



UNIVERSIDAD DE VALLADOLID

E.T.S.I. TELECOMUNICACIÓN

TRABAJO FIN DE GRADO

GRADO EN INGENIERÍA DE TECNOLOGÍAS  
DE TELECOMUNICACIÓN

**DESARROLLO DE LA APLICACIÓN CLIENTE  
DEL ENTORNO VIRTUAL *AIM-MOBILE*  
*LEARNING* PLATFORM EN ANDROID**

Autor:

**D. Samir Gemali Pinilla**

Tutora:

**Dña. Noemí Merayo Álvarez**

Valladolid, 12 de Mayo de 2017



---

**TÍTULO:**                    **Desarrollo de la aplicación cliente del entorno virtual AIM-Mobile Learning Platform en Android**

**AUTOR:**                    **D. Samir Gemali Pinilla**

**TUTORA:**                    **Dña. Noemí Merayo Álvarez**

**DEPARTAMENTO:**   **Teoría de la Señal y Comunicaciones e Ingeniería Telemática**

---

**TRIBUNAL**

---

**PRESIDENTE:**   **D. Ignacio de Miguel Jiménez**

**VOCAL:**                    **Dña. Noemí Merayo Álvarez**

**SECRETARIO:**   **D. Juan Carlos Aguado Manzano**

**SUPLENTE:**           **D. Ramón J. Durán Barroso**

**SUPLENTE:**           **Dña. Patricia Fdez del Reguero**

---

**FECHA:**                    **19 de Mayo de 2017**

**CALIFICACIÓN:**

---



## **Resumen del TFG**

Las herramientas e-learning y m-learning se están convirtiendo en un apoyo esencial para las nuevas metodologías docentes, por lo que dentro del Grupo de Comunicaciones Ópticas de la Universidad de Valladolid ha desarrollado el entorno de aprendizaje virtual e-learning llamada AIM-Mobile Learning Platform. Esta plataforma pretende que un alumno pueda acceder a los eventos, a la lectura de los libros electrónicos diseñados, a las preguntas lanzadas por el profesor y a los datos y estadísticas de su evaluación continua en las asignaturas en las que esté matriculado y de una forma rápida y sencilla. La plataforma virtual de aprendizaje consta de una aplicación para el usuario profesor y otra para el usuario alumno, esta última desarrollada en un entorno multiplataforma. Este TFG se centra en desarrollar la aplicación del alumno en código Android nativo. Además, también se implementarán un conjunto de nuevas funcionalidades que también se están programando para la aplicación multiplataforma con la finalidad de otorgarle mayor flexibilidad y potencialidad de cara a su difusión y uso en diferentes entornos y disciplinas educativas.

## **Palabras clave**

Android, AIM-Mobile Learning Platform, aplicaciones móviles, m-learning, e-learning.

## **ABSTRACT**

E-learning and m-learning tools are becoming an essential support for the new teaching methodologies, so the Optical Communications Group of the University of Valladolid has developed a virtual e-learning environment called AIM-Mobile Learning Platform. This platform permits students to access events, learning electronic books, questions designed and activated by teachers and data and marks of students achieved during the continuous assessment in different subjects in a fast and easy way. The virtual learning platform consists of one application for the teacher and another one for the student, the last one developed using a multiplatform environment. This TFG is focus on developing the student application in native Android code. Furthermore, it will be implemented a set of new functionalities, also programmed in the multiplatform version, with the aim to provide more flexibility and potentiality to the learning platform so that it can be diffused and used in other educational disciplines and environments.

## **Keywords**

Android, AIM-Mobile Learning Platform, aplicaciones móviles, m-learning, e-learning.

## **Agradecimientos**

Quiero dar gracias a los profesores del grado, que me han formado hasta este punto en el casi soy graduado, en especial a mi tutora Noemi, que ha conseguido que este proyecto saliera adelante con unos resultados bastante satisfactorios.

También quiero agradecer a mi compañero del proyecto AIM-Mobile Learning platform, Alberto, que me ha ayudado mucho cuando tenía cualquier duda, y a mi compañero y amigo Jorge, que hizo la aplicación Evalcoa para Android, que también me ha ayudado sobre todo al empezar a programar.

Por último, agradecer a mi familia y amigos por el apoyo durante todos estos años, ya que sin ellos no habría llegado hasta este punto en el grado.





# ÍNDICE

1. INTRODUCCIÓN.....	1
1.1. Motivación.....	1
1.2. Objetivos.....	2
1.3. Estructura de la memoria.....	2
2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO .....	5
2.1. Introducción.....	5
2.2. Entorno de aprendizaje virtual AIM-Mobile Learning platform.....	5
2.2.1 Herramientas de desarrollo de la aplicación del profesor.....	6
2.2.2 Herramientas de desarrollo de la aplicación del alumno .....	7
2.3. Herramientas empleadas en el desarrollo de la nueva aplicación .....	7
2.4. Metodología.....	10
2.5. Conclusiones.....	11
3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE .....	13
3.1. Introducción.....	13
3.2. Desarrollo de las vistas de la aplicación cliente .....	13
3.3. Peticiones GET a la API del entorno virtual .....	14
3.4. Desarrollo de funcionalidades en la aplicación cliente .....	15
3.5. Peticiones mediante el método POST y desarrollo de funcionalidades .....	22
3.6. Conclusiones.....	26

4. DESARROLLO DE NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE .....	27
4.1. Introducción.....	27
4.2. Funcionalidad para seleccionar el servidor de alojamiento del entorno de aprendizaje.....	27
4.3. Funcionalidad para restablecer contraseña .....	29
4.4. Manual de ayuda de la aplicación al alumno.....	30
4.5. Visualización de las notas y de las preguntas respondidas.....	32
4.6. Funcionalidad crear preguntas.....	33
4.7. Conclusiones.....	36
5. CONCLUSIONES Y LÍNEAS FUTURAS .....	37
5.1. Conclusiones.....	37
5.2. Líneas futuras .....	38
6. BIBLIOGRAFÍA .....	41

# ÍNDICE DE FIGURAS

Figura 1. Diseño global del entorno virtual de aprendizaje AIM-Mobile Learning Platform desde el punto de vista del desarrollo web. ....	6
Figura 2. Entorno de desarrollo de Android Studio.....	8
Figura 3. Entorno del editor Notepad .....	9
Figura 4. Entorno de la extensión Advanced Rest Client para Google Chrome .....	9
Figura 5. Ejemplo de vista creada en Android Studio.....	14
Figura 6. Pantalla inicio aplicación .....	16
Figura 7. Menú Principal .....	17
Figura 8. Ejemplo de pestañas.....	<b>¡Error! Marcador no definido.</b>
Figura 9. Avisos de una asignatura y datos de un aviso .....	19
Figura 10. Libros de una asignatura y un índice de un libro .....	20
Figura 11. Vista de cada libro html .....	20
Figura 12. Preguntas de una asignatura y gráfica con las notas conseguidas y nota media .....	21
Figura 13. Vista para registrarse y diálogos con errores varios.....	23
Figura 14. Vista de las preguntas con contraseña y un error al dar una contraseña errónea .....	24
Figura 15. Vista para responder preguntas .....	25
Figura 16. Funcionalidad para elegir el servidor .....	28
Figura 17. Cuadro de diálogo de error en la conexión y cuadro para guardar la dirección del servidor .....	28
Figura 18. Vistas para restablecer la contraseña e introducir el nombre de usuario .....	29
Figura 19. Cuadro de fallo cuando el estudiante no está registrado y cuadro con el correo enviado .....	30
Figura 20. Botones de ayuda .....	30
Figura 21. Ayudas de la aplicación .....	31
Figura 22. Preguntas y respuestas de una asignatura.....	32
Figura 23. Detalles de la nota .....	33
Figura 24. Vista crear preguntas y distintos cuadros de diálogo al añadir el enunciado o soluciones .....	34

Figura 25. Vista de crear pregunta después de añadir 2 soluciones y cuadro de diálogo con error de fecha de inicio anterior a la del servidor .....	36
---	----

# 1.INTRODUCCIÓN

## 1.1. Motivación

Con el paso de los años y el avance tecnológico, hemos pasado de tener simples teléfonos que sólo servían para hacer llamadas o enviar SMS (*Short Message Service*), a tener pequeños ordenadores que se usan para muchas más cosas que únicamente hacer llamadas o enviar SMS.

El uso de *smartphones* y *tablets* se está convirtiendo en un pilar esencial en la sociedad, así como en el ámbito educativo, por lo que cada vez se está implementando más el m-learning (*mobile learning*), que es el uso de *smartphones*, *tablets*, o cualquier dispositivo con conexión inalámbrica en una metodología de aprendizaje educativa.

Teniendo en cuenta lo anterior, el grupo de Grupo de Comunicaciones Ópticas de la Universidad de Valladolid decidió crear una plataforma m-learning, con la que profesor y alumno pudieran trabajar a través de su dispositivo con conexión inalámbrica.

Para ello, dicho grupo de investigación ha creado un entorno de aprendizaje virtual, denominado AIM-Mobile Learning Platform, que constará de dos aplicaciones distintas. Por una parte, el profesor usará una aplicación web a través de la cual podrá diseñar y lanzar preguntas en tiempo real a los alumnos y así evaluarlos teniendo en cuenta sus respuestas. Además podrá colgar avisos y apuntes de sus asignaturas, así como gestionar en tiempo real la evaluación continua del alumno. Por otro lado, también existe una aplicación cliente, que será la que usará el alumno y que en una versión anterior permitía al alumno leer los avisos y los apuntes que hubiera subido el profesor, responder a las preguntas lanzadas y ver sus notas.

La aplicación del alumno está programada en Sencha y Cordova, con la idea de tener una aplicación multiplataforma en diferentes sistemas operativos móviles y en su versión web. Sin embargo, en este Trabajo Fin de Grado se desea ir un paso más allá y al ser Android uno de los sistemas operativos más usados en *smartphones* y *tablets*, se ha decidido hacer una versión nativa en Android, con la finalidad de poder comparar su

## 1. INTRODUCCIÓN

rendimiento y prestaciones con la versión análoga multiplataforma. Además, la nueva versión en Android incluye un conjunto de nuevas funcionalidades, tales como permitir al alumno crear preguntas en tiempo real, conectarse a cualquier servidor en el que esté alojada la plataforma de aprendizaje, o recibir una nueva contraseña en el correo electrónico en caso de haberla olvidado. Estas funcionalidades otorgarán a la plataforma de aprendizaje mayor potencialidad y proyección futura, pues podrá ser usada por cualquier institución educativa de una forma más segura y personalizada al poder alojarla en sus propios servidores. Además, el reseteo automático de la contraseña ofrecerá mayor seguridad al sistema y mayor nivel de independencia al alumnado.

### 1.2. Objetivos

Se podría estudiar los objetivos como un objetivo global y varios secundarios.

- **Objetivo Global:** Desarrollar una aplicación nativa en Android desde cero, que reproduzca con exactitud la aplicación creada en Sencha y Cordova.
- **Objetivos Específicos:**
  - Desarrollar tres nuevas funcionalidades en la aplicación que son:
    - Posibilidad de conectarse a cualquier servidor web donde esté alojado el entorno virtual de aprendizaje.
    - Posibilidad de que el alumno pueda recibir una nueva contraseña, en caso de olvidarse de ella o desear resetearla.
    - Posibilidad de que el alumno pueda crear sus propias preguntas y lanzarlas en tiempo real al resto de alumnos.
  - Corregir los posibles errores que tuviera la anterior plataforma.

### 1.3. Estructura de la memoria

En el Capítulo 2 se hace una introducción a la plataforma de aprendizaje desarrollada en trabajos anteriores. En este sentido, se explica la estructura global de la plataforma, se describen las herramientas involucradas en el desarrollo de la versión actual, así como las que se utilizarán en el diseño de la nueva versión móvil de la aplicación cliente en este Trabajo Fin de Máster y finalmente se definirá la metodología utilizada.

## 1. INTRODUCCIÓN

En el Capítulo 3 se describen los pasos relacionados con el diseño y programación de la nueva aplicación cliente en Android a partir de una de las primeras versiones de la aplicación cliente multiplataforma.

En el Capítulo 4 se explican las nuevas funcionalidades de la aplicación y cómo fueron llevadas a cabo.

El Capítulo 5 de esta memoria se resumen las principales conclusiones extraídas en este Trabajo Fin de Grado así como las posibles futuras modificaciones que podría tener la aplicación.

El Capítulo 6 resume la bibliografía utilizada en este trabajo.

## 1. INTRODUCCIÓN



## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

### 2.1. Introducción

En este capítulo de la memoria se explicará a grandes rasgos el entorno de aprendizaje virtual AIM-Mobile Learning Platform. Así pues, el primer paso a realizar en este Trabajo Fin de Grado fue entender cómo funcionaban las versiones preliminares del entorno de aprendizaje, tanto la aplicación servidora como la cliente, y qué herramientas se utilizaban. A partir de ahí, el siguiente paso será aprender a desarrollar aplicaciones simples en Android. Para ello se cursó un *Massive Open Online Course* impartido por la Universidad de Valencia denominado “*Android: Introducción a la programación*” [1]. Finalmente, en este capítulo se mostrarán las herramientas utilizadas en el desarrollo de la nueva aplicación cliente del alumno en Android.

### 2.2. Entorno de aprendizaje virtual AIM-Mobile Learning platform

El entorno o plataforma de aprendizaje virtual AIM-Mobile Learning Platform, desde el punto de vista de desarrollo web, se puede dividir en dos partes diferentes, el back-end y el front-end, tal y como se muestra en la Figura 1. El back-end, hace referencia a la aplicación servidora, la que utiliza el profesor, en ella se generan contenidos y puede ver o tratar con los datos enviados por el cliente. Por otro lado, el front-end hace referencia a la aplicación cliente, la que usa el alumno. En esta aplicación se visualizan los contenidos creados por el profesor, y el cliente no accede a los datos directamente, sino que se utiliza una API. Los datos enviados desde el front-end se almacenan en una base de datos del lado del servidor.

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

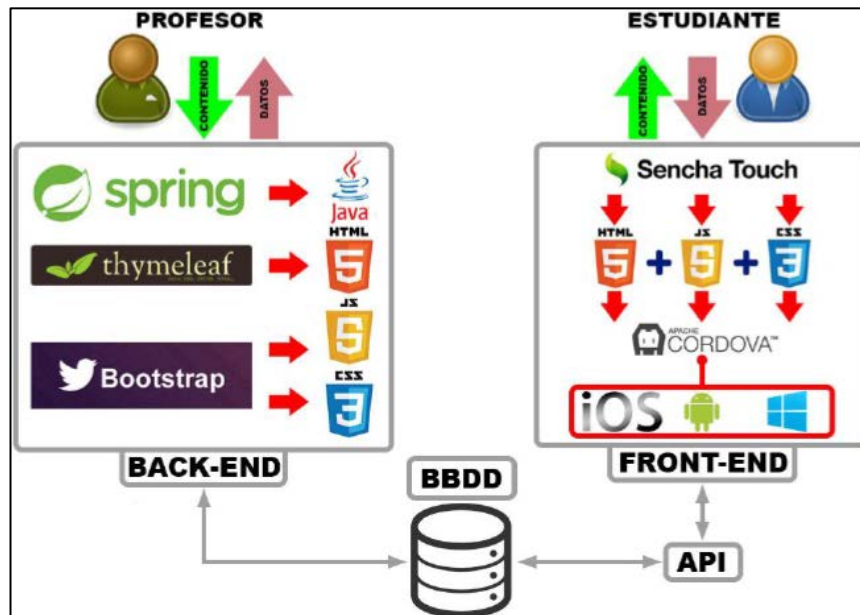


Figura 1. Diseño global del entorno virtual de aprendizaje AIM-Mobile Learning Platform desde el punto de vista del desarrollo web.

### 2.2.1 Herramientas de desarrollo de la aplicación del profesor

A la aplicación servidora, se puede acceder desde un navegador web, desde cualquier ordenador, tablet o dispositivo móvil, ya que el diseño web es adaptable al tamaño de la pantalla. Desde ella, se puede generar contenidos que se puedan visualizar desde la aplicación cliente a tiempo real, además de gestionar los datos enviados por parte de la aplicación cliente.

Más concretamente, desde la aplicación servidora, se podrá dar de alta y asociar profesores a una o varias asignaturas, en las que el profesor asociado podrá crear libros electrónicos, avisos y preguntas. Además, se podrá gestionar usuarios, asignaturas estudiantes y copias de seguridad si se tiene rol administrativo.

Para el desarrollo de la plataforma del profesor se ha empleado Spring utilizando Java como lenguaje de programación y Thymeleaf y Bootstrap utilizando HTML5, CSS3 y JavaScript para el desarrollo de la interfaz de usuario.

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

### 2.2.2 Herramientas de desarrollo de la aplicación del alumno

Al igual que en la aplicación servidora, a la aplicación cliente se puede acceder desde un navegador web, a través de cualquier ordenador, tablet o dispositivo móvil, puesto que el diseño web es adaptable. Sin embargo, puesto que la aplicación del cliente ha sido programada en un lenguaje multiplataforma, también se puede acceder a ella desde aplicaciones en IOS (iPhone OS) o Android.

En la aplicación cliente, se pueden visualizar los contenidos colgados por la aplicación servidora, tales como avisos, libros electrónicos y preguntas que se podrán responder. Para acceder a todos estos contenidos, además de estar dado de alta en la aplicación servidora, hay que estar registrado en la aplicación cliente.

Para el desarrollo de la plataforma del alumno se ha empleado Sencha Touch, y a partir de la aplicación web generada, se ha utilizado Apache Cordova para generar la aplicación nativa para los sistemas operativos móviles configurados.

### 2.3. Herramientas empleadas en el desarrollo de la nueva aplicación

Las herramientas de trabajo que se han utilizado para desarrollar el proyecto serán las siguientes:

- **Android Studio** [2]: Android Studio es el Entorno de Desarrollo Integrado (IDE) oficial de Android. Está diseñado para que Android pueda acelerar el desarrollo y te permita crear las aplicaciones (apps) de mejor calidad para todos los dispositivos de Android. Ofrece herramientas personalizadas para programadores de Android. Se incluyen herramientas completas de edición, depuración, pruebas y perfilamiento de códigos. Ésta Será la herramienta utilizada para programar todo el proyecto. En la Figura 2 se observa la interfaz del programa.

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

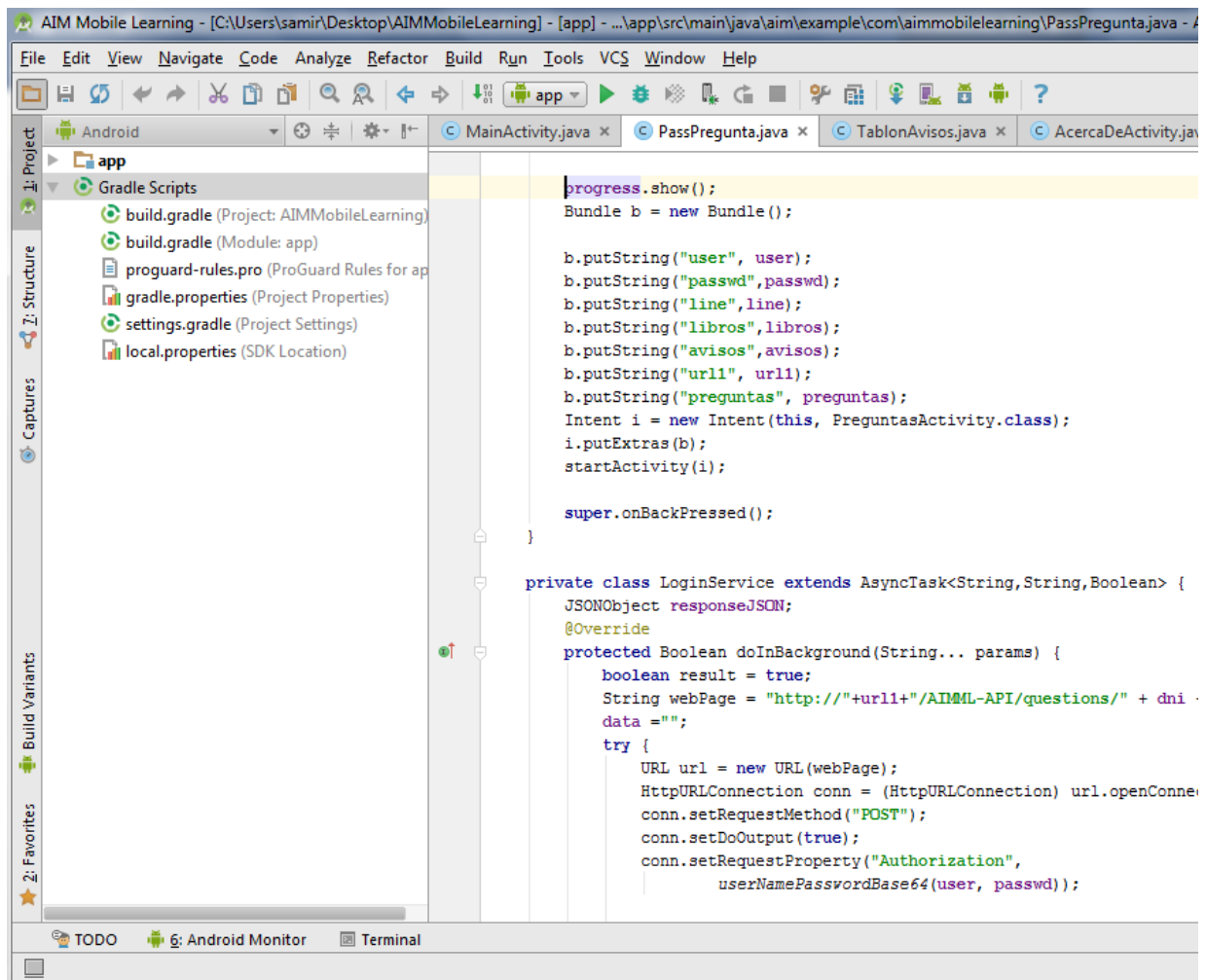


Figura 2. Entorno de desarrollo de Android Studio

- **Notepad** [3]: es un programa para editar código fuente de cualquier lenguaje de programación. Como tiene soporte para una gran cantidad de lenguajes, interesa no sólo a los desarrolladores de webs, sino en general a toda la comunidad de programadores.

Utilizada para leer el código recibido de la anterior aplicación necesario para comprender su funcionamiento, y para leer el código de la API para poder hacer todo tipo de peticiones (Figura 3).

- **Advanced Rest Client** [4]: Es una extensión del navegador Google Chrome que nos permite lanzar peticiones a servicios o API Restful.

Esta herramienta es utilizada para ver las respuestas de las distintas peticiones a la API antes de probar estas peticiones en la aplicación cliente (Figura 4).

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

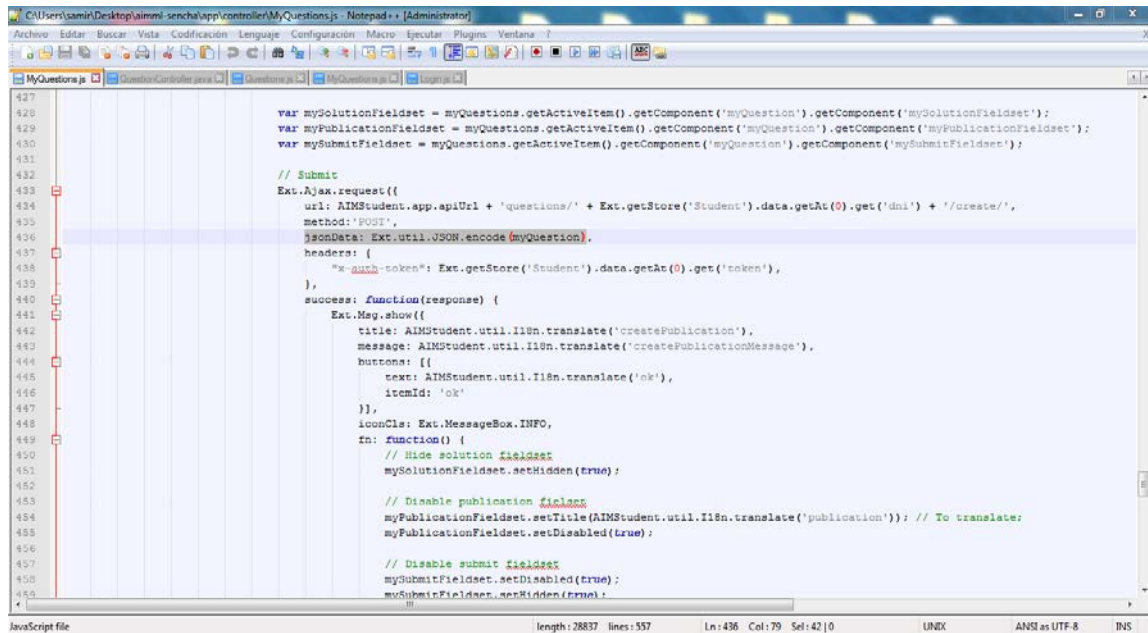


Figura 3. Captura del entorno del editor Notepad

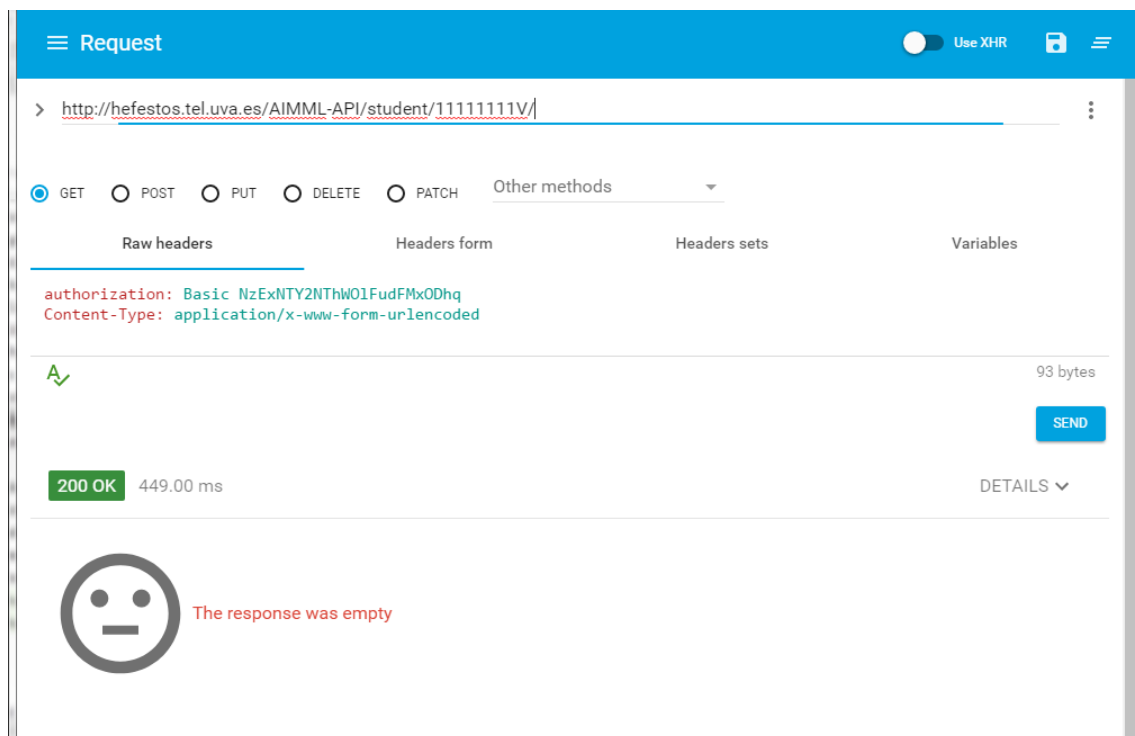


Figura 4. Entorno de la extensión Advanced Rest Client para Google Chrome

Por último, mencionar los dispositivos utilizados para las múltiples pruebas de funcionamiento de la aplicación, que fueron los smartphones BQ Aquaris E4.5 [5] y HUAWEI y6ii compact [6].

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

### 2.4. Metodología

El método de trabajo llevado a cabo ha sido decidir objetivos semanales, que la tutora iba corrigiendo para la creación del proyecto. Entre estos objetivos podemos distinguir los siguientes pasos seguidos a través de la creación de la aplicación:

- Después de haber hecho el tutorial de Android y decidida la herramienta de trabajo, lo primero fue diseñar e implementar las vistas de la aplicación, sin ninguna funcionalidad programada, pero con la finalidad de que fueran muy parecidas a las de la versión multiplataforma existente.
- Una vez hechas las vistas, lo siguiente fue establecer conexión con el servidor a través del método GET, que lleva los datos de forma “visible” al cliente mediante la URL [7].
- Una vez establecida la conexión con el método GET, ya pudimos comprobar si el nombre y contraseña de un usuario alumno eran correctos, ya que se tenía conexión con el servidor web donde está alojada la plataforma de aprendizaje y se podría autenticar con los datos correspondientes. Además, ya se podían de forma correcta descargar los avisos, libros y preguntas creados por el profesor desde su aplicación y las notas que tuviera el alumno al responder dichas preguntas.
- Después de dar forma a todo lo recibido por el servidor, se pasó a implementar el establecimiento de la conexión con el servidor utilizando el método POST, que lleva los datos de forma “oculta” al cliente, lo cual lo hace más seguro. Este método de conexión es necesario para enviar las respuestas del alumno y registrar a un alumno nuevo.
- Una vez implementadas todas las vistas y funcionalidades de la aplicación multiplataforma del alumno, se pasó a diseñar y desarrollar un conjunto de nuevas funcionalidades. Se comenzó por la funcionalidad para poder elegir el servidor al que conectarse, es decir, donde se encuentra alojado el entorno virtual de aprendizaje. A continuación, se programó una funcionalidad para que el alumno desde su aplicación pudiera crear preguntas tipo test y lanzarlas al resto de alumnos, análoga a la ya existente desarrollada para el profesor. Finalmente, se implementó una nueva funcionalidad en la que el alumno pudiera recibir una nueva contraseña en su correo electrónico, en caso de haber olvidado su contraseña anterior.

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO

- Por último, se pasó a la fase de prueba y depuración de fallos y se dio por finalizada la primera versión en Android de la aplicación del alumno.

### 2.5. Conclusiones

En este Capítulo de la memoria se han descrito las herramientas de trabajo así como la metodología utilizada a lo largo de este Trabajo Fin de Grado. Tal y como se ha explicado, se comenzó por aprender cómo funcionaba la plataforma de aprendizaje analizando el diseño de la misma, en concreto, las funcionalidades de las aplicaciones servidor y cliente. El siguiente paso fue elegir las herramientas que se iban a utilizar en el diseño de la nueva aplicación del alumno en Android y los pasos se llevaron a cabo para realizarlo.

## 2. METODOLOGÍA Y HERRAMIENTAS DE TRABAJO



# ***3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE***

## **3.1. Introducción**

En este Capítulo de la memoria se va a explicar cómo se diseñó y se implementó la aplicación cliente en Android a partir de una de las primeras versiones multiplataforma. Así mismo, se describirán los pasos llevados a cabo para su programación.

## **3.2. Desarrollo de las vistas de la aplicación cliente**

Cómo ya se ha indicado en la metodología expuesta en el Capítulo 2, lo primero que se propuso fue el diseño y programación en Android de las vistas, una vez aprendidas las bases de la programación en este lenguaje.

En Android Studio cada actividad creada en un archivo java, va acompañada de una vista o *layout* guardado en un archivo XML (*eXtensible Markup Language*). Así pues, para crear todas las vistas, lo que se hizo fue crear una actividad por cada vista de la aplicación acompañada de su *layout* correspondiente. Estas actividades sólo contenían el código necesario para cargar el *layout* y poder comprobar cómo se veían en la pantalla de un smartphone.

Una vez creadas todas las actividades, empezamos a crear cada *layout* correspondiente a cada una de las funcionalidades de navegación de la aplicación cliente. Una de las

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

ventajas de Android Studio es que además de programar las vistas escribiendo código (*Text*), tiene una pestaña (*Design*) que te permite programar los distintos elementos sin necesidad de saber programación en xml. Además, permite ver cómo se visualizan todos los elementos que se insertan y diseñan en la vista, como si fuera la pantalla del propio dispositivo, pudiendo elegir entre diferentes dispositivos móviles. Esta funcionalidad es muy útil, pues permite ver cómo quedaría la vista en cada uno de ellos.

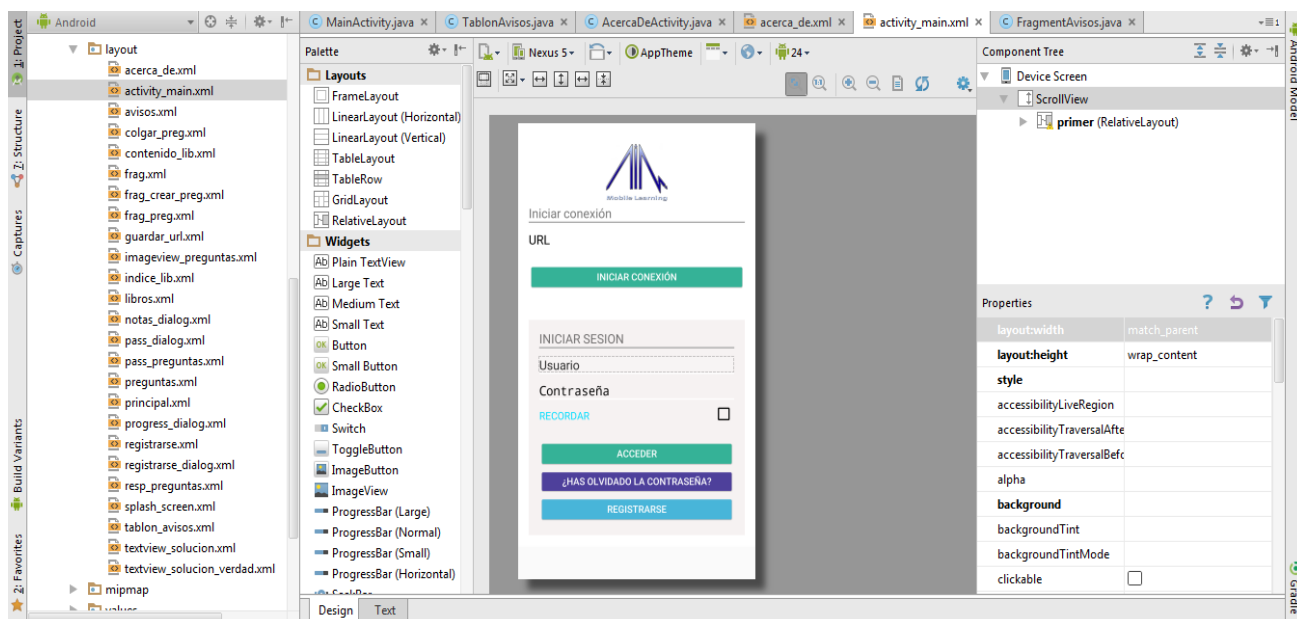


Figura 5. Ejemplo de vista creada en Android Studio

En la Figura 5 se puede ver cómo quedaría la vista de entrada de la aplicación en un dispositivo Nexus 5 (en la barra de arriba se ve el tipo de dispositivo), también podemos ver todas las vistas que se han creado para esta aplicación.

Para poder navegar entre las distintas vistas de la aplicación, se añadió la funcionalidad correspondiente a los botones para cargar otras actividades y vistas.

#### 3.3. Peticiones GET a la API del entorno virtual

El primer paso para empezar a implementar las funcionalidades de la aplicación era conseguir hacer peticiones a la API (*Application Programming Interface*)[8], ya que tanto los libros electrónicos, como los avisos y las preguntas, vienen en respuestas a las peticiones. En primer lugar, se comenzó usando la extensión *Advanced Rest Client* de *Google Chrome* para ver en que formato venían las respuestas. El formato era JSON

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

(*JavaScript Object Notation*) [9], que es un formato para el intercambio de datos. Básicamente JSON describe los datos con una sintaxis dedicada que se usa para identificar y gestionar los datos.

Después de ciertas tareas de investigación, se pasaron a realizar las peticiones a la API en el foro *stack overflow* [10], donde se resolvieron muchas dudas. En segundo plano, usando las urls (*Uniform Resource Locator*) necesarias y mandando una autorización básica junto con el dni y la contraseña, se recibe cada JSON correspondiente a cada petición. A partir de aquí con las peticiones GET, se reciben las asignaturas en las que el alumno está matriculado. Además, con otras peticiones también se pueden recibir los libros electrónicos, los avisos y las preguntas colgadas por el profesor.

Con el uso de las peticiones GET básicas, se pueden realizar todas las funcionalidades excepto registrar un usuario, recibir las preguntas que necesitan contraseña y contestar a las preguntas.

#### **3.4. Desarrollo de funcionalidades en la aplicación cliente**

Una vez conseguida la información necesaria en forma de JSON, se pueden probar la mayoría de las funcionalidades existentes en la versión multiplataforma de la aplicación cliente. Es conveniente avisar de que las capturas están tomadas la versión de la aplicación, por lo que hay cosas que no están explicadas en este capítulo que aparecerán en ellas.

Comenzamos por la página de bienvenida, donde comprobamos en el servidor donde está alojado el entorno virtual ([hefestos.tel.uva.es](http://hefestos.tel.uva.es)) que el alumno está registrado, tal y como se muestra en la Figura 6. Para ello se introduce su contraseña correspondiente, y en caso de estar registrado, el alumno pasaría a ver el menú principal junto con sus funcionalidades principales. En el caso de que el alumno no esté registrado o introduzca mal la contraseña saldrá un mensaje de error correspondiente. Además, la aplicación del alumno permite guardar el nombre de usuario y contraseña para que al volver a entrar en ella no haya que escribirlo de nuevo.

Una vez que se validan correctamente los datos del alumno en la aplicación, se accede y se visualiza en el menú principal (Figura 7), desde aquí podremos acceder a una pantalla de acerca de, a los avisos, a los libros electrónicos y a las preguntas.

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

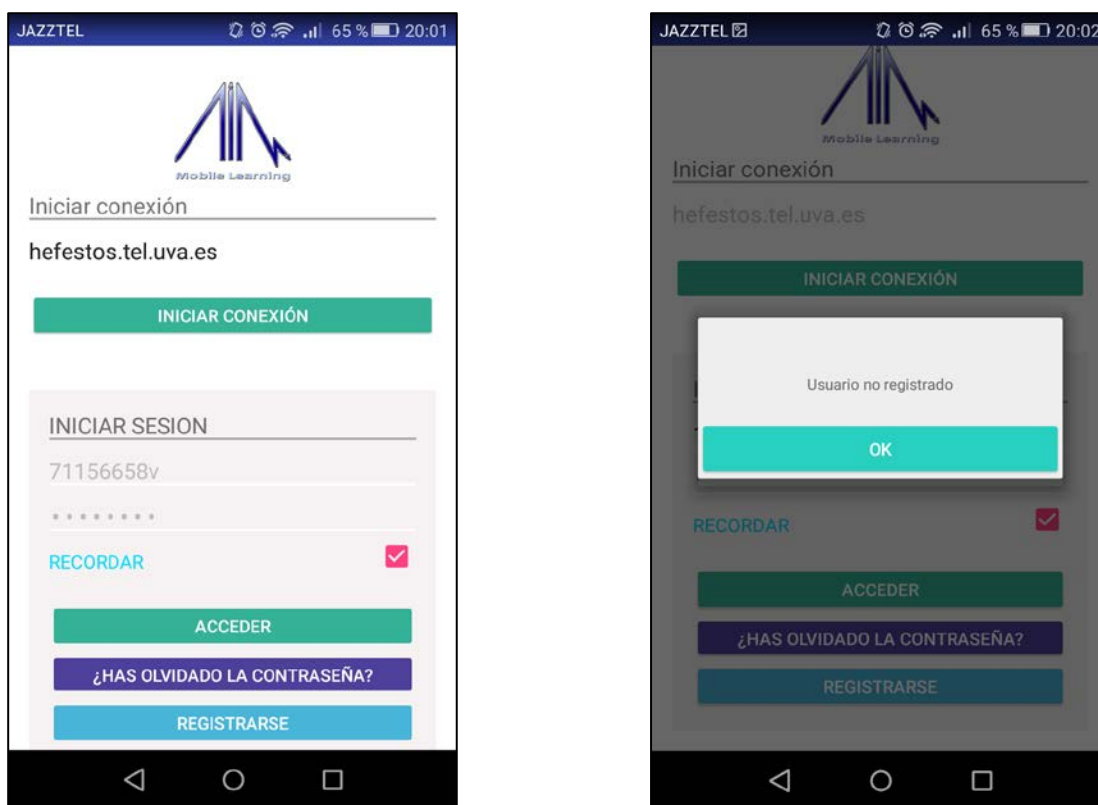


Figura 6. Pantalla de inicio y conexión a la aplicación del alumno

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE



**Figura 7. Visualización del Menú Principal y sus funcionalidades**

Antes de implementar cada funcionalidad, se generó un sistema de pestañas mediante fragmentos, ya que, al hacer cualquier petición para un alumno concreto, se recibe un JSON con todos los datos correspondientes a esa petición independientemente de la asignatura. Por ejemplo, si un alumno apuntado a las asignaturas cuyo código identificativo son *SA0* y *SA1*, entra en la sección de avisos, se hace una petición para la cual se recibe un JSON con los avisos de las dos asignaturas, teniendo que mirar en cada aviso el identificador de la asignatura a la que pertenece.

Debido a esto, se generó una pestaña por asignatura a la que el alumno está apuntado y se compara el identificador de cada objeto del JSON recibido en cada petición con el de la asignatura de esta pestaña. Con esto conseguimos que en cada pestaña se muestren los datos de cada asignatura, evitando que se junten los datos de todas.

A continuación se describen las funcionalidades accesibles desde el menú principal. En concreto:

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

- **Funcionalidad “Acerca de”:**

La pestaña de acerca de, no es más que un párrafo en el que se pueden ver las personas y entidades implicadas en que el desarrollo de la plataforma de aprendizaje.

- **Funcionalidad “Avisos”:**

En la sección de Avisos, se hace una petición GET que nos envía un JSON con todos los avisos de todas las asignaturas a las que esté un alumno apuntado. Usando el sistema de pestañas, se mostrarán en cada asignatura los avisos correspondientes, tal y como se observa en la Figura 8. Los avisos están separados dependiendo del mes y del año en el que han sido creados y se visualizarán según el orden de creación. Al entrar dentro de cada aviso, se visualiza una nueva vista en la que se muestran todos los datos del aviso creado por el profesor, es decir, fecha, nombre y descripción del mismo (Figura 8 derecha).

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

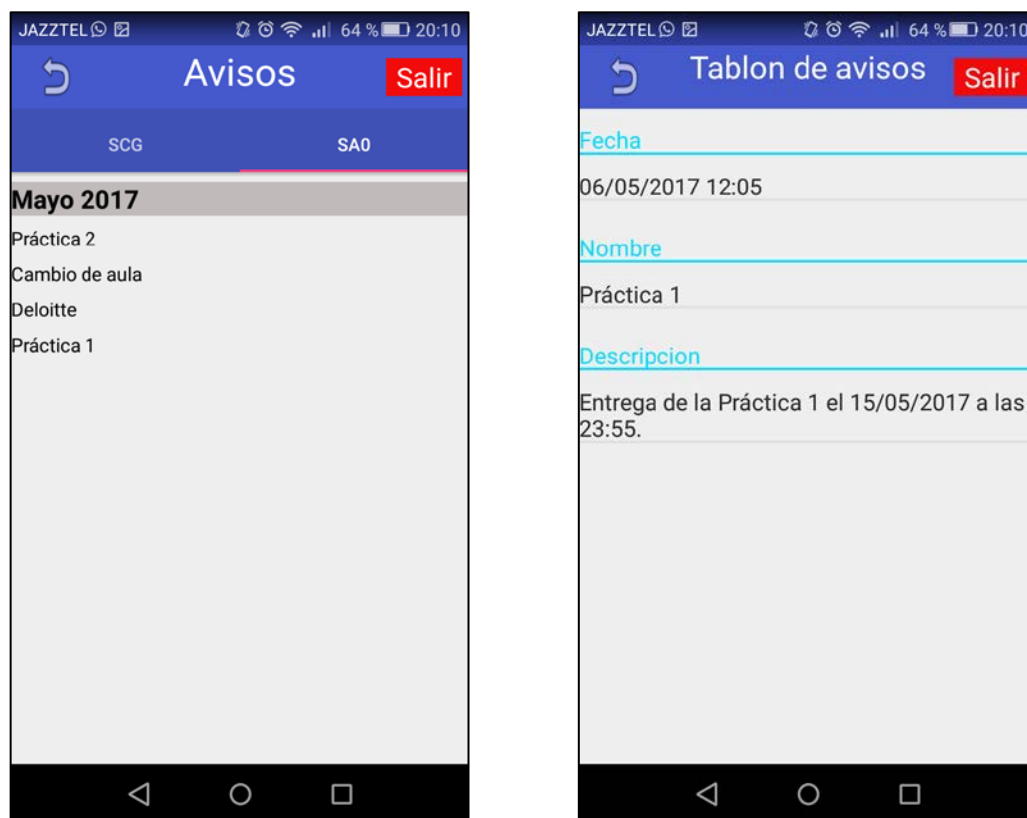


Figura 8. Visualización de todos los avisos de una asignatura y los datos completos de un aviso concreto

- **Funcionalidad “Libros”:**

La funcionalidad de libros es parecida a la de avisos, ya que con una petición GET se reciben todos los libros electrónicos de todas las asignaturas en las que está matriculado el alumno. Se muestran los libros separados también por el mes y el año en que fueron creados, y por orden de creación, tal y como se observa en la Figura 9. Al entrar dentro de cada libro aparece una nueva vista con el índice completo del libro. Al acceder a una página de un libro, se hace una última petición GET con la que se recibe un código HTML (*HyperText Markup Language*) usado en la programación de páginas web. Mediante dicho código, se visualizará en el dispositivo la página entera, con los vídeos, imágenes y texto que haya cargado el profesor, como muestra el conjunto de capturas que se observan en la Figura 10.

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE



Figura 9. Vista de los libros de una asignatura y del índice de un libro



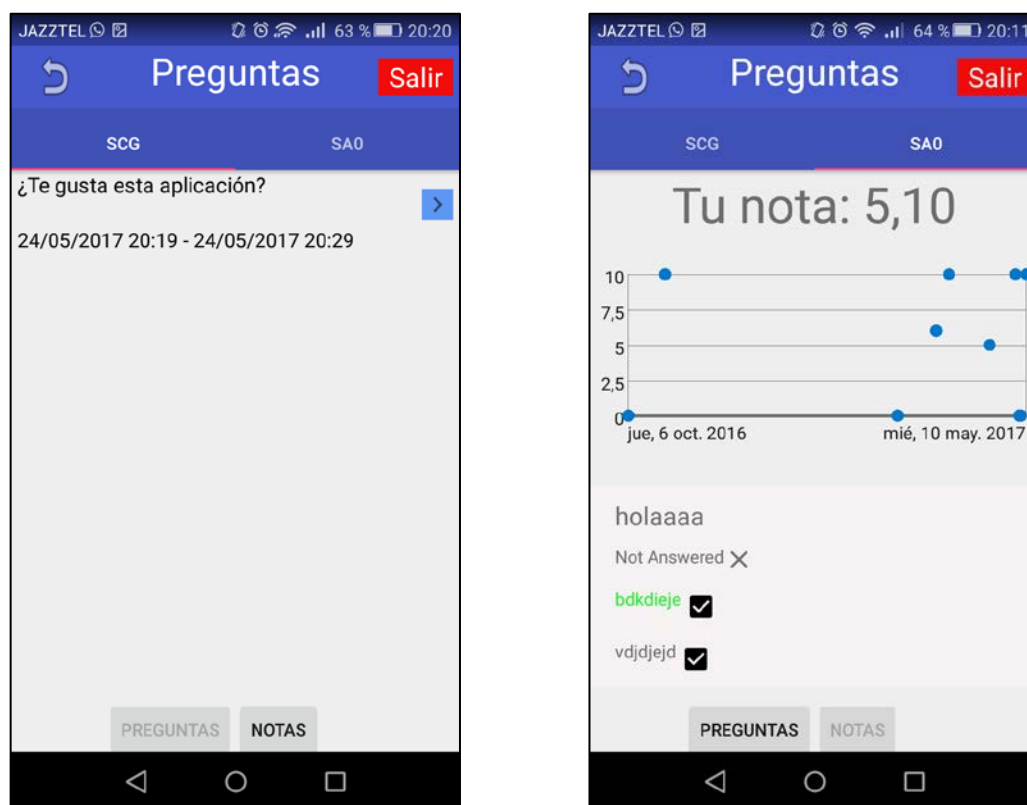
Figura 10. Vista de diferentes páginas dentro de un libro electrónico



### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

- **Funcionalidad “Preguntas”:**

En la sección de Preguntas utilizando sólo peticiones GET, se pueden visualizar las preguntas que haya activado o lanzado el profesor (Figura 11). En el caso de no haber ninguna pregunta activa, se visualizará un texto en la pantalla que dice “no hay preguntas”, pero hasta que no se usen las peticiones POST no será posible responderlas. Por otra parte, se ha implementado otro pequeño sistema con pestañas inferiores, para poder ver las notas que ha sacado el alumno en las diferentes asignaturas. Tal y como se observa en la Figura 11, dichas notas se visualizan a través de una gráfica, en la que muestran las notas de cada día junto con la nota media acumulada que lleva el alumno en esa asignatura.



**Figura 11. Visualización de las preguntas de una asignatura y gráfica con las notas conseguidas y nota media**

Antes de comenzar con las peticiones POST, se tendrían implementadas todas las funcionalidades de la versión multiplataforma excepto responder a las preguntas colgadas por el profesor, visualizar las respuestas que precisan de contraseña y registrar a un alumno nuevo.

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

#### 3.5. Peticiones mediante el método POST y desarrollo de funcionalidades

Para realizar las peticiones POST, se necesita enviar junto con la autorización, los datos que recibe la API de la aplicación cliente, y a continuación se recibe una respuesta de código 200 (OK), para saber que está bien enviada la petición.

Una vez conseguido hacer estas peticiones, pasamos a diseñar y programar las funcionalidades restantes de las primeras versiones de la aplicación del usuario, para las que se debía usar este método de peticiones. En concreto se desarrollaron las siguientes funcionalidades:

- **Funcionalidad “Registro”:**

Mediante la funcionalidad de registrarse, un usuario alumno que no estuviera dado de alta en el entorno pero sí que esté en la base de datos (el profesor lo ha subido a la plataforma), puede elegir su contraseña y darse de alta. Una vez registrado, puede entrar en el menú principal de la aplicación y visualizar los datos que haya subido el profesor a las asignaturas en las que esté matriculado, tal y como se observa en la Figura 12.

Por otro lado, en dicha funcionalidad de registro, si el alumno ha sido cargado en la base de datos por el profesor, o deja un campo de los tres del registro vacío, o las contraseñas no coinciden (repetición de contraseña), o la contraseña tiene menos de ocho caracteres, aparece un cuadro de diálogo con el correspondiente error para informar al alumno. Algunos de estos errores e informes al usuario alumno se pueden observar en el conjunto de capturas que se muestran en la Figura 12.

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

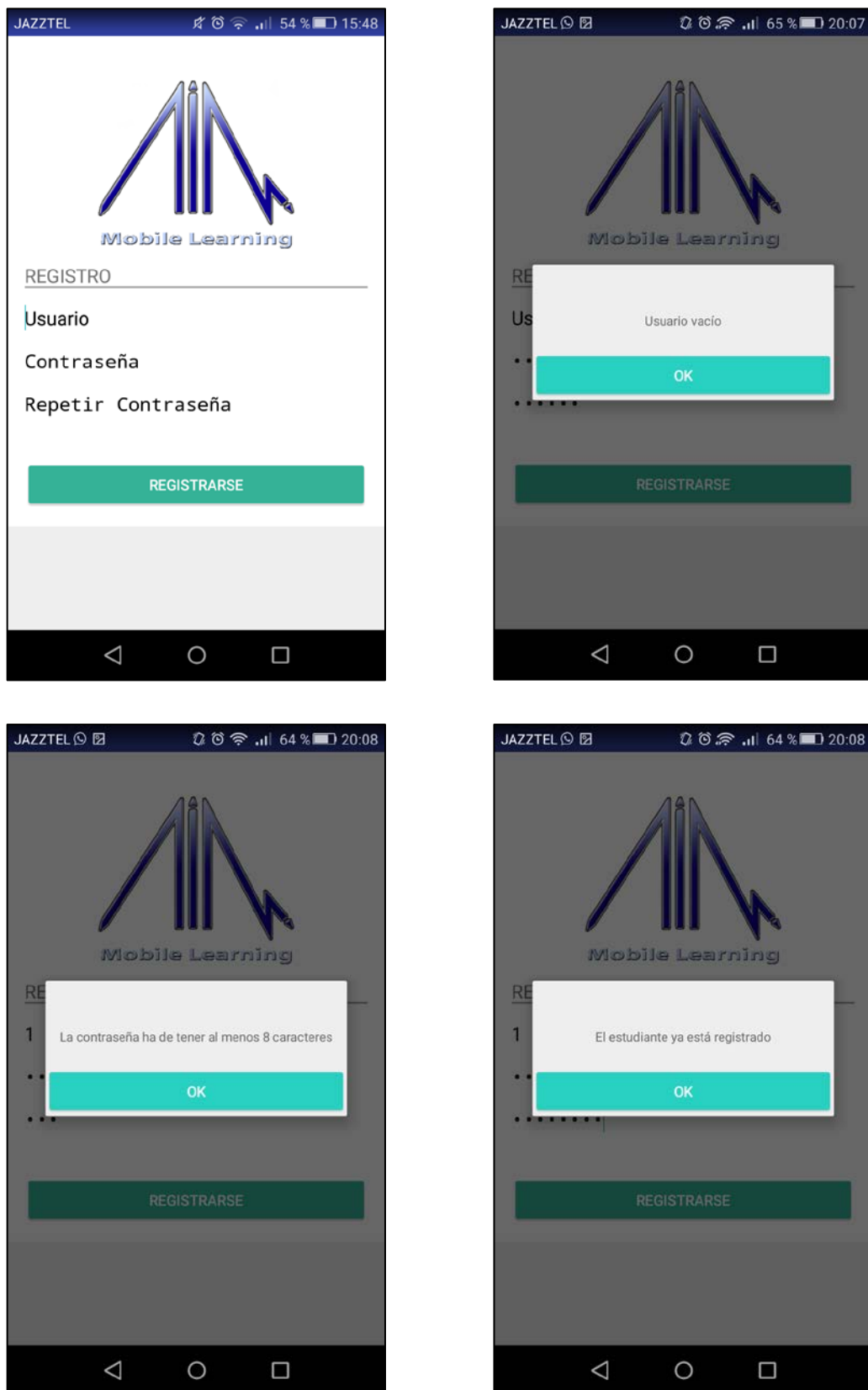
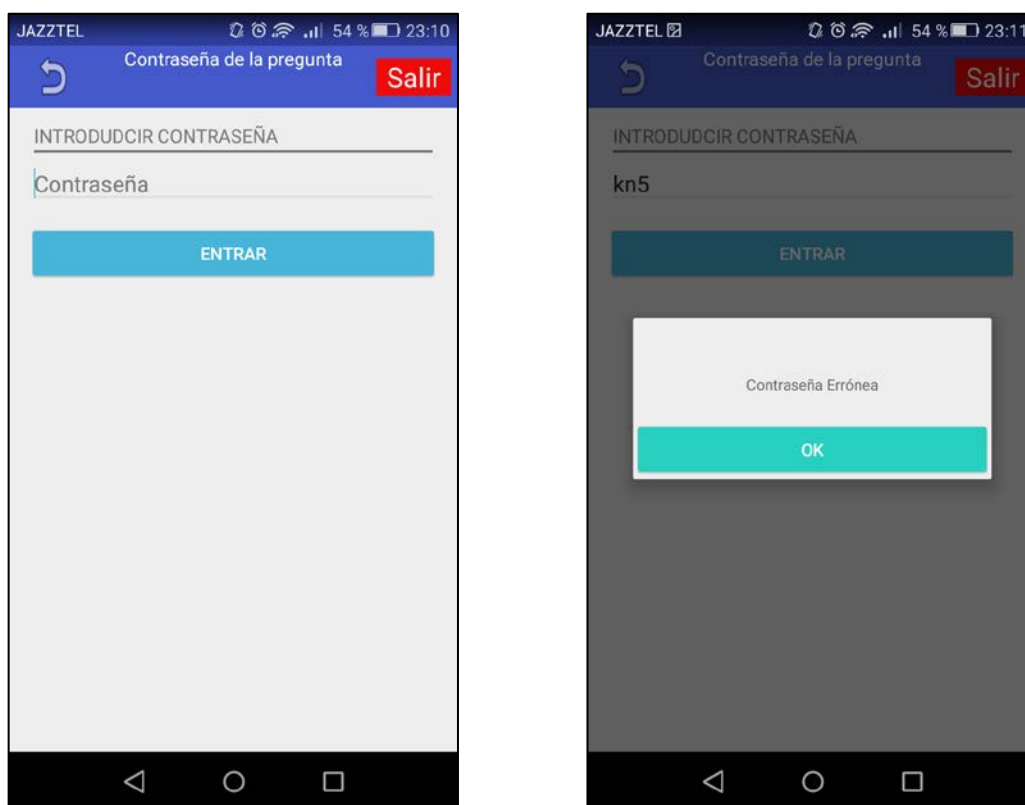


Figura 12. Vista de la aplicación del usuario alumno para que realice el proceso de registro

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

- **Funcionalidad “Responder preguntas”:**

Al entrar en la sección de preguntas, se visualizarán todas las preguntas que ha enviado el profesor y que estén activas en ese momento. Si la pregunta no requiere contraseña, introducida y distribuida por el profesor, se muestran directamente todas las respuestas. Si por el contrario se requiere contraseña, sale una nueva vista para introducirla (Figura 13 izquierda). Si la contraseña es incorrecta o está vacía aparece un diálogo con un mensaje de error, tal y como se observa en la (Figura 13 derecha). Y finalmente, en el caso de ser correcta se muestran todas las respuestas al igual que en el caso en el que no había contraseñas.



**Figura 13. Vista de las preguntas con contraseña y un error al dar una contraseña errónea**

Una vez dentro de la vista con todas las respuestas (Figura 14), la aplicación permite seleccionar sólo una respuesta de entre las posibles. Además, aunque sólo se pueda elegir una solución, puede haber una o más respuestas correctas. En la vista también aparece un temporizador que nos avisa del tiempo que va a estar la pregunta disponible para responderla. En caso de acabarse el tiempo y no haber seleccionado una respuesta, saldrá un diálogo indicando que se ha acabado

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

el tiempo y esa pregunta guardará una respuesta errónea (*Not Answered*). En caso de seleccionar una respuesta y pulsar en enviar, se guardará la respuesta elegida por el usuario. Al final del día se hace una media con las notas de las preguntas a las que se han contestado, guardándose en ese día esa nota media de todas las respuestas enviadas.



**Figura 14. Vista con todas las soluciones correspondientes a una pregunta activa**

En la pestaña de notas se podrá ver en la gráfica la nota media correspondiente a todas las respuestas enviadas de cada uno de los días en los que el alumno contestó a preguntas.

Con esta última funcionalidad estaría acabada la primera parte del proyecto, que se centraba en el desarrollo una aplicación de Android nativa, partiendo de la correspondiente versión multiplataforma. En el siguiente capítulo describiremos el diseño y programación de las nuevas funcionalidades que se han desarrollado en la aplicación en Android.

### 3. DESARROLLO EN ANDROID DE LA APLICACIÓN CLIENTE

#### **3.6. Conclusiones**

En este capítulo se ha descrito cómo se ha llevado a cabo el diseño y programación de la aplicación cliente a partir de la antigua versión multiplataforma. Esta parte del Trabajo Fin de Grado ha sido la más costosa, sobre todo el diseño de funcionalidades en un sistema de pestañas para diferenciarlas según las asignaturas en las que el alumno esté matriculado o las peticiones de datos a la API. A partir de este punto la implementación y desarrollo de nuevas funcionalidades se presupone que será menos costosa pues se dispone de una base que hará mucho más sencilla la integración de nuevas funcionalidades.

# ***4. DESARROLLO DE NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE***

## **4.1. Introducción**

En este capítulo de la memoria se describirán las nuevas funcionalidades que se han integrado en la aplicación del cliente, implementadas de forma paralela en la versión multiplataforma. Otro de los alumnos que ha estado trabajando en la aplicación cliente multiplataforma, además de encargarse de programar las mismas funcionalidades que las desarrolladas en la aplicación Android, también se encargó de actualizar el código de la API para que éstas funcionaran correctamente.

## **4.2. Funcionalidad para seleccionar el servidor de alojamiento del entorno de aprendizaje**

Se desea que en un futuro el entorno de aprendizaje se pueda utilizar en otras Instituciones educativas, las cuales tendrán su propio servidor. Debido a esto, en la página de inicio de la aplicación del alumno, se deshabilita inicialmente el acceso a la aplicación o el registro, obligando al alumno en primer lugar a introducir la dirección de un servidor, tal y como se muestra en la Figura 15. Si la dirección está vacía o no se puede establecer conexión a dicho servidor, salta un cuadro de diálogo con un mensaje de error. En caso de no poder establecer conexión con ningún servidor, no se podrá usar

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE

ninguna funcionalidad de la aplicación, es decir, no se accederá al menú principal (Figura 16). Además, al igual que en el caso del usuario y la contraseña, al establecer conexión sale un cuadro de diálogo que permite guardar la dirección del servidor para no tener que volver a escribirla al iniciar de nuevo una sesión.



Figura 15. Funcionalidad para elegir el servidor

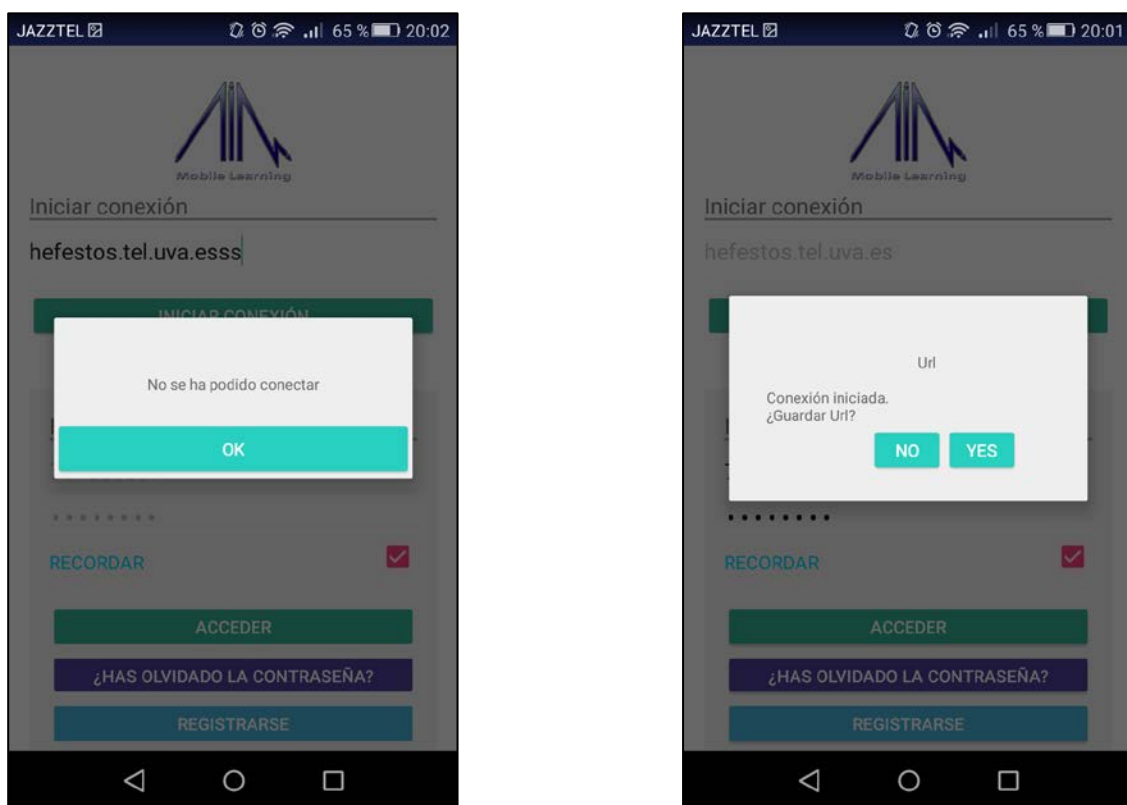


Figura 16. Cuadro de diálogo de error en la conexión con el servidor y cuadro de diálogo que permite guardar la dirección del servidor para próximas sesiones



## 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE

### 4.3. Funcionalidad para restablecer contraseña

En las primeras versiones de la aplicación, si se te olvidaba la contraseña o se perdía, la única forma de poder acceder a la aplicación era contactar con el profesor, y que éste reiniciara la cuenta del usuario desde su aplicación. Después de este proceso de reseteo de cuenta, el alumno debía volver a registrarse en el sistema.

Con el objetivo de no tener que estar pendiente de que el profesor reinicie esta cuenta y sólo depender de uno mismo, se ha creado esta nueva funcionalidad, que lo que hace es enviar una nueva contraseña al correo electrónico que está asociado al alumno en la base de datos, tal y como se muestra en la Figura 17 (derecha e izquierda).

Al pulsar en el botón de “¿Has olvidado la contraseña?”, te sale un cuadro de diálogo en el que se pide el nombre de usuario, en este caso el DNI, y en caso de ser un usuario previamente registrado en el sistema, se envía un correo electrónico con la nueva contraseña con la que podrá acceder a la aplicación (Figura 18 derecha). En caso de estar vacío el campo de usuario, o ser un usuario no registrado saldrá un mensaje de error y no cambiará la contraseña de este usuario (Figura 18 izquierda).

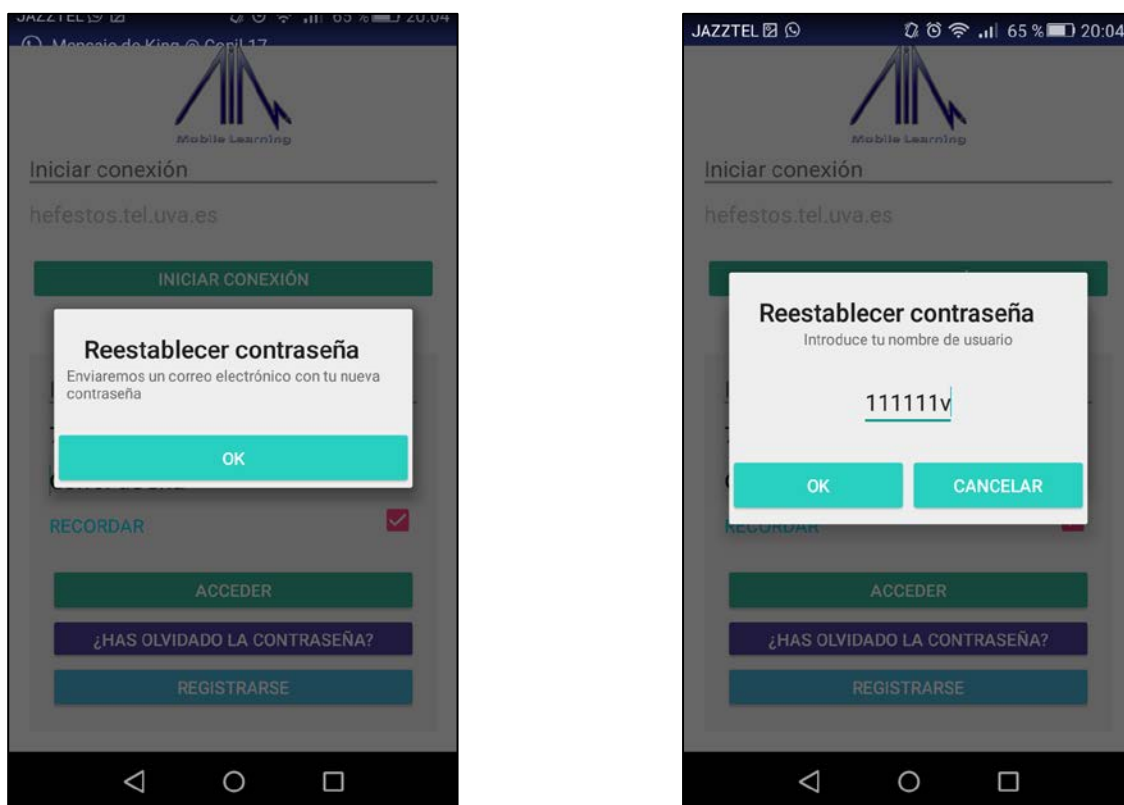


Figura 17. Vistas de la funcionalidad para restablecer la contraseña

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE

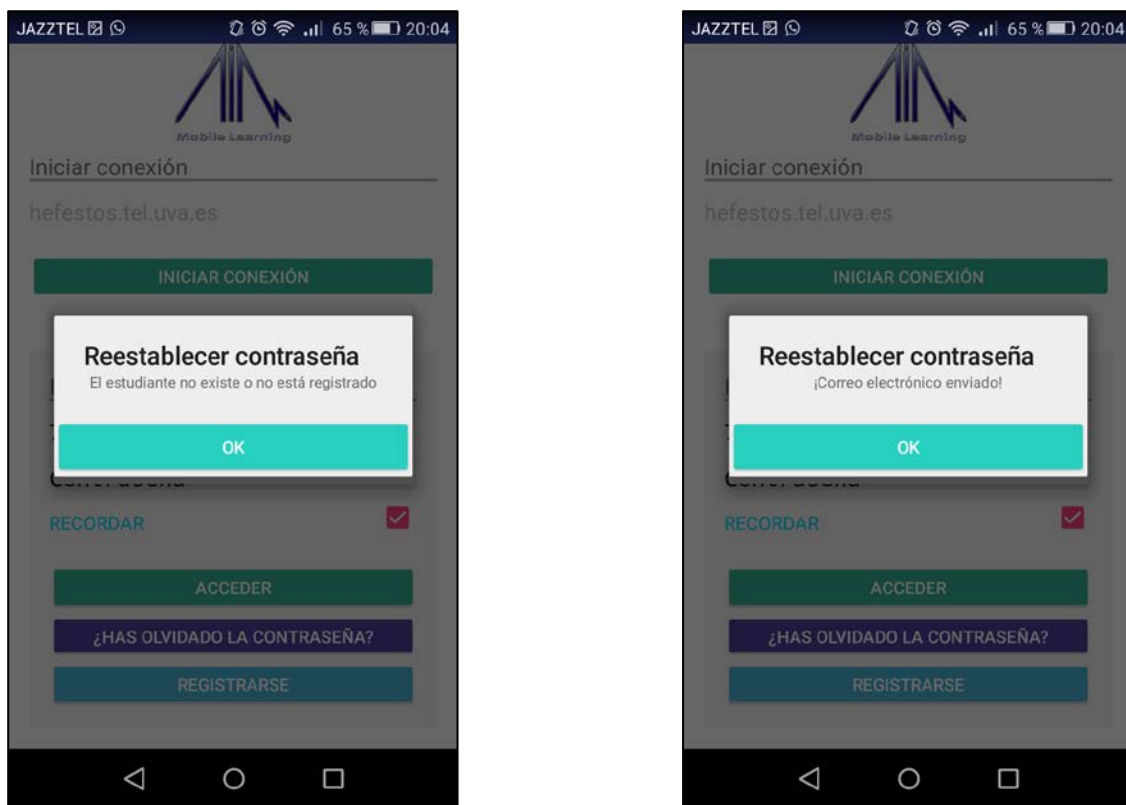


Figura 18. Cuadro de diálogo con el fallo que aparece cuando el estudiante no está registrado y cuadro de diálogo con el correo enviado con la nueva contraseña

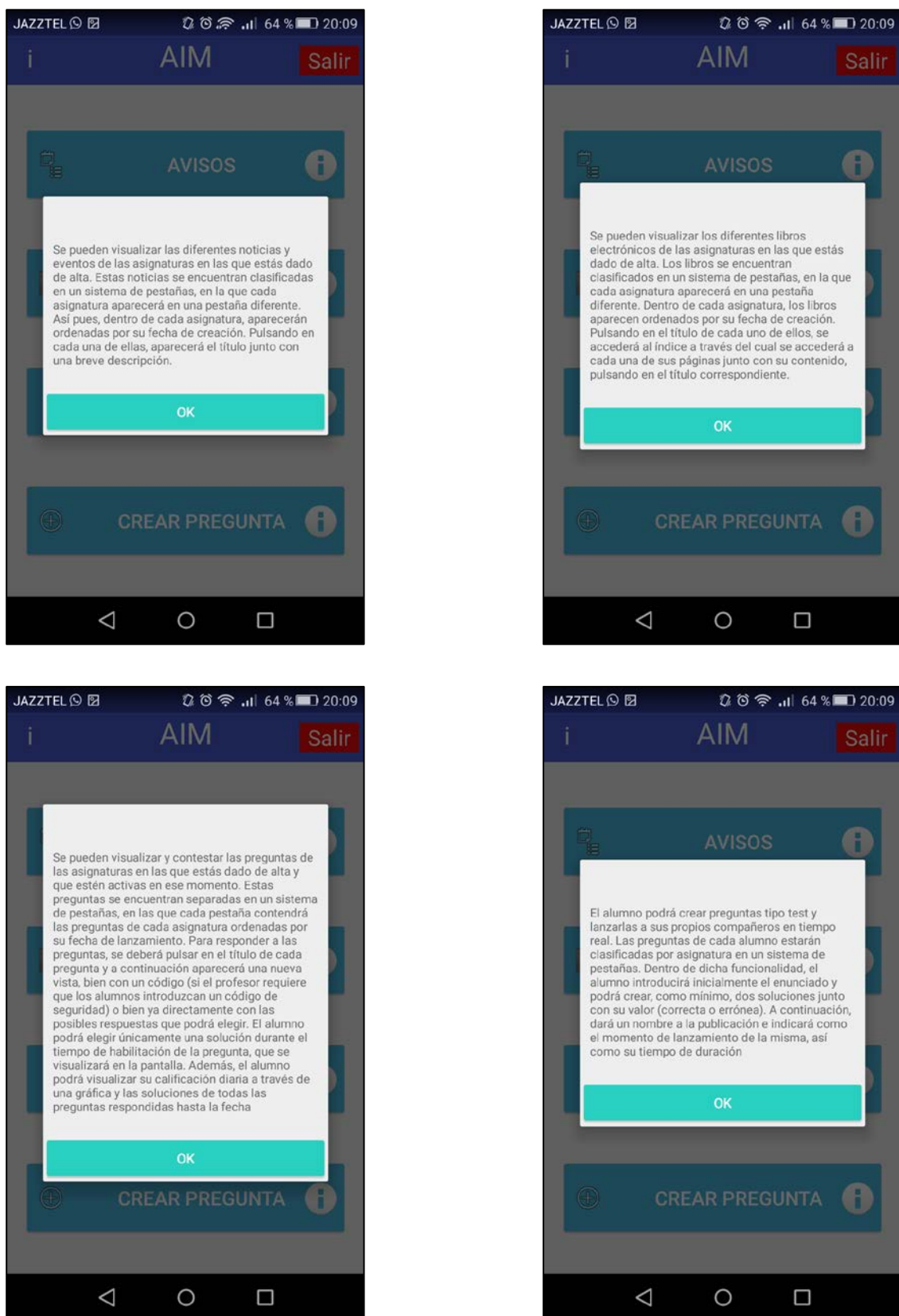
#### 4.4. Manual de ayuda de la aplicación al alumno

Dentro del menú principal de la aplicación se han añadido unos botones de información encima de cada botón de acceso a las distintas funcionalidades (Figura 19), avisos, libros, preguntas y crear preguntas. Tras pulsarlo, se abre un cuadro de diálogo ayuda de lo que se puede hacer en esa funcionalidad y cómo utilizarla (Figura 20).



Figura 19. Botones de ayuda en la aplicación cliente

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE



**Figura 20. Cuadro de diálogo del manual de ayuda de cada una de las funcionalidades de la aplicación del estudiante**

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE

##### 4.5. Visualización de las notas y de las preguntas respondidas

Al pulsar en la pestaña inferior de notas, se han incluido debajo de la gráfica todas las preguntas de esa asignatura respondidas y no respondidas (“*not answered*”). Dentro de dichas preguntas se visualizará la respuesta elegida con el color de la fuente en verde si se eligió la respuesta correcta, o en rojo si se eligió una respuesta incorrecta o no se contestó. al lado de cada respuesta sale un dibujo de una cruz si es una respuesta incorrecta y un dibujo de un *check* si es la respuesta correcta, como muestra la Figura 21.

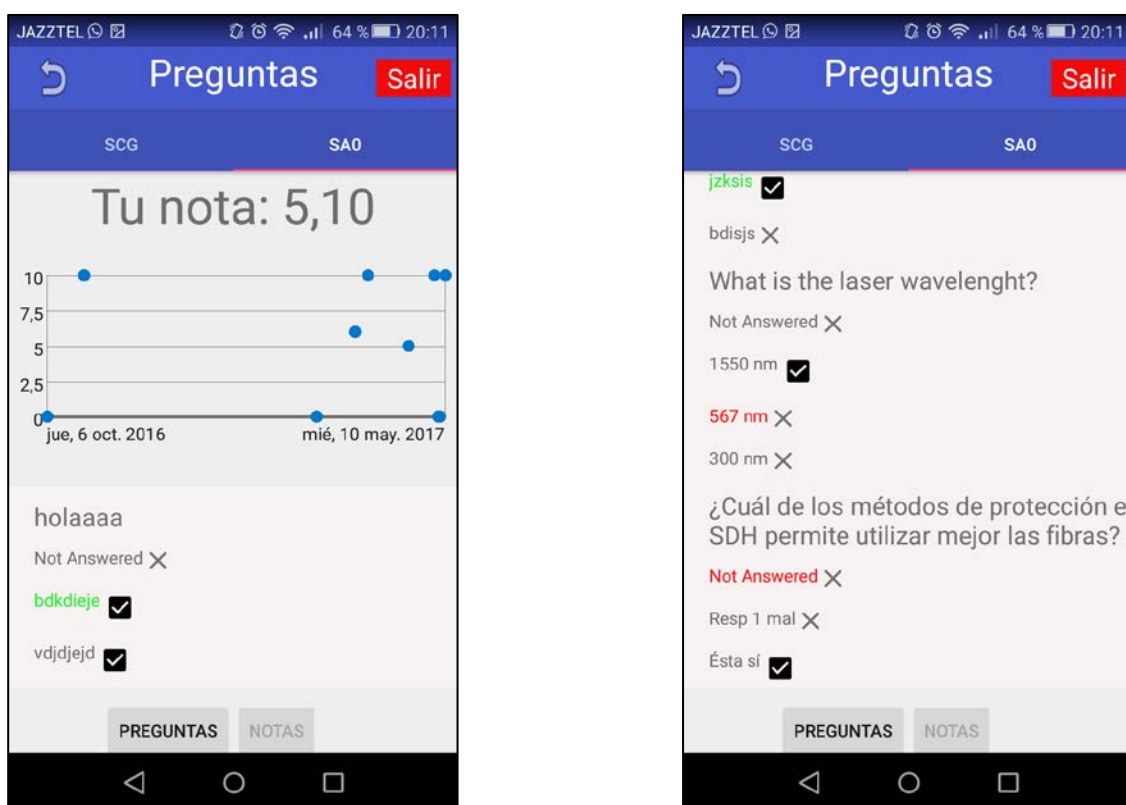
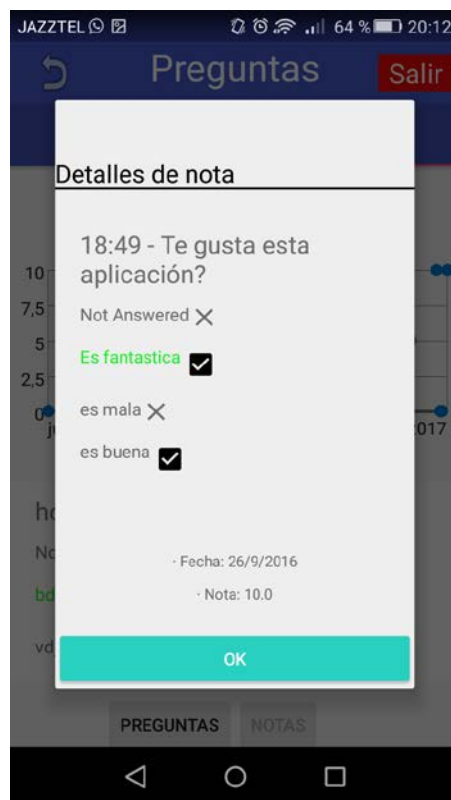


Figura 21. Preguntas respondidas por el alumno a lo largo de una asignatura

Por otra parte, al pulsar en cualquier punto de la gráfica, se abre un nuevo cuadro de diálogo llamado “detalles de la nota”, donde se puede ver las preguntas respondidas y no respondidas (“*not answered*”) en ese día, con la fecha, la hora y la nota que se sacó en cada pregunta (Figura 22). Además, en la parte superior del diálogo aparece la nota media de todas las preguntas de ese día y las respuestas en el mismo formato que en las preguntas que se visualizan debajo de la gráfica. Es decir, aparece la respuesta elegida en verde si es verdadera y en rojo si es falsa o no ha sido respondida, junto con una cruz al lado de cada respuesta falsa y un *check* al lado de cada respuesta verdadera.

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE



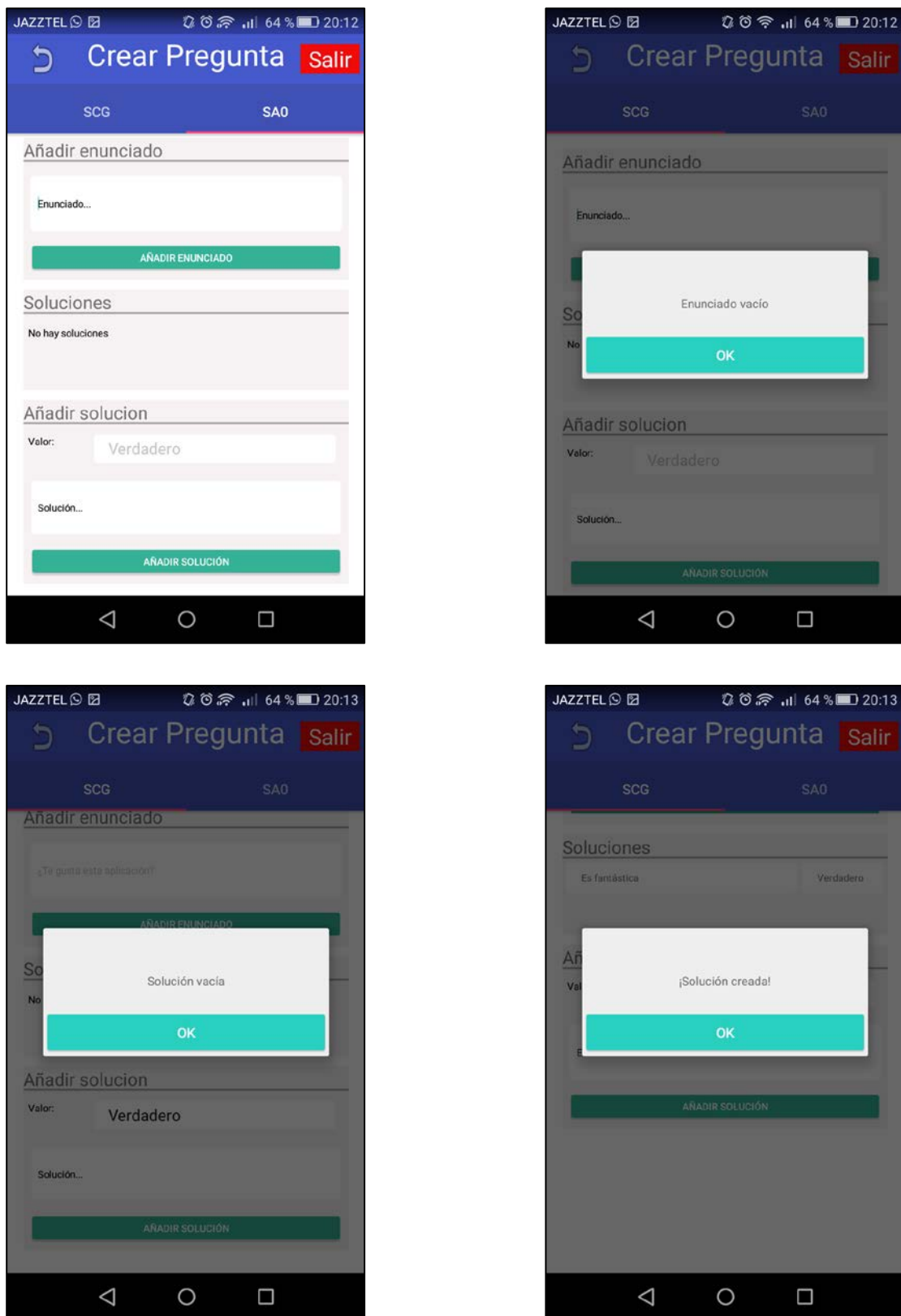
**Figura 22. Detalles de la nota acumulada en un día de un alumno junto con las preguntas respondidas por él**

#### 4.6. Funcionalidad crear preguntas

Esta nueva funcionalidad permite al alumno crear y lanzar preguntas en cualquier asignatura, para que sus compañeros la respondan y también se tenga en cuenta para la nota final la nota sacada en la evaluación continua de la asignatura.

Al entrar en esta funcionalidad, lo primero que se tiene que hacer es crear el enunciado de la pregunta. Si está vacío ese campo no se puede rellenar ningún campo más, por lo que al pulsar añadir enunciado con el campo vacío sale un cuadro de diálogo reportándote el error. Una vez añadido el enunciado, se tienen que crear las soluciones, que se irán mostrando en el cuadro soluciones debajo del enunciado. Al igual que antes, si se pulsa añadir solución estando el campo vacío, se abre un cuadro de diálogo reportando el error. La vista de la funcionalidad al crear el enunciado y las soluciones con los diferentes cuadros de diálogo se pueden observar en todas las capturas de pantalla mostradas en la Figura 23.

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE



**Figura 23. Vista de la funcionalidad crear preguntas y distintos cuadros de diálogo al añadir el enunciado o las diferentes soluciones**

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE

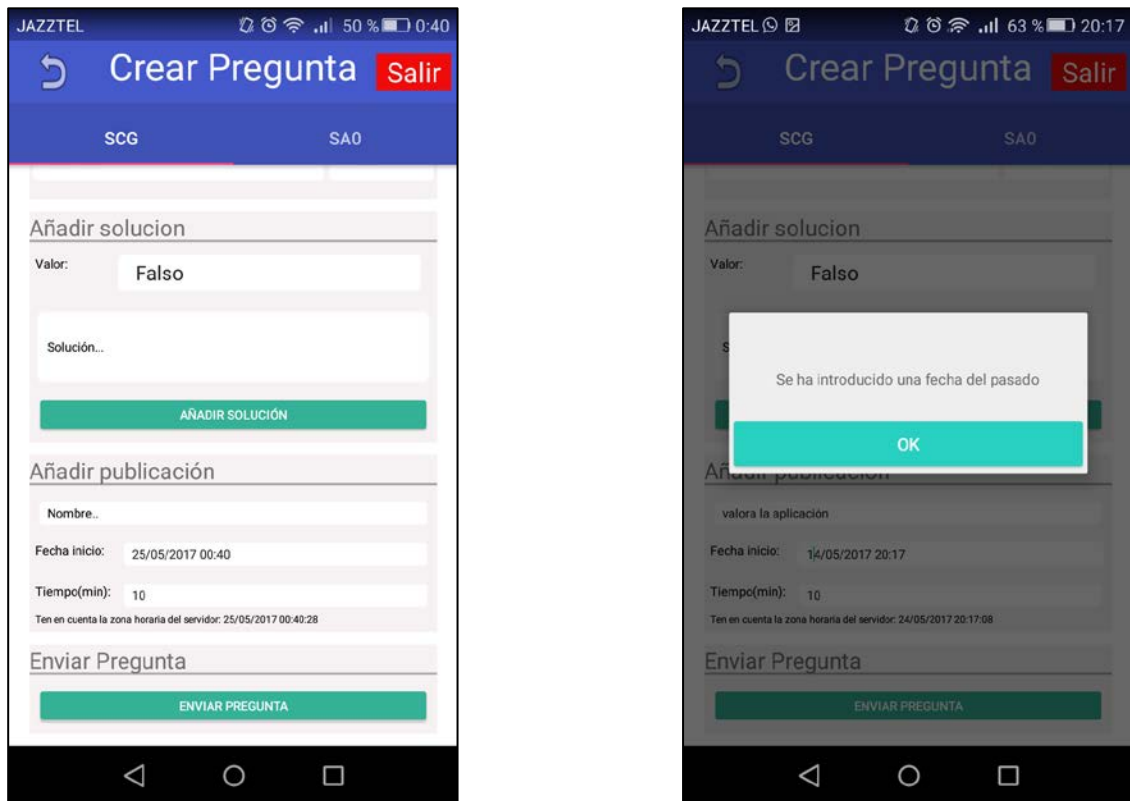
Es obligatorio crear un mínimo de dos soluciones, independientemente de si sus valores son verdaderos o falsos. Una vez añadidas dos soluciones, se vuelve visible el campo de añadir publicación, que se muestra en la Figura 24 (izquierda), con los siguientes campos por rellenar:

- Nombre de la publicación. no puede quedar vacío o al pulsar en el botón de enviar pregunta saldrá un cuadro de diálogo con el error.
- Fecha de inicio de la publicación: debe estar en formato dd/mm/aaaa hh:mm y de ser una fecha u hora anterior a la del servidor, o no estar en el formato indicado, saldrá un cuadro de diálogo indicando el error (Figura 24 derecha). A partir de la fecha y hora elegida se podrá contestar a la pregunta en la funcionalidad preguntas de la asignatura correspondiente.
- Tiempo (min): sólo acepta un entero con el número de minutos que va a estar activa la pregunta a partir de la fecha de inicio. Si no se introduce un número entero o está vacío al pulsar el botón de enviar pregunta, saldrá un cuadro de diálogo con el error.

Además, se ha añadido un texto con la hora del servidor por si el dispositivo inalámbrico estuviera mal configurado o en otra zona horaria.

Todos los datos recogidos para la pregunta se meten en un JSON, que es enviado haciendo una petición mediante un método POST a la API.

#### 4. NUEVAS FUNCIONALIDADES DE LA APLICACIÓN CLIENTE



**Figura 24. Vista de crear una publicación y cuadro de diálogo con error de fecha de inicio anterior a la del servidor**

#### 4.7. Conclusiones

En este capítulo del Trabajo Fin de Grado, se ha presentado el diseño y programación de un conjunto de nuevas funcionalidades integradas en la versión preliminar desarrollada. Además, se puede concluir que se ha diseñado una nueva aplicación totalmente funcional para los dispositivos Android, al máximo semejante a la versión multiplataforma.



# ***5. CONCLUSIONES Y LÍNEAS FUTURAS***

## **5.1. Conclusiones**

El uso de dispositivos móviles se está volviendo cada vez más importante en las nuevas metodologías docentes tanto universitarias como no universitarias, por lo que es de esperar que llegue un día muy cercano, en que el uso de un dispositivo inalámbrico por parte de alumnos y profesores sea algo muy normal. Este Trabajo Fin de Grado, y en general el entorno de aprendizaje virtual AIM-Mobile Learning Platform trata de facilitar la evolución hacia este punto, introduciendo poco a poco su uso en algunas asignaturas para que alumnos y profesores empiecen a familiarizarse con el uso de las tecnologías m-learning y el gran número de ventajas que éstas aportan al proceso de aprendizaje.

Para ello, el entorno virtual está formado por dos aplicaciones, servidora y cliente, con interfaces sencillas y entendibles para que tanto el alumno como el profesor no tengan ningún tipo de problemas al utilizarlas. Cabe mencionar que antes de trabajar en este Trabajo Fin de Grado tuve la suerte de poder probar una de las primeras versiones de la aplicación cliente en una asignatura del grado, por lo que ya sabía cómo funcionaba esta, y en parte me hizo decantarme por hacer la petición para ayudar en este proyecto, ya que me pareció una propuesta muy interesante.

La finalidad de este Trabajo Fin de Grado ha sido por lo tanto diseñar y programar una versión nativa en Android de la aplicación del alumno a partir de la aplicación multiplataforma, con sus mismas funcionalidades. Aunque la versión multiplataforma funcione en Android, lo que se quería probar es la capacidad de sacar más rendimiento a dicha aplicación si se programaba todo con las herramientas de Android.

Personalmente ha sido un reto muy grande este Trabajo Fin de Grado, ya que, aunque tuviera conocimientos básicos de programación aprendidos durante el grado, ha sido

## 5. CONCLUSIONES Y LÍNEAS FUTURAS

empezar de cero, ya que tanto el uso de la herramienta como la mayoría del lenguaje de programación usado ha sido totalmente nuevo para mí.

Por lo tanto, se ha conseguido hacer una aplicación en Android con las mismas funcionalidades que la aplicación multiplataforma, incluidas las nuevas funcionalidades que se han ido integrando en ambas versiones. Tras varias pruebas y análisis de la aplicación Android, hemos concluido que no se observa que su rendimiento sea mucho mejor al de la aplicación multiplataforma. No obstante, no se quiere decir que programar una aplicación en Android no tenga ventajas en este sistema operativo frente a programarla en multiplataforma, sino que al ser una aplicación teóricamente sencilla y sin grandes tareas a nivel computacional no se observan grandes mejoras. En este sentido, probablemente en aplicaciones con una gran carga computacional se observaría que el rendimiento al programar en lenguaje nativo aumentaría.

### 5.2. Líneas futuras

A continuación, se va a proceder a enumerar y describir posibles funcionalidades que se podrían incorporar a la aplicación cliente en un futuro con la finalidad de mejorar su potencialidad y difusión en diferentes entornos educativos:

- Mejorar las vistas al cambiar el tamaño del dispositivo:

Para desarrollar esta aplicación se utilizó un dispositivo de 4.5 pulgadas y otro de 5. Al hacer las pruebas en una tablet, se observó que aunque todas las vistas se visualizaban correctamente, el tamaño de la mayoría de textos era el mismo que el de las otras pantallas, por lo que en dicha pantalla el tamaño no se relativizaba.

- Cambiar tu contraseña:

Se ha dado un paso muy importante introduciendo la funcionalidad de recibir una contraseña en el correo electrónico en caso de haber olvidado la ya existente en el entorno de aprendizaje. Pero se puede observar que resulta muy fastidioso tener que utilizar obligatoriamente una contraseña que te manda el servidor de forma obligatoria sin tener opción a usar una con la que estés más cómodo. Por ello una de las nuevas funcionalidades debería ser poder cambiar la contraseña a tu gusto introduciendo tu contraseña antigua, como se hace en otras muchas aplicaciones hoy en día.

## 5. CONCLUSIONES Y LÍNEAS FUTURAS

- *Diseño de un sistema de notificaciones:*

Sería de gran interés integrar una nueva funcionalidad de modo que cuando un profesor colgara material nuevo, o un compañero introdujera una pregunta para contestarla, que se utilizara el sistema de avisos del dispositivo móvil para que, aunque estuvieras fuera de la aplicación, te pudieras enterar de esa notificación.

- *Diseño de un sistema de intercambio de mensajes:*

Resultaría muy interesante y útil la programación de un pequeño “chat”, en el que se pudiera contactar con el profesor o con compañeros sin necesidad de tener que escribir un correo electrónico, todo ello, desde el entorno de aprendizaje virtual.

- *Descarga de archivos:*

La idea es que esta aplicación se utilice para un intercambio de información fluida con otros profesores y alumnos, por lo que permitir al alumno descargar archivos (por ejemplo en PDF), sería muy útil por ejemplo para ver notas, guía docentes o archivos de apoyo para usar en prácticas.

- *Diseño de un Calendario:*

Otra mejora de gran interés, sería tener un calendario interactivo con la programación y eventos que se van a seguir de la asignatura, con la posibilidad de que el alumno se entere en tiempo real de eventos y posibles modificaciones.

- *Diseño de un sistema de envío de archivos desde la aplicación del alumno:*

También podría ser muy útil que el alumno pudiera enviar archivos al profesor o a los compañeros, tales como memorias de prácticas, entrega de respuestas en forma de foto tomadas desde el dispositivo móvil, etc.

- *Funcionalidad de Perfil de usuario:*

Al igual que en el campus virtual, la idea de tener un perfil que puedas actualizar con una imagen o cierta información sería de gran interés.

- *Integración de un foro:*

Resultaría muy útil la integración de foro en el que se puedan exponer dudas o resolverlas.

## 5. CONCLUSIONES Y LÍNEAS FUTURAS

## 6. BIBLIOGRAFÍA

- [1] TOMÁS GIRONÉS, JESÚS. El Gran Libro de Android. 2a ed. Barcelona: Marcombo S.A., 2012.
- [2] Todo lo que necesitas para realizar compilaciones en Android <https://developer.android.com/studio/features.html?hl=es-419>
- [3] Notepad ++. Editor de código fuente con soporte para diversos lenguajes de programación, gratuito y de código libre. Por Miguel Alvarez. <https://desarrolloweb.com/articulos/notepad.html>
- [4] Advanced Rest Client para probar API por Fran Rives. <http://www.franriveshurtado.com/blog/advanced-rest-client-para-probar-api/>
- [5] Bq Aquaris E4.5 especificaciones en movilcelular.es <http://www.movilcelular.es/movil/bq-aquaris-e45/1968>
- [6] Huawei Y6 II Compact, especificaciones en the phone house <https://www.phonehouse.es/movil/huawei/y6-ii-compact.html>
- [7] Métodos POST Y GET por Marcelo en pixelar.me <http://pixelar.me/metodos-post-get/>
- [8] ¿Qué es una API y para qué sirve? Artículo publicado en la web de ABC. <http://www.abc.es/tecnologia/consultorio/20150216/abci-201502132105.html>
- [9] ¿Qué es y para qué sirve JSON? <https://geekytheory.com/json-i-que-es-y-para-que-sirve-json>
- [10] Foro stack overflow. <https://stackoverflow.com/>