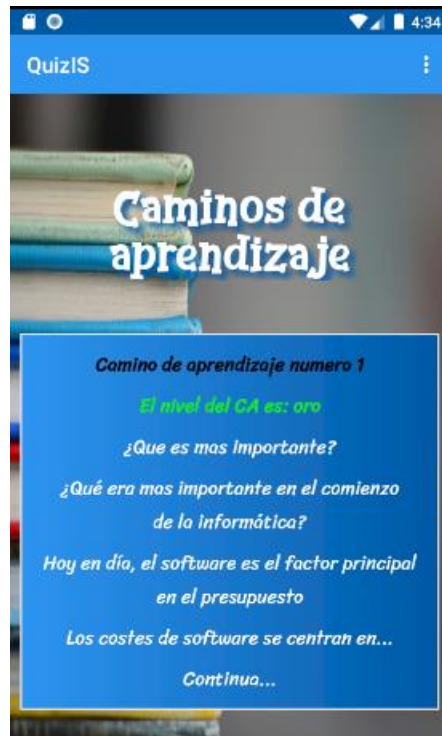


Figura 124: Resumen de exámenes no publicados

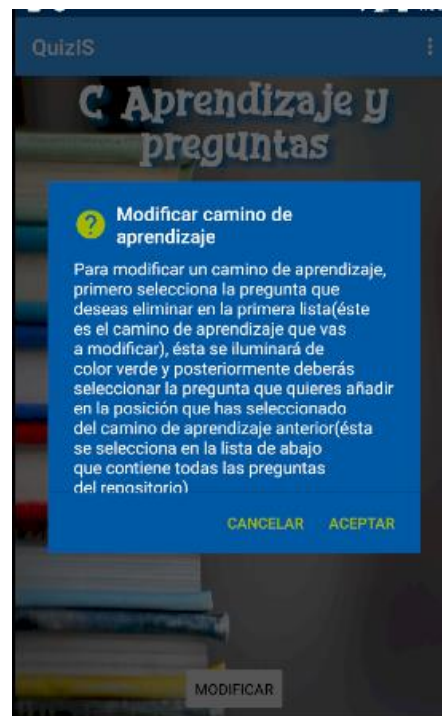


Figura 125: Ayuda sobre modificación de CA

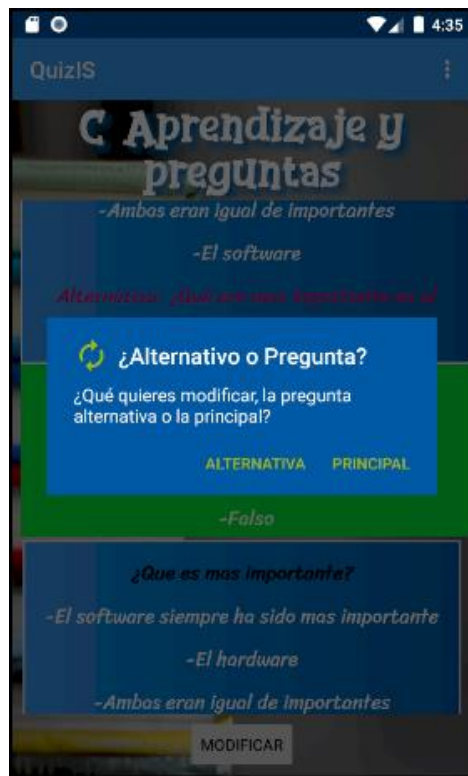


Figura 126: Modificación de examen

#### 4.5 Publicar y Borrar examen

Si el usuario quiere borrar un examen, debe pulsar sobre “Borrar CA”, esta opción le llevará a una pantalla similar a la de la figura x, con un resumen de todos los exámenes, publicados y no publicados. Si pincha sobre alguno de esos, la aplicación cargará una pantalla con la información completa del examen (Ver figura 127) y deberá pulsar sobre “Eliminar” para borrarlo del sistema.



*Figura 127: Borrar examen*

Para publicar un examen, el proceso es similar al anterior, el único cambio es que, en el momento de mostrar los exámenes, en resumen, solo mostrará los que aún no han sido publicados.

## Anexo II. Guía de instalación

Primero se describen los pasos para la instalación de la API y base de datos y posteriormente se describen los pasos para la instalación de la aplicación móvil.

### 1. Instalación y arranque del servidor y Base de datos

#### 1.1 Instalación

Para la instalación de la API y poder arrancar el servidor debemos tener instalado Java en su versión 1.7, Maven y el IDE Spring Tools Suite, si es posible en la versión 4. Se pueden conseguir Java y Maven de la página web de Oracle y, además, tras instalarlas debemos configurar las variables de entorno correspondientes para que sean reconocidas por el sistema. El proceso de instalación y configuración de Java y Maven puede seguirse desde este enlace: <http://www.laurafolgado.es/blog/2015/10/22/instalar-maven-en-windows/>

Para comprobar la versión de Java instalada en el sistema ejecutamos:

```
#java -version
```

El siguiente paso sería clonar el proyecto de la API desde GitHub o el repositorio donde se encuentre y copiar este proyecto al workspace correspondiente.

El repositorio público donde se puede encontrar el proyecto de la API es el siguiente:

[https://github.com/RubenGarciaMateos/TFG\\_QUIZIS](https://github.com/RubenGarciaMateos/TFG_QUIZIS)

Utilizando Git bash de Windows, situarse sobre el directorio de trabajo y ejecutar:

```
#git clone [url de github de la API]
```

Dentro de Spring tool suite, archivo->importar->Maven->proyecto Maven existente y seleccionamos el proyecto de la API.

Descargar XAMPP versión v3.2.4 a ser posible, sobre todo la parte del servidor Apache y MySQL. Se arranca el servidor de Apache y la base de datos MySQL (Ver figura 128)

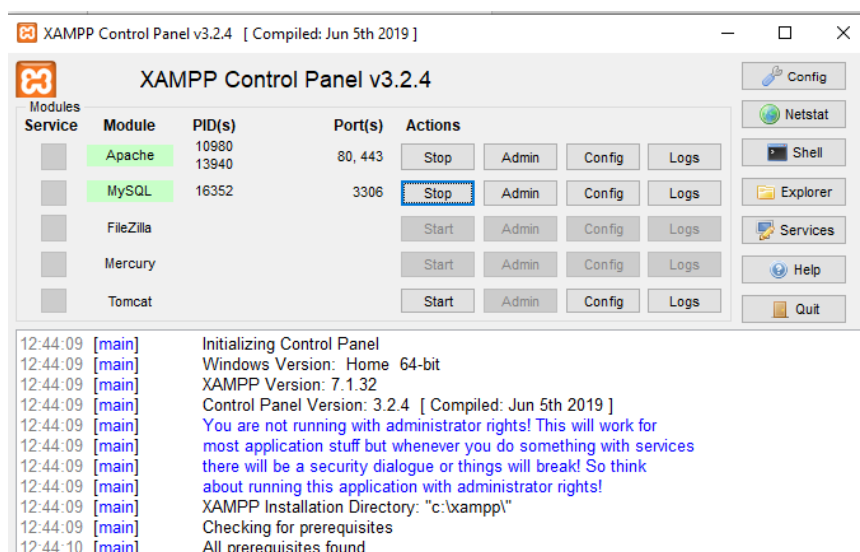


Figura 128: Arranque de MySQL y Apache desde XAMPP

Acceder a <http://localhost/phpmyadmin/> desde cualquier navegador, y crear una nueva base de datos llamada “quizis” para que sea reconocida por el servidor e Hibernate (ver figura 129)



Figura 129: Creación de base de datos “quizis” en XAMPP

## 1.2 Arranque

Ya que por defecto Spring trae un Tomcat embebido, se arrancará el servidor con la aplicación Spring escuchando en el puerto 8080 de la máquina local.

Para generar el modelo de datos de la aplicación e introducir unos datos iniciales para poder trabajar con ellos, se indicará mediante una serie de propiedades que Hibernate genere el modelo de datos y ejecute un script de inicialización con una serie de usuarios, exámenes y preguntas de prueba.

Para ellos abrir el fichero propiedades del proyecto en la ruta `\src\main\resources\application.properties` y cambiar las propiedades,

`spring.datasource.initialization-mode=never` por `always`

`spring.jpa.hibernate.ddl-auto=none` por `create`

Este fichero puede visualizarse en la siguiente figura:

```
application.properties X
1
2 #Conexión con BBDD, datasource y JPA
3 spring.datasource.url=jdbc:mysql://localhost:3306/quizis?serverTimezone=UTC
4 spring.datasource.username=root
5 spring.datasource.password=root
6 spring.jpa.database-platform=org.hibernate.dialect.MySQL57Dialect
7 #spring.datasource.platform=data-oracle.sql
8 spring.datasource.data=classpath:data-oracle.sql
9 #Propiedad para ejecutar script de insercion de datos
10 spring.datasource.initialization-mode=never
11 #Propiedad para crear esquema de bd
12 spring.jpa.hibernate.ddl-auto=none
13 spring.jpa.show-sql=true
14 logging.level.org.hibernate.SQL=debug
15 spring.servlet.multipart.enabled=true
16 spring.jpa.hibernate.naming.physical-strategy=org.hibernate.boot.model.naming.PhysicalNamingStrategyStandardImpl
17
18
19 #DATOS ACCESO JELASTIC
20 #spring.datasource.url=jdbc:mysql://node27652-env-3569932.es-1.axarnet.cloud/quizis
21 #spring.datasource.username=root
22 #spring.datasource.password=AZIebp57370
23
24 #spring.jackson.serialization.fail-on-empty-beans=false
25
```

Figura 130: Propiedades para generar el modelo de datos e introducir datos iniciales en la BD.

El script que ejecutaría Hibernate con el cambio de propiedades anteriormente realizado sería el que se encuentra en la ruta `\src\main\resources\data-oracle.sql`.

Para finalizar y arrancar el servidor con los datos cargados guardar cambios y botón derecho sobre el proyecto, Run As → Spring Boot App.

Si todo ha salido correctamente en la consola de Spring se podrá visualizar el mensaje al final:

Tomcat started on port(s): 8080 (http) with context path

Started QuizisApplication in 8.783

El resultado puede verse en la siguiente figura 131.

```
Problems Javadoc Declaration Search Console X Diagrams Servers Boot Dashboard
TFG QUIZIS - QuizisApplication [Spring Boot App] C:\Program Files\Java\jre1.8.0_221\bin\javaw.exe (2 jun. 2020 21:52:58)
2020-06-02 21:53:05.392 INFO 11768 --- [ restartedMain] org.hibernate.Version : HHH000412: Hibernate Core [5.3.12.Final]
2020-06-02 21:53:05.395 INFO 11768 --- [ restartedMain] org.hibernate.cfg.Environment : HHH000206: hibernate.properties not found
2020-06-02 21:53:05.671 INFO 11768 --- [ restartedMain] org.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotations [5.0.4.Final]
2020-06-02 21:53:05.941 INFO 11768 --- [ restartedMain] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect.MySQL57Dialect
2020-06-02 21:53:07.415 INFO 11768 --- [ restartedMain] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for persistence unit 'default'
2020-06-02 21:53:08.377 INFO 11768 --- [ restartedMain] o.s.s.web.DefaultSecurityFilterChain : Creating filter chain: any request, [org.springframework.security.web.
2020-06-02 21:53:08.593 INFO 11768 --- [ restartedMain] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'applicationTaskExecutor'
2020-06-02 21:53:08.594 WARN 11768 --- [ restartedMain] web.ConfigurationJpaWebMvcConfiguration : spring.jpa.open-in-view is enabled by default. Therefore, database que
2020-06-02 21:53:08.707 INFO 11768 --- [ restartedMain] o.s.b.d.a.OptionalLiveReloadServer : LiveReload server is running on port 35729
2020-06-02 21:53:08.757 INFO 11768 --- [ restartedMain] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) with context path ''
2020-06-02 21:53:08.768 INFO 11768 --- [ restartedMain] com.backend.QuizisApplication : Started QuizisApplication in 8.783 seconds (JVM running for 10.171)
```

Figura 131: Servidor Tomcat con aplicación Spring arrancado en el puerto 8080

No olvidar que, si se para la aplicación, volver a dejar el fichero de propiedades anteriormente modificado como estaba, para no estar creando el modelo de datos cada vez que se arranque la aplicación.

## 2. Instalación y arranque de la aplicación móvil

Para poder instalar la aplicación móvil es necesario el IDE Android Studio, y el ejecutable Ngrok para crear el túnel de acceso a la API desde la aplicación.

### 2.1 Android Studio

Descargar e instalar Android Studio recomendable versión 3.5.1, aunque cualquiera es válido, desde la web <https://developer.android.com/studio?hl=es>.

Configuración por defecto, en el instalador también seleccionar que se quiere instalar emulador o Android Virtual Device. Este instalador trae las herramientas del SDK necesarias para trabajar con la aplicación.

El siguiente paso sería clonar el proyecto de la aplicación desde GitHub o el repositorio donde se encuentre y copiar este proyecto al workspace que Android haya creado por defecto, suele estar en C:\Usuarios\[UsuarioActual]\AndroidProjects

El repositorio público donde se puede encontrar el proyecto de la API es el siguiente:

[https://github.com/RubenGarciaMateos/TFG\\_QUIZIS\\_APP](https://github.com/RubenGarciaMateos/TFG_QUIZIS_APP)

Utilizando Git bash de Windows, situarse sobre el directorio de trabajo anterior y ejecutar:

```
#git clone [url de GitHub de la API]
```

Desde Android Studio, File->Open y seleccionar el proyecto del workspace clonado anteriormente.

Una vez haya compilado todos los archivos, configurar un dispositivo móvil desde el emulador Android Virtual Device. Este dispositivo representa virtualmente un teléfono móvil como si del propio se tratara.

➔ Seleccionar Tools->AVD Manager

Podrá visualizarse algo similar a lo que se ve en la figura 132:

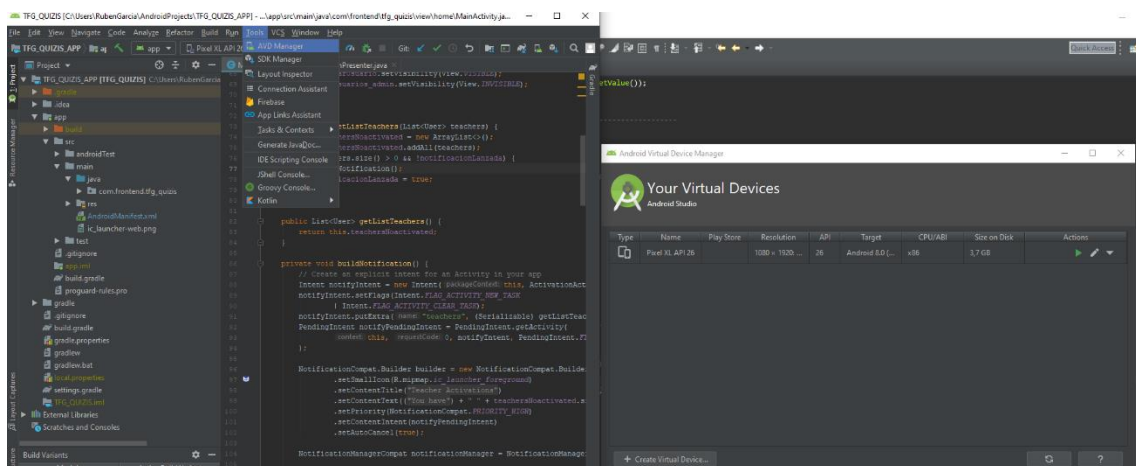


Figura 132: Configuración de dispositivo virtual para instalar la aplicación

En la pestaña de la derecha se tiene el asistente para añadir un dispositivo virtual, pulsar sobre “Create Virtual Device” abajo a la izquierda y configurar un dispositivo como el de la figura 133. Modelo Píxel XL con la API de desarrollo 26. (Debería funcionar con cualquier dispositivo, pero por evitar problemas se generará de esta forma)

Arrancar el dispositivo pulsando el botón “Play” de color verde, una vez termine de cargar el sistema operativo se podrá visualizar la interfaz de un smartphone en la pantalla. (Ver figura 133)

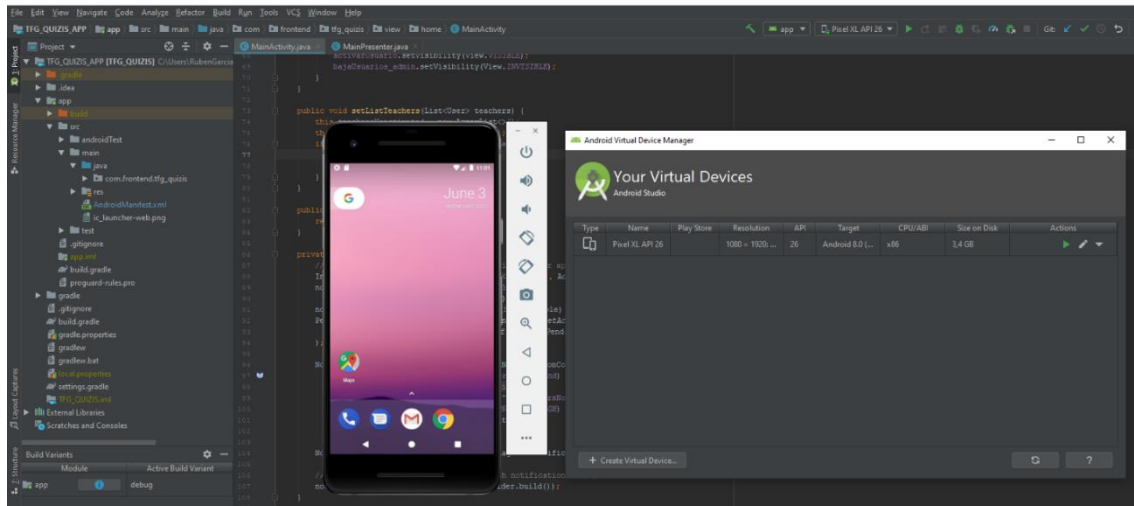


Figura 133: Dispositivo virtual arrancado y en espera de instalar la aplicación

## 2.1 Ngrok

Como ya está arrancado el dispositivo virtual, lo único que quedaría pendiente es establecer un túnel de comunicación entre la API y el dispositivo móvil virtual. Con Ngrok se puede realizar fácilmente.

- ➔ Descargar Ngrok de la web <https://ngrok.com/download>
- ➔ Ejecutar el archivo “.exe” de descarga

En la consola que aparece ejecutar el comando:

```
#ngrok http 8080
```

Este comando lo que permite es crear el túnel a la máquina local que escucha en el puerto 8080, por lo que cualquier usuario que acceda a la url que devuelve Ngrok tendrá acceso a la API. Puede visualizarse la URL de acceso que ha creado Ngrok en la siguiente figura:

```

Seleccionar C:\Users\Ruben Garcia\Desktop\ngrok.exe - ngrok http 8080
ngrok by @inconshreveable

Session Status      online
Session Expires    7 hours, 59 minutes
Version             2.3.35
Region              United States (us)
Web Interface       http://127.0.0.1:4040
Forwarding           http://67d78305ddc6.ngrok.io -> http://localhost:8080
Forwarding          https://67d78305ddc6.ngrok.io -> http://localhost:8080

Connections
  ttl    opn    rt1    rt5    p50    p90
   0     0     0.00  0.00  0.00  0.00

```

Figura 134: Túnel creado con Ngrok y url de acceso al puerto 8080

El último paso para realizar es cambiar la url de acceso a la API en el código de la aplicación Android, por lo que se debe acudir a los dos ficheros de Strings de la aplicación y cambiar esta URL estableciendo la que ha facilitado Ngrok.

### Cambiar en los dos ficheros

Estos ficheros de Strings de la aplicación se encuentran en:

`\app\src\main\res\values -> fichero strings.xml`

`\app\src\main\res\values-es -> fichero strings.xml`

Cambiar la propiedad “ip\_acceso\_backend” con el valor facilitado por Ngrok anteriormente. (Ver figura 135)

```

Edit translations for all locales in the translations editor.
...
197 <string name="publicacion_correcta">Se ha publicado correctamente el CA</string>
198 <string name="publicacion_incorrecta">No se ha podido publicar el CA</string>
199 <string name="actualizacion_correcta_ca">Se ha actualizado correctamente el CA</string>
200 <string name="actualizacion_incorrecta_ca">No se ha podido actualizar el CA</string>
201 <string name="no_eliminado_pregunta">No se ha podido eliminar la pregunta</string>
202 <string name="actualizacion_pregunta_correcta">Se ha actualizado correctamente la pregunta</string>
203 <string name="actualizacion_pregunta_incorrecta">No se ha podido actualizar la pregunta</string>
204 <string name="level_of_ca_text">El nivel del CA es</string>
205 <string name="continue_text">Continuar</string>
206 <string name="save_CA_correct">Camino de aprendizaje guardado correctamente</string>
207 <string name="save_CA_error">No se ha podido guardar el Camino de aprendizaje</string>
208 <string name="question_borrada_correct">Se ha borrado correctamente la pregunta</string>
209 <string name="volver_text">Volver</string>
210 <string name="no_existen_preguntas">No existen preguntas</string>
211 <string name="description_question_empty">No te sientas triste al ver que no hay ninguna pregunta creada?, añade preguntas para poder crear exámenes</string>
212 <string name="no_examenes_para_mostrar">No hay exámenes para mostrar</string>
213 <string name="description_ca_empty">No tienes disponible ningún examen para visualizar en esta sección.</string>
214 <!-- IP PARA ACCESO A BACKEND -->
215 <string name="ip_acceso_backend">http://63d629ca2b.ngrok.io/</string>
216 <!-- MENU -->
217 <string name="perfil_menu">Perfil</string>
218 <string name="perfil_jugar">Jugar</string>
219 <string name="perfil_ranking">Ranking</string>
220 <string name="perfil_ayuda">Ayuda</string>
221 <!-- PANTALLA ACTUALIZAR PREGUNTA -->

```

Figura 135: Ficheros Strings con propiedad de ip a cambiar

Para ejecutar la aplicación, pulsar sobre Run->Run app y seleccionar el dispositivo virtual creado anteriormente. La aplicación arrancará en el emulador configurado.

Usuarios iniciales cargados para pruebas en el apartado de instalación de la API con los que se puede iniciar sesión:

**Alumno:**

→ Usuario: rgarcia, Password: 02551077d

**Profesor:**

→ Usuario: jgarcia, Password: 12345678

**Administrador:**

→ Usuario: admin, Password: istrador

### **Anexo III. Preguntas de evaluación completas**

A continuación, se ilustrarán las preguntas de evaluación completas realizadas en el apartado 8 de este documento.

Como se introdujo, existen 3 bloques de preguntas, el bloque 1 contenía preguntas más genéricas sobre los encuestados, el bloque 2 opinión personal sobre las distintas funcionalidades de la aplicación dependiendo del rol que el que realizasen la encuesta y el bloque 3 cambios propuestos por el encuestado y valoración general.

#### **1. Primer bloque**

El primer bloque de preguntas que se muestran al usuario son preguntas relacionadas con la edad, relación con la universidad o docencia, y sexo, es decir, más generales.

# QUIZIS

## Valoración de QuizIS

\*Obligatorio

Sección sin título

Introduce tu nombre \*

Tu respuesta

Introduce tu edad \*

Tu respuesta

Sexo \*

Hombre

Mujer

¿Estás relacionado con la universidad? \*

Sí, soy profesor

Si, soy alumno

No

¿Vas a responder como profesor, alumno o administrador? \*

Elige ▼

Figura 136: Preguntas bloque 1

## 2. Segundo bloque

El segundo bloque muestra preguntas relacionadas con el rol del usuario que haya decidido al realizar la encuesta en la última pregunta del bloque 1.

### 2.1 Preguntas de evaluación para rol registrado como alumno

**Alumno**

valora de 0 a 5 las siguientes opciones

**Inicio de aplicación**  
Inicia la aplicación, cuando tengas la pantalla principal cargada, pulsa sobre el botón "Registro". Rellena los datos y regístrate como alumno.

Indica tu nivel de satisfacción con la pantalla de registro \*

Muy malo   0   1   2   3   4   5   Muy bueno

○   ○   ○   ○   ○   ○

**Login**  
Ahora si todo ha ido correctamente, habrás vuelto a la pantalla de login. Loguéate con esa cuenta que has creado. Ahora mismo estarás viendo la pantalla principal o home de tu usuario, con tus datos, fotografía de perfil, y mas opciones...

Indica tu satisfacción con la pantalla de login \*

Muy malo   0   1   2   3   4   5   Muy bueno

○   ○   ○   ○   ○   ○

Indica tu satisfacción con la pantalla de Home \*

Muy malo   0   1   2   3   4   5   Muy bueno

○   ○   ○   ○   ○   ○

Figura 137: Preguntas alumno parte I

**Perfil**  
 Edita tu perfil, después entra en la pantalla de editar imagen de perfil y cambialá también(La imagen que quieras subir debe estar en la memoria interna del teléfono, si este paso no consigues realizarlo, sáltalo). Tras realizar estos pasos, volverás a la pantalla de inicio y verás tus datos actualizados.

Indica tu satisfacción con la facilidad y usabilidad del proceso realizado \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu satisfacción con la pantalla de editar perfil y añadir una nueva imagen de perfil \*

0 1 2 3 4 5

Muy malo       Muy bueno

**Ranking alumnos**  
 Pulsa sobre el menú y accede a la opción de Ranking de alumnos, podrás visualizar el ranking de los mejores alumnos por porcentaje de juego completado. Además si pulsas sobre un alumno del ranking, podrás acceder a una pantalla que muestra los niveles que tiene en cada tema, y si dejas pulsado sobre tu usuario dentro del ranking, podrás visualizar tus estadísticas (exámenes suspensos, aprobados, preguntas falladas, tiempo medio de respuesta en los exámenes...etc).

Indica tu grado de satisfacción con la pantalla de ranking de alumnos \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con las pantallas de estadísticas del alumno \*

0 1 2 3 4 5

Muy malo       Muy bueno

Figura 137: Preguntas alumno parte II

Indica tu grado de satisfacción con el modo de visualización de los exámenes y preguntas suspensas \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

**Pestaña Ayuda**  
Pulsa ahora en el menú sobre la pestaña "Ayuda" y navega por las distintas pantallas hasta que llegues al final.

---

Indica tu grado de satisfacción con las distintas pantallas de ayuda \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

Indica tu grado de satisfacción con la ayuda que crees que se proporciona (Si la información te resulta útil) \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

**Pestaña Juego**  
Ahora en el menú ve a la pestaña de "Juego", primero te aparece una pantalla con todos los temas disponibles sobre los que puedes realizar exámenes. Pulsa sobre el tema 1, por ejemplo, ya que es el que tendrá preguntas añadidas, y realiza un examen.

---

Indica tu grado de satisfacción con la pantalla que muestra los temas y niveles \*

0 1 2 3 4 5

Muy malo       Muy bueno

Figura 138: Preguntas alumno parte III

Indica tu grado de satisfacción con la pantalla que va mostrando las preguntas del examen. \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con la última pestaña que muestra los resultados obtenidos junto con un sonido \*

0 1 2 3 4 5

Muy malo       Muy bueno

**Vuelta**  
Ahora como ya has realizado un examen, podrás ver mejor tus estadísticas en la pantalla de Ranking nuevamente (Pruébalo). Cierra sesión y vuelve a la pantalla de login

Indica tu grado de satisfacción con este proceso \*

0 1 2 3 4 5

Muy malo       Muy bueno

Figura 139: Preguntas alumno parte IV

## 2.2 Preguntas de evaluación para rol registrado como profesor

**Profesor**

valora de 0 a 5 las siguientes opciones

**Inicio de aplicación**  
Inicia la aplicación, cuando tengas la pantalla principal cargada, pulsa sobre el botón "Registro". Rellena los datos y regístrate como alumno.

Indica tu nivel de satisfacción con la pantalla de registro \*

Muy malo   0   1   2   3   4   5   Muy bueno

○   ○   ○   ○   ○   ○

**Login**  
Ahora si todo ha ido correctamente, habrás vuelto a la pantalla de login. Loguéate con esa cuenta que has creado. Ahora mismo estarás viendo la pantalla principal o home de tu usuario, con tus datos, fotografía de perfil, y mas opciones...

Indica tu satisfacción con la pantalla de login \*

Muy malo   0   1   2   3   4   5   Muy bueno

○   ○   ○   ○   ○   ○

Indica tu satisfacción con la pantalla de Home \*

Muy malo   0   1   2   3   4   5   Muy bueno

○   ○   ○   ○   ○   ○

Figura 140: Preguntas profesor parte I

**Perfil**  
 Edita tu perfil, después entra en la pantalla de editar imagen de perfil y cambialá también. Tras realizar estos pasos, volverás a la pantalla de inicio y verás tus datos actualizados.

Indica tu satisfacción con la facilidad y usabilidad del proceso realizado \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu satisfacción con la pantalla de editar perfil y añadir una nueva imagen de perfil \*

0 1 2 3 4 5

Muy malo       Muy bueno

**Ranking alumnos**  
 Pulsa sobre el menú y accede a la opción de Ranking de alumnos, podrás visualizar el ranking de los mejores alumnos por porcentaje de juego completado. Además si pulsas sobre un alumno del ranking, podrás acceder a una pantalla que muestra los niveles que tiene en cada tema, y si dejas pulsado sobre tu usuario dentro del ranking, podrás visualizar las estadísticas del usuario (exámenes suspensos, aprobados, preguntas falladas, tiempo medio de respuesta en los exámenes...etc).

Indica tu grado de satisfacción con la pantalla de ranking de alumnos \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con las pantallas de estadísticas del alumno \*

0 1 2 3 4 5

Muy malo       Muy bueno

*Figura 141: Preguntas profesor parte II*

Indica tu grado de satisfacción con el modo de visualización de los exámenes y preguntas suspensas \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

**Pestaña Ayuda**  
Pulsa ahora en el menú sobre la pestaña "Ayuda" y navega por las distintas pantallas hasta que llegues al final.

Indica tu grado de satisfacción con las distintas pantallas de ayuda \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

Indica tu grado de satisfacción con la ayuda que crees que se proporciona (Si la información te resulta útil) \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

**Pestaña Juego**  
Ahora en el menú ve a la pestaña de "Juego", primero te aparece una pantalla con todos los temas disponibles sobre los que puedes crear preguntas y exámenes. Pulsa sobre el tema 1, por ejemplo, y crea una pregunta y un examen (Tiene que haber más de 10 preguntas añadidas al repositorio, pero ya habrá algunas añadidas).

Indica tu grado de satisfacción con la pantalla que muestra los temas \*

0 1 2 3 4 5

Muy malo       Muy bueno

*Figura 142: Preguntas profesor parte III*

Indica tu grado de satisfacción con la pantalla de creación de pregunta \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

Indica tu grado de satisfacción con la pantalla de creación de examen \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

Indica tu grado de satisfacción con la facilidad en la creación de un examen \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

**Pestaña Juego**  
 Ahora pulsa en el botón de actualizar pregunta, Busca la pregunta que añadiste anteriormente e introduce nuevos datos para la pregunta. Si quieres volver a visualizarla con los datos actualizados deberías entrar otra vez en Borrar Pregunta por ejemplo.

---

Indica tu grado de satisfacción con la pantalla que muestra la lista de preguntas del repositorio \*

0 1 2 3 4 5

Muy malo       Muy bueno

---

Indica tu grado de satisfacción con la pantalla de actualizar pregunta \*

0 1 2 3 4 5

Muy malo       Muy bueno

*Figura 143: Preguntas profesor parte IV*

Pestaña Juego  
Pulsa ahora sobre actualizar examen o camino de aprendizaje. Modifica según la ayuda que te aparece un examen. Sólo se pueden modificar exámenes no publicados.

Indica tu grado de satisfacción con la pantalla de actualizar un examen \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con la facilidad para actualizar un examen \*

0 1 2 3 4 5

Muy malo       Muy bueno

Pestaña Juego  
Pulsa sobre Publicar Camino de aprendizaje, te debería de aparecer el examen que has creado anteriormente, publícalo y cuando lo hayas hecho, Pulsa sobre borrar Camino de aprendizaje, y ese exámen que has creado elimínalo.

Indica tu grado de satisfacción con la pantalla de publicar camino de aprendizaje o examen \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con la pantalla de eliminar un camino de aprendizaje o examen \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu satisfacción con el proceso \*

0 1 2 3 4 5

Muy malo       Muy bueno

Figura 144: Preguntas profesor parte V

## 2.3 Preguntas de evaluación para rol administrador

**Administrador**

valora de 0 a 5 las siguientes opciones

**Login**  
Loguéate con la cuenta de administrador. Ahora mismo estarás viendo la pantalla principal o home de tu usuario, con tus datos, fotografía de perfil, y mas opciones...

Indica tu satisfacción con la pantalla de login \*

0 1 2 3 4 5

Muy malo       Muy bueno

Pulsa sobre "Eliminar usuarios" Introduce el nombre de un usuario a eliminar por ejemplo, jgarcia. Loguéate con la cuenta de administrador. Ahora mismo estarás viendo la pantalla principal o home de tu usuario, con tus datos, fotografía de perfil, y mas opciones...

Indica tu grado de satisfacción con la pantalla y la interacción con eliminar usuarios. \*

0 1 2 3 4 5

Muy malo       Muy bueno

**Perfil**  
Edita tu perfil, después entra en la pantalla de editar imagen de perfil y cambialá también. Tras realizar estos pasos, volverás a la pantalla de inicio y verás tus datos actualizados.

Indica tu satisfacción con la facilidad y usabilidad del proceso realizado \*

0 1 2 3 4 5

Muy malo       Muy bueno

Figura 145: Preguntas administrador parte I

Indica tu satisfacción con la pantalla de editar perfil y añadir una nueva imagen de perfil \*

0 1 2 3 4 5

Muy malo       Muy bueno

**Ayuda**  
Pulsa ahora en el menú sobre la pestaña "Ayuda" y navega por las distintas pantallas de ayuda hasta que llegues al final.

Indica tu grado de satisfacción con las pantallas de ayuda \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con la ayuda que crees que se proporciona(Si es útil) \*

0 1 2 3 4 5

Muy malo       Muy bueno

*Figura 146: Preguntas administrador parte II*

### 3. Tercer bloque

El tercer bloque de preguntas consta de una serie de preguntas sobre evaluación global de la aplicación y propuestas de mejoras y cambios.

Indica tu grado de satisfacción con los colores e interfaz de la aplicación. \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con las animaciones de la aplicación \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción con la facilidad de uso de la aplicación. \*

0 1 2 3 4 5

Muy malo       Muy bueno

Indica tu grado de satisfacción general. \*

0 1 2 3 4 5

Muy malo       Muy bueno

¿Qué mejorarías?

Tu respuesta \_\_\_\_\_

¿Qué echas en falta?

Tu respuesta \_\_\_\_\_

Figura 147: Preguntas generales bloque 3