



UNIVERSIDAD DE VALLADOLID

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS DE  
TELECOMUNICACIÓN

TRABAJO FIN DE MÁSTER

MÁSTER UNIVERSITARIO  
EN INGENIERÍA DE TELECOMUNICACIÓN

**Integración de funcionalidades en las  
aplicaciones cliente y servidor del entorno  
virtual AIM-Mobile Learning Platform**

Autor:

**D. Alberto Medina Gilsanz**

Tutor:

**D.<sup>a</sup> Noemí Merayo Álvarez**

Valladolid, 24 de mayo de 2017



This work is licensed under a [Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License](https://creativecommons.org/licenses/by-nc-nd/4.0/).

---

**TÍTULO:** Integración de funcionalidades en las aplicaciones cliente y servidor del entorno virtual AIM-Mobile Learning Platform

**AUTOR:** D. Alberto Medina Gilsanz

**TUTOR:** D.<sup>a</sup> Noemí Merayo Álvarez

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

---

**TRIBUNAL**

---

**PRESIDENTE:** D.<sup>a</sup> Patricia Fernández del Reguero

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

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

**SUPLENTE:**

**P. SUPLENTE:** M.<sup>a</sup> Jesús Verdú Pérez

**S. SUPLENTE:** Javier Manuel Aguiar Pérez

**V. SUPLENTE:** Ramón de la Rosa Steinz

---

**FECHA:** 24 de mayo de 2017

**CALIFICACIÓN:**

---



## **Resumen de TFM**

Este Trabajo Fin de Máster consiste en el desarrollo *software* de una nueva versión de la plataforma *e-learning* AIM-Mobile Learning Platform para integrar nuevas funcionalidades en la aplicación cliente y en la aplicación servidora del entorno virtual de aprendizaje, centrándose principalmente en el desarrollo *software* de nuevas funcionalidades en la plataforma del alumno mediante el *framework* Sencha Touch para obtener una aplicación web accesible desde cualquier navegador web en cualquier momento y en cualquier lugar desde cualquier dispositivo móvil a través de Internet, así como una aplicación nativa en Android e iOS, en la que el alumno, además de visualizar el contenido generado por su profesor como en versiones anteriores de la plataforma de *e-learning*, puede generar su propio contenido en forma de preguntas para mejorar el proceso de aprendizaje o también puede, entre otras nuevas funcionalidades, acceder a la base de datos de su elección independientemente del servidor en el que esté alojada para que cualquier centro de enseñanza pueda instalar su propia plataforma de *e-learning* AIM-Mobile Learning Platform.

## **Palabras clave**

Aprendizaje electrónico, aprendizaje móvil, plataforma de aprendizaje electrónico, desarrollo *software*, aplicación web, aplicación nativa, cliente, servidor, HTML5, CSS3, JavaScript, Sencha Touch, Apache Cordova, Java, Spring Framework.

## **Abstract**

*This Master's Thesis consists of the software development of a new version of the AIM-Mobile Learning Platform e-learning platform to integrate new functionalities in the client application and in the server application of the virtual learning environment, focusing mainly on software development of new functionalities on the student's platform through the Sencha Touch framework to obtain a web application accessible from any web browser anytime and anywhere from any mobile device through the Internet, as well as a native application on Android and iOS, in which the student, in addition to visualizing the content generated by your teacher as in previous versions of the e-learning platform, your own content in the form of questions can be generated to improve the learning process or, among other new functionalities, the database of your choice can be access regardless of the server on which it is hosted so that any educational institution can install its own AIM-Mobile Learning Platform e-learning platform.*

## **Keywords**

*e-learning, m-learning, e-learning platform, software development, web application, native app, client, server, HTML5, CSS3, JavaScript, Sencha Touch, Apache Cordova, Java, Spring Framework.*

## **Agradecimientos**

En primer lugar, gracias por su ayuda a mis amigos y compañeros del Máster en Ingeniería de Telecomunicación, en especial a mis compañeros de la especialidad en Telemática, Luis y Javier, a mis compañeros en los laboratorios de las asignaturas del mismo, Daniel y Fernando, y a compañeros del proyecto *software* AIM-Mobile Learning Platform, Jorge, Álvaro, Guillermo y Samir.

En segundo lugar, gracias por enseñarme tanto a los profesores del Máster en Ingeniería de Telecomunicación, en especial a mi tutora en este TFM, Noemí, por enseñarme, por su ayuda, por sus ánimos, por tirar de mí y por su paciencia conmigo porque no se lo he puesto fácil. De verdad, muchísimas gracias.

Y en último lugar, gracias por su ayuda en el día a día a mi familia, mi madre Alicia y mis hermanos Jorge y Pablo porque tampoco se lo he puesto fácil. Y en especial, también quiero dedicárselo y agradecerse a mi padre Miguel Ángel y a mis abuelos Anastasia, Delfín y Eladio porque, aunque no estén entre nosotros, de una forma o de otra también me han ayudado a lo largo de mis estudios.

# Índice

<b>1</b>	<b>Introducción .....</b>	<b>1</b>
1.1.	Introducción.....	1
1.2.	Objetivos.....	2
1.3.	Fases y métodos .....	3
1.4.	Estructura de la memoria del TFM.....	5
<b>2</b>	<b>Metodología y herramientas de trabajo.....</b>	<b>7</b>
2.1.	Introducción.....	7
2.2.	Entorno virtual AIM-Mobile Learning Platform .....	7
2.3.	Herramientas de desarrollo de la plataforma del profesor .....	9
2.3.1.	Java.....	9
2.3.2.	Spring Framework .....	10
2.3.3.	HTML5, CSS3 y JavaScript .....	12
2.3.4.	Thymeleaf.....	12
2.3.5.	Bootstrap .....	13
2.3.6.	Otras herramientas de desarrollo utilizadas .....	14
2.4.	Herramientas de desarrollo de la plataforma del alumno.....	14
2.4.1.	Sencha Touch .....	15
2.4.2.	Apache Cordova .....	16
2.5.	Herramientas de desarrollo de la API.....	17
2.6.	Entorno de desarrollo.....	17
2.6.1.	Spring Tool Suite .....	17
2.6.2.	Sublime Text.....	18
2.7.	Conclusiones .....	19
<b>3</b>	<b>Entorno virtual AIM-Mobile Learning Platform .....</b>	<b>21</b>
3.1.	Introducción.....	21
3.2.	Descripción de la plataforma del profesor.....	21

3.2.1.	Arquitectura software de la aplicación del profesor.....	21
3.2.2.	Funcionalidades de la plataforma del profesor.....	25
3.2.2.1.	Acceso a la aplicación .....	25
3.2.2.2.	Menú principal .....	26
3.2.2.3.	Tablón de avisos .....	27
3.2.2.4.	Configuración de libros.....	28
3.2.2.5.	Sistemas de evaluación continua del alumnado.....	31
3.2.2.6.	Configuración de asignaturas.....	37
3.2.2.7.	Tareas de administración de la plataforma .....	39
<b>3.3.</b>	<b>Descripción de la plataforma del alumno.....</b>	<b>43</b>
3.3.1.	Arquitectura software de la aplicación del alumno.....	43
3.3.2.	Funcionalidades de la aplicación del alumno .....	45
3.3.2.1.	Registro en la aplicación.....	45
3.3.2.2.	Acceso a la aplicación .....	46
3.3.2.3.	Menú principal de la aplicación del alumno .....	47
3.3.2.4.	Tablón de avisos en la aplicación del alumno .....	47
3.3.2.5.	Libros electrónicos en la aplicación del alumno .....	49
3.3.2.6.	Evaluación continua en la aplicación del alumno.....	51
3.3.2.7.	Información sobre la versión de la aplicación del alumno.....	53
<b>3.4.</b>	<b>Descripción de la API .....</b>	<b>54</b>
3.4.1.	Arquitectura software de la API.....	54
<b>3.5.</b>	<b>Descripción de la base de datos .....</b>	<b>55</b>
3.5.1.	Arquitectura software de la base de datos.....	56
<b>3.6.</b>	<b>Conclusiones .....</b>	<b>62</b>
<b>4</b>	<b>Nuevas funcionalidades de la plataforma del alumno.....</b>	<b>63</b>
<b>4.1.</b>	<b>Introducción.....</b>	<b>63</b>
<b>4.2.</b>	<b>Inicio de conexión con la API y el servidor.....</b>	<b>63</b>
<b>4.3.</b>	<b>Re-inicialización de la contraseña.....</b>	<b>70</b>
<b>4.4.</b>	<b>Manual de ayuda de la aplicación del estudiante .....</b>	<b>74</b>
<b>4.5.</b>	<b>Nuevas funcionalidades en el sistema de evaluación continua .....</b>	<b>76</b>
<b>4.6.</b>	<b>Funcionalidad para crear preguntas.....</b>	<b>78</b>
<b>4.7.</b>	<b>Otras modificaciones.....</b>	<b>83</b>
<b>4.8.</b>	<b>Conclusiones .....</b>	<b>84</b>



<b>5</b>	<b>Conclusiones</b> .....	<b>87</b>
<b>5.1.</b>	<b>Introducción</b> .....	<b>87</b>
<b>5.2.</b>	<b>Conclusiones</b> .....	<b>87</b>
<b>5.3.</b>	<b>Posibles trabajos futuros</b> .....	<b>89</b>



# Índice de figuras

Figura 2.1. Diseño global del entorno virtual de aprendizaje AIM-Mobile Learning Platform desde el punto de vista del desarrollo web [2]. .....	8
Figura 2.2. Ejecución de un programa Java [14]. .....	10
Figura 2.3. Arquitectura de una aplicación en Spring según el patrón MVC [2]. .....	11
Figura 2.4. Responsive web design [26]. .....	13
Figura 2.5. Anatomía de una aplicación en Sencha Touch [33]. .....	16
Figura 3.1. Árbol de directorios del proyecto «AIMML-Admin». .....	22
Figura 3.2. Paquetes Java dentro del directorio «src/main/java» del proyecto «AIMML-Admin». .....	23
Figura 3.3. Ficheros dentro del directorio «src/main/resources» del proyecto «AIMML-Admin». .....	24
Figura 3.4. Acceso a la plataforma del profesor. ....	25
Figura 3.5. Menú principal de la plataforma del profesor. ....	26
Figura 3.6. Tablón de avisos en la plataforma del profesor. ....	27
Figura 3.7. Configuración de libros en la plataforma del profesor. ....	28
Figura 3.8. Edición de un libro electrónico en la plataforma del profesor. ....	29
Figura 3.9. Configuración de preguntas en la plataforma del profesor. ....	31
Figura 3.10. Visualización de una pregunta en la plataforma del profesor. ....	32
Figura 3.11. Publicación de una pregunta en la plataforma del profesor. ....	33
Figura 3.12. Configuración de bancos de preguntas en la plataforma del profesor. ....	34
Figura 3.13. Evaluación individual basado en un sistema de positivos y negativos en la plataforma del profesor. ....	35
Figura 3.14. Evaluación de laboratorio en la plataforma del profesor. ....	36
Figura 3.15. Configuración de asignaturas en la plataforma del profesor. ....	37
Figura 3.16. Creación de estudiantes en la plataforma del profesor. ....	37
Figura 3.17. Resultados de un estudiante de una asignatura en la plataforma del profesor. ....	38
Figura 3.18. Configuración de usuarios en la plataforma del profesor. ....	39
Figura 3.19. Configuración de asignaturas en la plataforma del profesor. ....	40
Figura 3.20. Configuración y creación de nuevos estudiantes dados de alta en la plataforma del profesor. ....	40
Figura 3.21. Detalle de un estudiante en la plataforma del profesor. ....	41

Figura 3.22. Copias de seguridad de la plataforma del profesor. ....	42
Figura 3.23. Otras funcionalidades de administración en la plataforma del profesor. .....	42
Figura 3.24. Árbol de directorios de la plataforma del alumno. ....	44
Figura 3.25. Registro en la aplicación del alumno (versión web).....	45
Figura 3.26. Registro en la aplicación del alumno (versión Android).....	45
Figura 3.27. Acceso a la aplicación del alumno (versión web).....	46
Figura 3.28. Acceso a la aplicación del alumno (versión Android).....	46
Figura 3.29. Menú principal en la aplicación del alumno (versión web). ....	47
Figura 3.30. Menú principal en la aplicación del alumno (versión Android). ....	47
Figura 3.31. Tablón de avisos en la aplicación del alumno (versión web). ....	48
Figura 3.32. Tablón de avisos en la aplicación del alumno (versión Android). ....	48
Figura 3.33. Detalle de un aviso en la plataforma del alumno (versión web). ....	48
Figura 3.34. Detalle de un aviso en la plataforma del alumno (versión Android). ....	48
Figura 3.35. Libros electrónicos en la aplicación del alumno (versión web).....	49
Figura 3.36. Libros electrónicos en la aplicación del alumno (versión Android). ....	49
Figura 3.37. Páginas de un libro electrónico en la aplicación del alumno (versión web). ....	50
Figura 3.38. Páginas de un libro electrónico en la aplicación del alumno (versión Android). ....	50
Figura 3.39. Página de un libro electrónico en la aplicación del alumno (versión web). ....	50
Figura 3.40. Página de un libro electrónico en la aplicación del alumno (versión Android). ....	50
Figura 3.41. Preguntas publicadas en la aplicación del alumno (versión web). ....	51
Figura 3.42. Preguntas publicadas en la aplicación del alumno (versión Android). ..	51
Figura 3.43. Pregunta publicada en la aplicación del alumno (versión web).....	52
Figura 3.44. Pregunta publicada en la aplicación del alumno (versión Android). ....	52
Figura 3.45. Evolución de las notas en la plataforma del alumno (versión web). ....	52
Figura 3.46. Evolución de las notas en la plataforma del alumno (versión Android).52	
Figura 3.47. Información en la aplicación del alumno (versión web). ....	53
Figura 3.48. Información en la aplicación del alumno (versión Android). ....	53
Figura 3.49. Árbol de directorios de la API. ....	54
Figura 3.50. Paquetes Java dentro del directorio «src/main/java» del proyecto «AIMML-core». ....	56

Figura 3.51. Ficheros Java dentro del paquete «es.uva.tel.aimml.core.domain» del directorio «src/main/java» del proyecto «AIMML-core».....	57
Figura 3.52. Ficheros Java dentro del paquete «es.uva.tel.aimml.core.repository» del directorio «src/main/java» del proyecto «AIMML-core».....	58
Figura 3.53. Diagrama de la base de datos de AIM-Mobile Learning Platform. ....	59
Figura 4.1. Inicio de conexión con la API y el servidor en la aplicación del alumno (versión web).....	64
Figura 4.2. Inicio de conexión con la API y el servidor en la aplicación del alumno (versión Android). ....	64
Figura 4.3. Conexión con la API y el servidor iniciada en la aplicación del alumno (versión web).....	65
Figura 4.4. Conexión con la API y el servidor iniciada en la aplicación del alumno (versión Android). ....	65
Figura 4.5. Opción de almacenar localmente la URL (versión web).....	65
Figura 4.6. Opción de almacenar la URL localmente (versión Android).....	65
Figura 4.7. URL localmente almacenada (versión web). ....	66
Figura 4.8. URL localmente almacenada (versión Android). ....	66
Figura 4.9. Conexión con la API y el servidor iniciada (versión web). ....	67
Figura 4.10. Conexión con la API y el servidor iniciada (versión Android). ....	67
Figura 4.11. Conexión con la API y el servidor no iniciada (versión web). ....	67
Figura 4.12. Conexión con la API y el servidor no iniciada (versión Android). ....	67
Figura 4.13. URL vacía (versión web).....	68
Figura 4.14. URL vacía (versión Android).....	68
Figura 4.15. Inicio de conexión con la API y el servidor con la URL localmente almacenada (versión web).....	69
Figura 4.16. Inicio de conexión con la API y el servidor con la URL localmente almacenada (versión Android).....	69
Figura 4.17. Inicio de conexión con la API y el servidor con la URL y las credenciales localmente almacenada (versión web).....	69
Figura 4.18. Inicio de conexión con la API y el servidor con la URL y las credenciales localmente almacenada (versión Android).....	69
Figura 4.19. Proceso de reinicio de cuenta en la aplicación del profesor.....	70
Figura 4.20. Re-inicializar contraseña (versión web).....	71
Figura 4.21. Re-inicializar contraseña (versión Android).....	71
Figura 4.22. Mensaje de re-inicialización de contraseña (versión web). ....	71
Figura 4.23. Mensaje de re-inicialización de contraseña (versión Android). ....	71

Figura 4.24. Introducción de nombre de usuario para re-inicializar contraseña (versión web). .....	72
Figura 4.25. Introducción de nombre de usuario para re-inicializar contraseña (versión Android). .....	72
Figura 4.26. Mensaje de alerta en la re-inicialización de contraseña (estudiante) (versión web). .....	73
Figura 4.27. Mensaje de alerta en la re-inicialización de contraseña (estudiante) (versión Android). .....	73
Figura 4.28. Mensaje de alerta en la re-inicialización de contraseña (correo electrónico) (versión web). .....	73
Figura 4.29. Mensaje de alerta en la re-inicialización de contraseña (correo electrónico) (versión Android). .....	73
Figura 4.30. Manual de ayuda de la aplicación (versión web). .....	75
Figura 4.31. Manual de ayuda de la aplicación (versión Android). .....	75
Figura 4.32. Ventana emergente de información sobre una funcionalidad principal (versión web). .....	75
Figura 4.33. Ventana emergente de información sobre una funcionalidad principal (versión Android). .....	75
Figura 4.34. Lista de preguntas respondidas (versión web). .....	77
Figura 4.35. Lista de preguntas respondidas (versión Android). .....	77
Figura 4.36. Detalles de nota (versión web). .....	78
Figura 4.37. Detalles de nota (versión Android). .....	78
Figura 4.38. Funcionalidad para crear preguntas (versión web). .....	79
Figura 4.39. Funcionalidad para crear preguntas (versión Android). .....	79
Figura 4.40. Crear pregunta (versión web). .....	79
Figura 4.41. Crear pregunta (versión Android). .....	79
Figura 4.42. Enunciado creado (versión web). .....	80
Figura 4.43. Enunciado creado (versión Android). .....	80
Figura 4.44. Solución creada (versión web). .....	81
Figura 4.45. Solución creada (versión Android). .....	81
Figura 4.46. Creación de publicación y pregunta (versión web). .....	81
Figura 4.47. Creación de publicación y pregunta (versión Android). .....	81
Figura 4.48. Publicación creada (versión web). .....	82
Figura 4.49. Publicación creada (versión Android). .....	82
Figura 4.50. Autoría y opción de borrado de la publicación creada desde la plataforma del alumno en la plataforma del profesor. ....	83

# 1

## Introducción

### 1.1. Introducción

En la Sociedad de la Información, entendiendo esta como el conjunto de transformaciones sociales y económicas producidas como consecuencia del desarrollo exponencial y convergente de redes y servicios de telecomunicaciones, tecnologías de la información y medios de comunicación, Internet no es solo un instrumento de búsqueda de información y de comunicación, sino que se ha convertido en un instrumento para la formalización de transacciones comerciales entre organizaciones o entre.

La forma en que las organizaciones han logrado involucrar las *e-technologies*, tecnologías de la información basadas en Internet, con sus necesidades, se ha producido a través de nuevos servicios basados en Internet, denominados *e-services*, lo que ha modificado sus procesos de negocio.

Uno de los principales *e-services* es el *e-learning* que hace referencia al uso de nuevas tecnologías de información, multimedia y de Internet para mejorar la calidad del aprendizaje mediante el acceso a recursos, servicios, a colaboraciones o a intercambios a larga distancia. Es un término que cubre un amplio grupo de aplicaciones y procesos, tales como aprendizaje basado en web, aprendizaje basado en ordenadores, aulas virtuales y colaboración digital.

La evolución natural del *e-learning*, basado en tecnologías web principalmente, y el auge de las tecnologías móviles ha dado lugar al concepto *m-learning*, metodología de aprendizaje que hace uso de dispositivos móviles tales como *smartphones*, *tablets* o cualquier dispositivo de mano que disponga de conexión inalámbrica.

Tanto el *e-learning* como el *m-learning* sirven de apoyo al proceso de enseñanza-aprendizaje empleando diferentes soportes. Una plataforma de *e-learning* integra un conjunto de herramientas para el aprendizaje *online*, permitiendo una enseñanza de tipo no presencial (*e-learning*) y/o mixta (*b-learning* o *blended-learning*), donde se combinan las clases presenciales con la formación a través de una plataforma de *e-learning*.<sup>[1]</sup>

La plataforma de aprendizaje *e-learning* que nos atañe en este proyecto, denominada AIM-Mobile Learning Platform, desarrollada por el Grupo de Comunicaciones Ópticas de la Universidad de Valladolid en proyectos anteriores, está orientada a ser aplicable a cualquier nivel o disciplina educativa, de enseñanza primaria, media o superior. Como toda plataforma de *e-learning*, tiene multitud de funcionalidades como autenticación al sistema, generación y visualización de contenidos, etc.

El entorno de aprendizaje virtual AIM-Mobile Learning Platform dispone de dos aplicaciones, una aplicación cliente y una aplicación servidora. La aplicación servidora consiste en una aplicación web accesible desde cualquier navegador en cualquier dispositivo a la que un profesor puede acceder con sus credenciales. Dentro de ella, el usuario profesor podrá crear y editar contenidos educativos, tales como avisos en tiempo real, libros electrónicos enriquecidos, preguntas para evaluar de forma continua a sus alumnos y administrar alumnos y asignaturas, todo ello a través de una interfaz de usuario sencilla e intuitiva. Mientras, la aplicación cliente, además de consistir en una aplicación web accesible desde cualquier navegador, también dispone de aplicaciones nativas para sistemas operativos móviles como Android o iOS, a la que un alumno puede acceder con sus credenciales. Desde ella, el usuario alumno podrá visualizar el contenido creado desde la aplicación servidora y responder a las preguntas en tiempo real a través de una interfaz de usuario también muy fácil e intuitiva. Estas características convierten al entorno virtual de aprendizaje AIM-Mobile Learning Platform en un sistema multiplataforma rápido, simple y fácil de usar.

## 1.2. Objetivos

En este TFM (Trabajo Fin de Máster) se pretende cumplir un objetivo a nivel general que se puede dividir en una serie de objetivos a nivel específico:

- **Objetivo General:** el principal objetivo de este TFM es desarrollar una nueva versión del *software* AIM-Mobile Learning Platform a partir de la versión 3.1.4 previamente desarrollada en proyectos académicos anteriores [2]–[4]. Por lo tanto, se pretende añadir nuevas funcionalidades y mejoras al entorno virtual, el cual ofrece una vía nueva de comunicación de contenidos educativos, respecto a la clásica docencia, y un sistema de evaluación continua en tiempo real en línea usando metodologías *e-learning* y *m-learning*.
- **Objetivos Específicos:**
  - Desarrollar nuevas funcionalidades en la aplicación cliente, es decir, a la plataforma del alumno.
  - Desarrollar nuevas funcionalidades en la aplicación servidora, es decir, a la plataforma del profesor, a la API (*Application Programming Interface*, Interfaz de Programación de Aplicación) y a la base de datos.
  - Mantener la versión actual de AIM-Mobile Learning Platform, es decir, corregir los errores, no detectados en fases de prueba anteriores,



encontrados por los usuarios al distribuir el *software* o lo que es lo mismo, al entrar AIM-Mobile Learning Platform en producción.

## 1.3. Fases y métodos

En este TFM, para cumplir los objetivos, se pretende seguir una metodología que consta de las siguientes fases:

1. **Fase de investigación:** en esta primera fase se analiza y se estudia la última versión del entorno virtual de aprendizaje AIM-Mobile Learning Platform. Para ello, en primer lugar, desde el punto de vista del usuario, tanto la aplicación cliente o plataforma del alumno como la aplicación servidora o plataforma del profesor. Y, en segundo lugar, desde el punto de vista del desarrollador *software*, tanto la aplicación cliente como la aplicación servidora, plataforma del profesor, API y base de datos. En concreto se realizan los siguientes pasos:
  1. Se repasa el lenguaje de programación Java a partir de apuntes de asignaturas del Máster en Ingeniería de Telecomunicación [5].
  2. Se repasan las versiones actuales de los lenguajes de desarrollo web (HTML5, CSS3, JavaScript, etc.) a partir de apuntes de asignaturas de la titulación de Ingeniero Técnico de Telecomunicación, especialidad en Telemática [6] y de ejemplos, referencias y tutoriales [7].
  3. Se estudia Spring Framework (Spring Boot, Spring Framework, Spring Tool Suite, Spring Data, Spring Security, etc.) a partir de guías, tutoriales y documentación de referencia [8].
  4. Se estudia Sencha Touch a partir de guías, tutoriales, documentación de referencia [9], [10].
  5. Se estudia Apache Cordova a partir de guías, tutoriales y documentación de referencia [11].
  6. Se estudia el resto de herramientas de desarrollo *software* a partir de guías, tutoriales y documentación de referencia en sus sitios web oficiales.
2. **Fase de análisis:** en esta segunda fase se decide qué se supone que han de hacer las nuevas funcionalidades que se van a desarrollar en el entorno virtual AIM-Mobile Learning Platform, es decir, se especifican los requisitos que tiene que cumplir la nueva versión de la plataforma. Para ello, de forma conjunta con la tutora de este TFM como futura usuaria del entorno virtual de aprendizaje, se analizan y definen los problemas a resolver. Problemas como la poca participación del alumno en el aprendizaje (solo se limita a visualizar contenido y responder preguntas en la aplicación cliente) o la falta de información sobre las funcionalidades en la aplicación del alumno, etc. Y a partir de estos problemas, se especifican los requisitos: el alumno tiene que poder crear sus propias preguntas, ventanas de información sobre las diferentes opciones del menú principal, etc.

3. **Fase de diseño:** una vez que se conoce qué ha de hacer la nueva versión del entorno virtual AIM-Mobile Learning Platform se necesita indicar cómo debe conseguirlo, qué flujo deben seguir las nuevas funcionalidades para poder cumplir los requisitos establecidos. Para ello, en esta fase se especifica qué hace cada nuevo módulo y sus relaciones con el resto, puesto que la versión previa del entorno virtual de aprendizaje se divide en módulos para facilitar el desarrollo de nuevas funcionalidades. Hay que indicar que esta fase y la anterior se desarrollan de la misma forma sea cuál sea el lenguaje de programación a utilizar.
4. **Fase de implementación:** en esta etapa de implementación o codificación se traduce el diseño en un programa escrito en un lenguaje de programación. Se implementa progresivamente cada nueva característica codificándola, depurándola y probándola independientemente. Para ello, principalmente se implementan nuevas funcionalidades en la aplicación cliente mediante el *framework* Sencha Touch gracias a la ayuda de la documentación [9], [10]. Funcionalidades como la creación de preguntas por parte del alumno, ventanas emergentes de información sobre cada uno de los apartados del menú principal de la aplicación del alumno, la posibilidad de obtener una nueva contraseña en caso de olvido por parte del alumno o la opción de poder revisar las preguntas respondidas a lo largo del curso junto a las respuestas dadas para un mejor aprendizaje, etc. Simultáneamente, en la aplicación servidora (principalmente en la API) se implementan las nuevas características que guardan relación con la aplicación del alumno, es decir, que responden a las peticiones realizadas desde la aplicación cliente para ofrecer el servicio. Para ello, y en ambos casos, aplicación cliente y servidora, se implementan las nuevas funcionalidades siguiendo el modelo de los módulos previamente desarrollados en versiones anteriores de la plataforma, además de la documentación ya mencionada.
5. **Fase de depuración:** en esta fase se deben encontrar y corregir o eliminar los errores de codificación, tanto de compilación como de ejecución y lógicos. Para ello, se desarrollan secuencias de *testing* que intentan abarcar cualquier posible uso del entorno virtual de aprendizaje, tanto en su versión web como en las aplicaciones nativas de la plataforma del alumno.
6. **Fase mantenimiento:** una vez que se desarrolla el *software* del entorno virtual AIM-Mobile Learning Platform, es decir, las nuevas funcionalidades, se distribuye entre los usuarios, y entra en producción. Término comúnmente conocido en el ámbito del desarrollo *software*. Aun así, debe ser mantenido y actualizado por si los usuarios encuentran errores no detectados en la fase anterior o se necesitan modificar o añadir nuevas características. Si los usuarios del entorno virtual de aprendizaje observan algún error o sugieren alguna modificación, se inicia de nuevo el ciclo de vida de desarrollo *software* en un proceso cíclico continuo de mejora.
7. **Fase de documentación:** no es una etapa en sí sino un proceso continuo que se da durante el resto de fases. Se diferencia entre la documentación interna, incluida entre las líneas de código en forma de comentarios, y la externa que

incluye análisis, manuales de usuario, etc. Hay que destacar que para el desarrollo de esta nueva versión del entorno virtual de aprendizaje AIM-Mobile Learning Platform se utiliza Git [12] como VCS (*Version Control System*, Sistema de Control de Versiones) que ayuda a gestionar los diferentes cambios que se realizan en el código fuente de su desarrollo *software*.

## 1.4. Estructura de la memoria del TFM

La estructura de la memoria de este TFM se ha dividido en los siguientes capítulos:

- Capítulo 2: se describe la metodología de trabajo y las principales herramientas de trabajo empleadas en el desarrollo *software* del entorno AIM-Mobile Learning Platform. En primer lugar, se describe brevemente el entorno virtual de aprendizaje. En segundo lugar, se describen las principales herramientas de desarrollo *software* usadas en la aplicación servidora y en la aplicación cliente.
- Capítulo 3: se describe el entorno virtual de aprendizaje desde el punto de vista del desarrollador *software* describiendo su arquitectura. Del mismo modo, desde el punto de vista del usuario se describen las principales funcionalidades de cada una de las partes de las que consta el entorno de aprendizaje virtual.
- Capítulo 4: se describe de forma detallada las nuevas funcionalidades de la aplicación cliente, aunque de forma indirecta también están relacionadas con pequeñas modificaciones en la aplicación servidora. Es decir, se describen las nuevas funcionalidades implementadas en la nueva versión, así como los ficheros creados y modificados en los proyectos *software* del entorno virtual.
- Capítulo 5: se realiza un resumen de las principales conclusiones y posibles líneas futuras a seguir para potenciar y mejorar el entorno virtual de aprendizaje AIM-Mobile Learning Platform.



# 2

## Metodología y herramientas de trabajo

### 2.1. Introducción

En este Capítulo 2 se describe la metodología y las herramientas de trabajo que se emplean para desarrollar el entorno virtual de aprendizaje AIM-Mobile Learning Platform. En primer lugar, se describe el entorno virtual de aprendizaje AIM-Mobile Learning Platform a nivel global. En segundo lugar, se describen las herramientas de trabajo empleadas en el desarrollo del entorno virtual de aprendizaje AIM-Mobile Learning Platform que se divide en la plataforma del profesor, la plataforma del alumno, la API y la base de datos. Y en último lugar, se describe el entorno utilizado para desarrollar el entorno virtual de aprendizaje AIM-Mobile Learning Platform, es decir, la plataforma del profesor, la plataforma del alumno, la API y la base de datos.

### 2.2. Entorno virtual AIM-Mobile Learning Platform

El entorno virtual de aprendizaje AIM-Mobile Learning Platform, desde el punto de vista del desarrollo web, se puede dividir principalmente en dos partes: el *back-end* y el *front-end*. En el desarrollo web, esta división hace referencia al lado del cliente de un sitio web conocido como *front-end* y al lado del servidor conocido como *back-end*.

Desde el lado del cliente o *front-end* de un sitio web, que en AIM-Mobile Learning Platform se corresponde con la plataforma del alumno, un usuario principalmente consume contenidos y en algunos casos puede enviar datos. Mientras que desde el lado del servidor o *back-end*, que en nuestra plataforma se corresponde principalmente con la plataforma del profesor, un usuario administrador genera contenidos y puede visualizar o tratar con los datos enviados por el cliente. Por otro lado, en el caso de que el *front-end* no pueda acceder

directamente a los datos, se utiliza una API, tal y como utilizamos en nuestro entorno virtual de aprendizaje.

En AIM-Mobile Learning Platform, los datos que se envían desde el *front-end* se almacenan de forma persistente en una base de datos relacional en el lado del servidor. Además, del lado del *front-end*, pero fuera del entorno web se desarrolla una aplicación nativa para diferentes plataformas móviles. En la Figura 2.1 se puede ver el diseño global del entorno virtual de aprendizaje AIM-Mobile Learning Platform desde el punto de vista del desarrollo web.

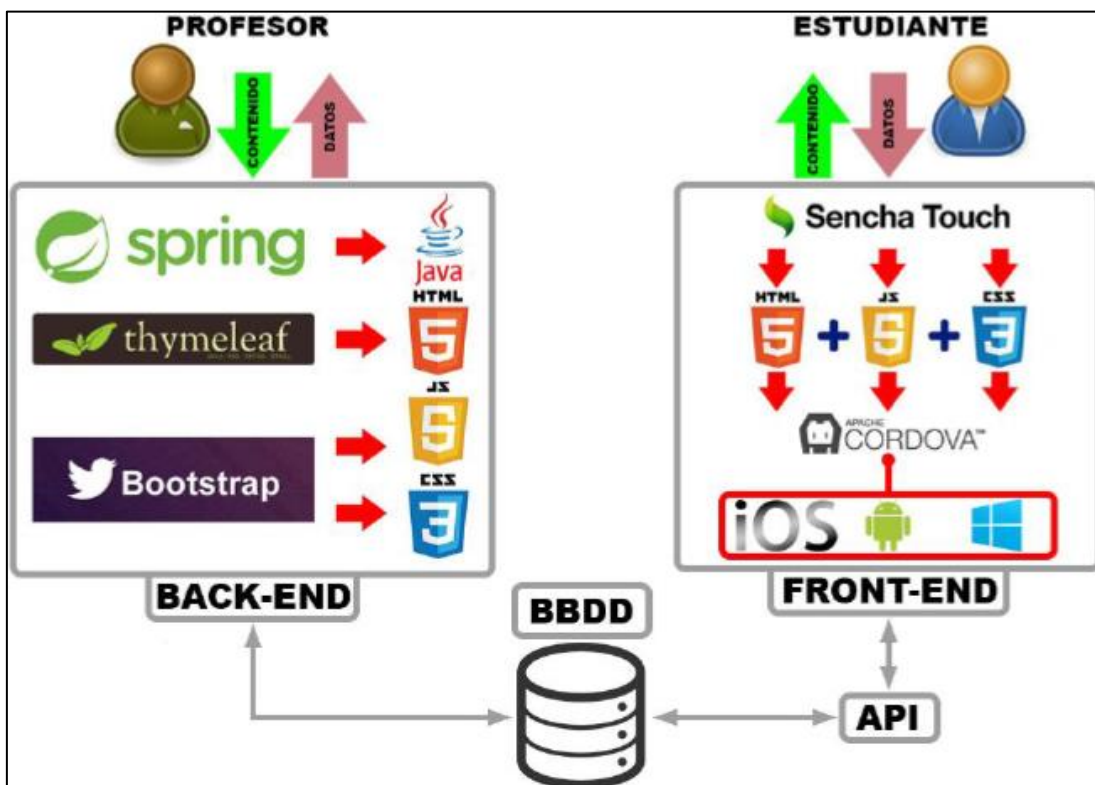


Figura 2.1. Diseño global del entorno virtual de aprendizaje AIM-Mobile Learning Platform desde el punto de vista del desarrollo web [2].

Desde la plataforma del profesor se pueden crear contenidos a consumir desde la plataforma del alumno y gestionar los datos enviados por los diferentes alumnos. El desarrollo de la plataforma del profesor se realiza mediante Spring utilizando Java como lenguaje de programación y mediante Thymeleaf y Bootstrap utilizando HTML5, CSS3 y JavaScript para el desarrollo de la interfaz de usuario. Así pues, el acceso a la plataforma del profesor se puede realizar desde cualquier navegador web.

Desde la plataforma del alumno se pueden consumir los contenidos creados desde la plataforma del profesor y dependiendo de la funcionalidad seleccionada, enviar datos a la plataforma del profesor. El desarrollo de la plataforma del alumno se realiza mediante Sencha Touch utilizando HTML5, CSS3 y JavaScript. Así pues, el acceso a la plataforma del alumno se puede realizar desde cualquier navegador web, aunque se recomienda el uso de Chrome o Safari. Además, también se puede acceder a la plataforma del alumno desde la aplicación nativa, disponible en

diferentes sistemas operativos móviles como Android o iOS, gracias a la herramienta Apache Cordova.

Finalmente, para el intercambio de datos en formato JSON (*JavaScript Object Notation*, Notación de Objeto JavaScript) entre la plataforma del alumno y la base de datos, se desarrolla una API que, al igual que la plataforma del profesor, también se realiza mediante Spring utilizando Java como lenguaje de programación.

## 2.3. Herramientas de desarrollo de la plataforma del profesor

En este apartado se describen las herramientas de trabajo que se emplean en el desarrollo de la plataforma del profesor.

Un usuario, normalmente un profesor, puede acceder a la plataforma del profesor desde cualquier navegador web en cualquier ordenador o dispositivo móvil puesto que el diseño web de la plataforma del profesor se adapta automáticamente a cualquier tamaño de pantalla para mantener la usabilidad del entorno virtual de aprendizaje.

La plataforma del profesor se emplea principalmente para crear contenidos a visualizar desde la plataforma del alumno y gestionar los datos que envían los estudiantes. Dentro de la plataforma, el administrador tendrá la capacidad de dar de alta a los profesores, a los que se les podrá asociar una o varias asignaturas en las que podrán crear avisos, libros electrónicos, preguntas para llevar a cabo una evaluación continua del alumnado, etc. Todos estos contenidos podrán ser visualizados desde la plataforma del alumno por los estudiantes matriculados. Desde la plataforma del profesor, también se podrán gestionar usuarios, asignaturas, estudiantes y las copias de seguridad si el usuario tiene un rol administrativo.

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

A continuación, se describen de forma resumida las principales herramientas de trabajo empleadas en el desarrollo de la plataforma del profesor.

### 2.3.1. Java

Java [13] es un lenguaje de programación de alto nivel orientado a objetos cuya sintaxis es similar a C++. Sus características básicas son:

- No usa explícitamente punteros, es decir, no hay operaciones explícitas sobre punteros accesibles para el programador.
- Solo admite herencia simple, es decir, cada clase sólo puede tener una superclase.

- Para ejecutar un programa Java se necesita la plataforma Java, es decir, un entorno *hardware* (tipo de ordenador) y *software* (sistema operativo) en el que se ejecuta un programa Java.

La plataforma Java está formada por la JVM (*Java Virtual Machine*, Máquina Virtual Java), esto es, un *software* que emula una máquina virtual cuyo código máquina son los Java *bytecodes* y por la Java API, conjunto de clases predefinidas que proporcionan funciones básicas. La JVM se implementa de una forma o de otra en función de la plataforma de ejecución (MS-Windows sobre PC, Mac OS X sobre Macintosh, etc.). El código Java se compila para producir un fichero de Java *bytecodes* independientemente de la plataforma de ejecución y este es interpretado por la JVM y traducido a instrucciones específicas de la plataforma de ejecución. Esta característica permite ejecutar un programa Java en cualquier plataforma de ejecución como se puede observar en la Figura 2.2.

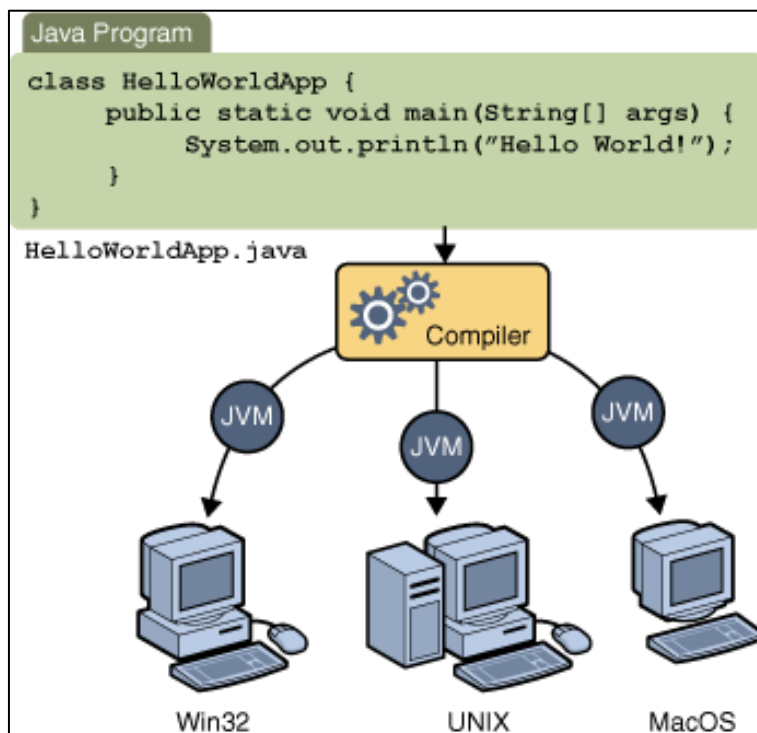


Figura 2.2. Ejecución de un programa Java [14].

### 2.3.2. Spring Framework

Spring Framework [15], también conocido como Spring, es un *framework* de código abierto para desarrollo de aplicaciones dentro de la plataforma Java.

Algunas de las características principales son la IoC (*Inversion of Control*, Inversión de Control), que permite hacer que la configuración sea rápida y sencilla, desacoplando así la configuración y especificación de dependencias de la lógica de la aplicación. La DI (*Dependency Injection*, Inyección de Dependencias), actúa de forma conjunta con IoC para poder incluir las dependencias entre objetos de forma transparente para el desarrollador. Y finalmente el paradigma de programación



AOP (*Aspect-Oriented Programming*, Programación Orientada a Aspectos), lo que permite tener aspectos de gestión de transacciones.

Otro de los aspectos más importantes es la colección de módulos que se pueden incluir por separado. Para este proyecto se han incluido dos módulos:

- Spring Data JPA [16] es un módulo del proyecto Spring Data [17]. Su funcionalidad es facilitar el acceso y explotación a diferentes tipos de tecnologías de persistencia, en este caso a base de datos relacionales. Su principal característica es la implementación de repositorios, del cual provee varias interfaces genéricas para implementar de forma transparente funciones de CRUD (*Create, Read, Update and Delete*; Crear, Leer, Actualizar y Eliminar), métodos de búsqueda, ordenación y paginación.
- Spring Security [18] es un módulo que proporciona una estructura de control de seguridad mediante autenticación. Además, ofrece diferentes filtros de seguridad y control de acceso por diferentes niveles de acceso mediante roles de usuario.

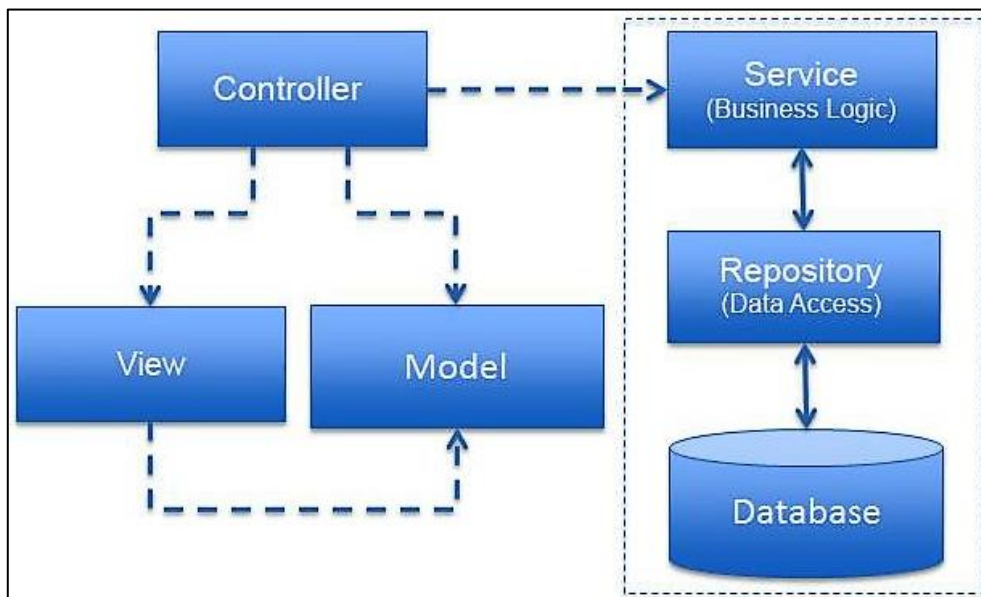


Figura 2.3. Arquitectura de una aplicación en Spring según el patrón MVC [2].

En Spring también se desarrolla una arquitectura *software* según el patrón MVC (*Model-View-Controller*, Modelo-Vista-Controlador) como se puede ver en la Figura 2.3 y consta de las siguientes partes:

- «Model» (Modelo): implementa la capa de dominio donde se encontrarán definidas todas las entidades de datos que necesita la aplicación.
- «View» (Vista): implementa la capa de presentación donde se desarrollará la interfaz de usuario de la aplicación.
- «Controller» (Controlador): implementa la capa de control encargada de gestionar el flujo de la aplicación mediante el direccionamiento de cada llamada realizada por el cliente.

- «Service» (Servicio): implementa la capa de negocio encargada de ejecutar las operaciones de la lógica de negocio llevando el control de las mismas; y además permite interactuar con las entidades mediante los repositorios.
- «Repository» (Repositorio): implementa la capa de persistencia de la aplicación donde se encontrarán las transacciones con la base de datos.

Gracias a todas estas características se consigue tener una aplicación escalable, de fácil configuración y con una gran integración con diversas características tanto propias como externas.

### 2.3.3. HTML5, CSS3 y JavaScript

HTML5 es la quinta revisión de HTML [19], herramienta que permite la creación y edición de documentos web. Está definido a partir de SGML (*Standard Generalized Markup Language*, Lenguaje de Marcado Generalizado Estándar) basado en etiquetas (*tags*). HTML5 incluye nuevos elementos y atributos que reflejan el uso de los sitios modernos como la inserción de multimedia de audio, video o contenidos gráficos.

CSS3 es la última revisión de CSS [20], lenguaje de estilo que se utiliza para definir la presentación de documentos HTML. Con esta herramienta se definen cuestiones relativas a fuentes, colores, márgenes, líneas, altura, anchura, imágenes de fondo y otras muchas opciones de estilo.

JavaScript es un lenguaje de programación interpretado y está orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Se utiliza para implementar en sitios web y en el lado de cliente, mejoras o personalizaciones en la interfaz del usuario y en páginas web dinámicas.

### 2.3.4. Thymeleaf

Thymeleaf [21] es una librería Java de código abierto que implementa un motor de plantillas XML (*eXtensible Markup Language*, Lenguaje de Marcado Extensible), XHTML (*eXtensible HTML*, HTML Extensible) y HTML5 siendo también extensible a otros formatos. Dentro de la implementación del patrón MVC de la plataforma del profesor se encargará de generar toda la capa de la vista y permite una fácil integración con Spring Framework.

Una de sus principales características es el soporte para plantillas naturales permitiendo que las plantillas sean prototipos válidos para trabajar y que se puedan ver en un navegador sin ser procesadas. Esto permite agilizar el proceso de desarrollo ya que permite trabajar de forma paralela tanto en su diseño como en su desarrollo.

Otra de sus características más interesantes es el soporte para la evaluación de expresiones, esto es, tanto el módulo estándar conocido como OGNL (*Object-Graph Navigation Language*) como el de *Spring* conocido como SpringEL (*Spring Expression Language*). Gracias a este soporte se puede realizar externalización de texto para

ofrecer internacionalización dentro de la plataforma, realizar comparaciones e igualdades y finalmente operaciones de texto, aritméticas o booleanas. Además, ofrece operaciones de iteración o soporte de patrones de composición de vista llamados fragmentos.

Por último, a esta herramienta se le puede incluir un módulo extra llamado Thymeleaf Extras Spring Security [22] que facilita la integración con Spring Security consiguiendo un nuevo dialecto, lo cual ofrece varias expresiones y atributos para acceder a datos de autenticación desde la vista.

### 2.3.5. Bootstrap

Bootstrap [23] es una herramienta de código abierto que proporciona un *framework* de HTML, CSS y JavaScript que permite simplificar el proceso de creación de sitios y aplicaciones web. Permite estilizar multitud de elementos HTML desde tablas, formularios o botones. Además, proporciona multitud de componentes propios como una barra de navegación, paginación, barra de progreso, mensajes de alerta contextual, paneles, pestañas, menús desplegables, e incluso una librería de iconos. Para poder usar todos los componentes de JavaScript que vienen incorporados requiere trabajar de forma conjunta con la librería jQuery [24]. Además, se usará de forma conjunta con la librería de iconos Font Awesome [25] para completar su galería de iconos.

Una de sus características más interesantes es el diseño web adaptativo (*responsive web design*), técnica que se usa para tener una misma web adaptada a las diferentes plataformas como son ordenadores, tabletas o dispositivos móviles, proporcionando una mejor experiencia de usuario. Para conseguirlo, Bootstrap utiliza *media queries* para crear puntos clave de ruptura en un sistema de cuadrícula por columnas que permite distinguir entre cuatro diseños para adaptar la aplicación web manteniendo un buen aspecto y una buena usabilidad independientemente del dispositivo de visualización (Figura 2.4). Estos puntos clave de ruptura se sitúan en 768, 922 y 1200 píxeles de resolución horizontal. Son para dispositivos extra-pequeños como *smartphones* con pantallas de baja resolución horizontal (menor que 768 píxeles), para dispositivos pequeños como *tablets* (entre 768 y 992 píxeles), para dispositivos medios (entre 992 y 1200 píxeles) y para grandes dispositivos (mayor que 1200 píxeles).

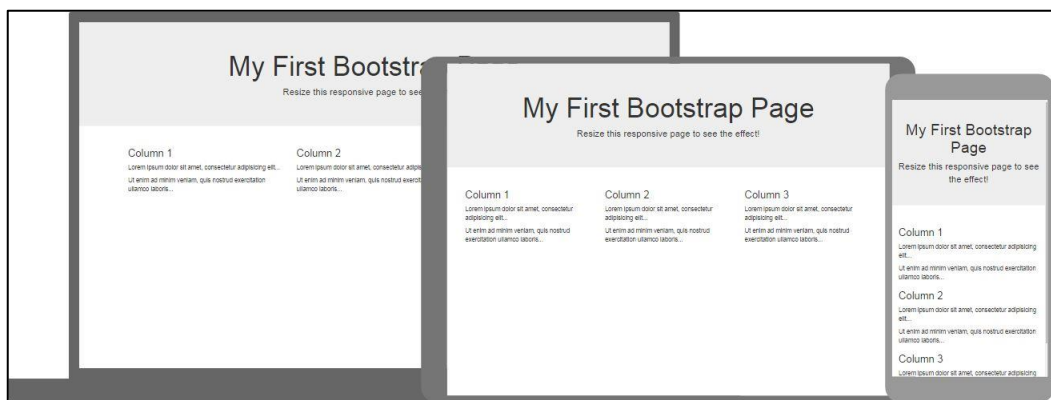


Figura 2.4. Responsive web design [26].

## 2.3.6. Otras herramientas de desarrollo utilizadas

Otras herramientas que se utilizarán como base para el desarrollo de la plataforma del profesor y que trabajarán de forma conjunta con Spring Framework son:

- Hibernate ORM [27] es una herramienta de ORM (*Object-Relational Mapping*, Mapeo Objeto-Relacional) para la plataforma Java. Esta herramienta permite acceder a la base de datos relacional y mapear los atributos del modelo de objetos y la base de datos relacional.
- c3p0 [28] es una librería Java para manejar agrupaciones de conexiones. Ofrece multitud de parámetros a configurar, tales como el número de conexiones mínimas o máximas que dispondrá la agrupación, el tamaño de la caché, los tiempos de conexiones o el número de hilos separados para operaciones.
- AspectJ [29] es una librería Java utilizada para la programación orientada a aspectos. Ha sido utilizada para crear un sistema de trazas para controlar los registros de la capa de negocio de la plataforma del profesor.
- Apache Log4j [30] es una librería Java que permite la creación de *logs*. Es muy configurable y permite tener jerarquía a nivel de componentes y distintos niveles información de lo que está sucediendo.
- SLF4J (*Simple Logging Facade for Java*) [31] es una librería Java que complementa a Apache Log4j que logra crear una capa de abstracción de *logs*.

## 2.4. Herramientas de desarrollo de la plataforma del alumno

En este apartado se describen las herramientas de trabajo que se han empleado en el desarrollo de la plataforma del alumno.

Un usuario, normalmente un alumno, puede acceder a la plataforma del alumno desde cualquier navegador web, aunque se recomienda el uso de Chrome o Safari, así como en cualquier dispositivo móvil ya que el diseño web de la plataforma del alumno también se adapta automáticamente a cualquier tamaño de pantalla. Además, también puede acceder desde la aplicación nativa en sistemas operativos móviles como Android o iOS gracias al uso de Apache Cordova.

La plataforma del alumno se emplea principalmente para visualizar contenidos generados desde la plataforma del profesor y enviar datos. Un alumno puede visualizar avisos, libros electrónicos, preguntas y notas de las diferentes asignaturas en las que está matriculado y enviar respuestas a las preguntas creadas por el

profesor Para ello, el alumno tiene que estar dado de alta en la plataforma del profesor y registrado en la plataforma del alumno.

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.

A continuación, se describen resumidamente las principales herramientas de trabajo empleadas en el desarrollo de la plataforma del alumno.

### 2.4.1. Sencha Touch

Sencha Touch [32] es un *framework* JavaScript para el desarrollo de aplicaciones web móviles multiplataforma basado en estándares web como HTML5, CSS3 y JavaScript. Además, ofrece una apariencia de las aplicaciones desarrolladas muy similar a las aplicaciones nativas en plataformas como iOS, Android, Windows Phone, BlackBerry o Tizen.

Una de las grandes ventajas es que implementa una gran cantidad de controles de interfaz para dispositivos táctiles. Entre estos eventos táctiles que soporta están: *tap* (pulsación con el dedo) *double tap* (doble pulsación con el dedo) o *swipe* (deslizar con el dedo). También incluye una serie de efectos de animación como *slide* (deslizamiento a modo de diapositiva) o *fade* (desvanecimiento de la imagen).

Otra de sus principales ventajas es su perfecto funcionamiento con Apache Cordova para poder distribuir aplicaciones de forma nativa.

Desde Sencha Touch se pueden manejar flujos de datos desde diferentes fuentes a través de AJAX (*Asynchronous JavaScript And XML*, JavaScript Asíncrono y XML), JSON o YQL (*Yahoo! Query Language*, Lenguaje de Consulta de Yahoo!). En concreto, en este proyecto se utiliza JSON como formato de intercambio de datos.

Por último, Sencha Touch también implementa el patrón MVC, como se puede ver en la Figura 2.5 definiendo una aplicación como una colección de los siguientes elementos:

- «Models»: los modelos son los objetos donde se estructuran los datos que necesita la aplicación. En ellos se encuentran la definición de campos.
- «Stores»: los almacenes se encargan de obtener los datos de fuentes internas o externas y proporcionar funciones de bajo nivel como ordenación, filtrado o consulta de registros.
- «Controllers»: los controladores se encargan de gestionar el flujo de la aplicación mediante la escucha de la interacción del usuario a través de sus eventos táctiles. Así mismo, son los encargados de contener toda la lógica de negocio.
- «Views»: las vistas se encargan de mostrar la información contenida en los almacenes y procesada por los controladores.

- «Profiles»: los perfiles se encargan de mejorar la implementación en función de las diferentes resoluciones de la interfaz de usuario para mejorar la usabilidad.

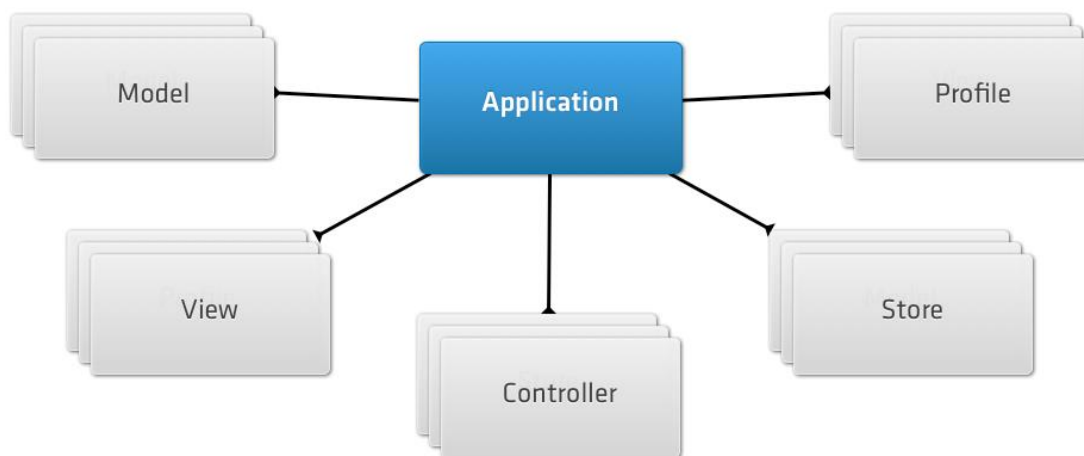


Figura 2.5. Anatomía de una aplicación en Sencha Touch [33].

Gracias a todas estas características se consigue llegar a un gran número de alumnos sin importar desde qué dispositivo o sobre qué plataforma use el alumno el entorno virtual, ya que le bastará con tener un navegador y conexión a Internet para poder hacer uso del entorno.

## 2.4.2. Apache Cordova

Apache Cordova [34] es una herramienta de código abierto para el desarrollo de aplicaciones móviles híbridas. El desarrollo bajo esta herramienta se basa en el uso de estándares web como HTML, CSS y JavaScript, permitiendo realizar desarrollo multiplataforma y por lo tanto evitando realizar el desarrollo en el lenguaje nativo de cada una de las plataformas. Además, ofrece un conjunto de API desarrolladas en JavaScript que permiten el acceso a funcionalidades nativas del dispositivo.

Combinando esta herramienta con el desarrollo realizado con Sencha Touch se obtendrá de manera rápida, sencilla y transparente las diversas aplicaciones móviles para diferentes plataformas como Android, iOS, Windows Phone, BlackBerry, Amazon Fire OS o Tizen.

El uso de esta herramienta permite empaquetar aplicaciones móviles usando los SDK (*Software Development Kit*, Kit de Desarrollo Software) de cada una de las plataformas que se requieran, consiguiendo una aplicación que puede ser distribuida por las diferentes tiendas oficiales de aplicaciones.

Gracias a todo ello, el alumnado tendrá un acceso sencillo y cómodo ya que podrá instalar y ejecutar, sin necesidad de usar el navegador, la aplicación en su móvil y poder usar el entorno virtual como cualquier otra aplicación instalada en su dispositivo.

## 2.5. Herramientas de desarrollo de la API

Para la comunicación entre la plataforma de alumno y los contenidos y preguntas generadas desde la plataforma del profesor se ha desarrollado una API. Esta API será la encargada de obtener las noticias, libros electrónicos y preguntas tipo test que están almacenados en la base de datos. Además, será capaz de obtener los vídeos del entorno virtual que están almacenados físicamente en el servidor donde está ejecutándose el entorno. Por último, también será capaz de almacenar en la base de datos las respuestas enviadas desde la plataforma del alumno para su posterior consulta por parte del profesor.

Para el desarrollo de la API se empleará Spring Framework por los mismos motivos explicados en el apartado anterior. Además, mantendrá una estructura similar a la plataforma del profesor, salvo que carecerá de interfaz web, por lo que también compartirá herramientas como Spring Data JPA, Hibernate, c3p0, AspectJ, log4j y SLF4J.

Para que la plataforma del alumno pueda interpretar los datos, la API representará los datos en formato JSON [35], formato ligero de intercambio de datos que utiliza la notificación literal de objetos JavaScript. Su principal ventaja sobre otros formatos de intercambio es que este código es analizado desde JavaScript de forma trivial, lo que supone leerlo de forma más sencilla y rápida que otros formatos.

## 2.6. Entorno de desarrollo

Para la realización de este proyecto se han necesitado varios entornos de desarrollo. En concreto Spring Tool Suite, donde se desarrollará la plataforma del profesor y la API y por otro lado Sublime Text, donde se desarrollará la plataforma del alumno.

### 2.6.1. Spring Tool Suite

Spring Tool Suite [36] es un entorno de desarrollo de código abierto basado en Eclipse personalizado para desarrollar aplicaciones en Spring. Proporciona un entorno fácil de usar para implementar, depurar, ejecutar y desplegar aplicaciones desarrolladas con Spring en sistemas como Windows, OS X o Linux.

Incluye integraciones con Git, Maven, AspectJ o Pivotal tc Server, Pivotal Cloud Foundry, Git, Maven, AspectJ, etc. Pivotal tc Server es una versión de Apache Tomcat optimizado para desarrollar con Spring, que ofrece gráficos en tiempo real de parámetros de rendimiento consiguiendo así identificar y diagnosticar problemas que pueda tener la aplicación desarrollada.

Algunas de sus características principales son:

- Analiza los archivos de configuración y muestra información detallada sobre los *beans* definidos, sus dependencias, el espacio de nombres utilizado y proporciona una visión general para ciertos estereotipos.
- Es capaz de validar si la configuración generada contiene errores previos a su despliegue.
- Ofrece refactorización en elementos de Spring.
- Ofrece autocompletado para archivos de configuración, además de ofrecer autocompletado a la hora de realizar el desarrollo.
- Ofrece un mejor soporte sobre AOP (*Aspect-Oriented Programming*, Programación Orientada a Aspectos) gracias a su integración con AspectJ.

## 2.6.2. Sublime Text

Sublime Text [37] es un editor de texto y un editor de código fuente multiplataforma creado en C++ y Python desarrollado originalmente como una extensión de Vim; aunque con el tiempo fue creando su propia identidad, por lo que mantiene un modo de edición tipo vi llamado «Vintage mode».

Su distribución se realiza de forma gratuita, aunque no es un *software* libre o de código abierto. Se puede obtener una licencia para su uso ilimitado, sin embargo, aunque no se disponga de ésta no genera ninguna limitación más que una alerta cada cierto tiempo.

Algunas de sus características principales son:

- Navegación por pestañas consiguiendo cambiar rápidamente de ficheros.
- Realiza resaltado de sintaxis además de ser configurable.
- Ofrece soporte nativo a multitud de lenguajes, además de proporcionar un autocompletado para facilitar el desarrollo.
- Contiene un mini-mapa que consiste en una pre-visualización de la estructura del código, lo cual resulta muy útil para realizar un desplazamiento rápido si se conoce la estructura del mismo.
- Permite selección múltiple con la finalidad de seleccionar un término en diferentes partes de un documento y modificar todas a la vez.
- Búsqueda dinámica a través de expresiones regulares o normales por archivos, proyectos, por directorios o una conjunción de ellos.
- Ofrece soporte de *snippets* (pequeños fragmentos de código que tienden a ser repetitivos).
- Se puede extender sus capacidades mediante el uso de *plugins*.



## 2.7. Conclusiones

En Internet existen múltiples plataformas *e-learning* y *m-learning* que tienen funcionalidades similares a las que se quieren implementar en el entorno virtual pero no cumplen con los requisitos estipulados acorde a nuestros objetivos. Algunas de estas herramientas no tienen versión de aplicación para dispositivos móviles o tabletas, lo que hace difícil su ejecución durante las clases. Otras herramientas simplemente no implementan todas las funcionalidades pensadas de forma integrada o solamente una parte de las mismas, por lo que tendrían que ser usadas en combinación de otras herramientas.

Para poder cumplir con todas las especificaciones del entorno virtual y buscar la mayor accesibilidad en cualquiera de las plataformas desde cualquier dispositivo inalámbrico, y además desde cualquier lugar y en cualquier momento, se han escogido una serie de lenguajes y herramientas que favorecen el desarrollo multiplataforma o el diseño adaptable.

El entorno virtual consta de tres partes. Por un lado, está la plataforma del profesor que hará las funciones de *back-end*. Esta plataforma será accesible desde cualquier navegador y usable gracias a su diseño adaptativo y, además, se encargará de crear y gestionar las asignaturas, sus alumnos y los contenidos de las mismas, así como un sistema de evaluación continua.

Por otro lado, está la plataforma del alumno que hará las funciones de *front-end*. Esta plataforma será accesible desde cualquier navegador y también tendrá aplicaciones nativas para diversas plataformas móviles. En ella se podrá acceder a los contenidos generados por los profesores y al sistema de evaluación continua donde poder contestar a las preguntas abiertas y ver sus notas. La comunicación necesaria para poder realizar estas tareas desde esta plataforma se llevará a cabo por medio del uso de una API, donde se recibirán las peticiones y se entregarán los contenidos solicitados.



# 3

## Entorno virtual AIM-Mobile Learning Platform

### 3.1. Introducción

En este Capítulo 3 se describe el entorno virtual de aprendizaje AIM-Mobile Learning Platform, el cual se divide en la aplicación del profesor, la del alumno, una API de comunicación y una base de datos con los contenidos. En concreto, se describirán tanto las arquitecturas *software* asociadas a cada parte de la plataforma como todas las funcionalidades que hay disponibles en cada una de ellas.

### 3.2. Descripción de la plataforma del profesor

En este apartado, en primer lugar, se describe la arquitectura *software* asociada a la plataforma del profesor siguiendo el patrón MVC que separa los datos y la lógica de negocio de la aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

Y a continuación, en segundo lugar, se describen las funcionalidades que realiza la plataforma del profesor dentro del entorno AIM-Mobile Learning Platform.

#### 3.2.1. Arquitectura *software* de la aplicación del profesor

En primer lugar, hay que indicar que el proyecto «AIMML» es el proyecto padre de los proyectos «AIMML-Admin», «AIMML-API» y «AIMML-core». Este proyecto no tiene nada de código fuente y se utiliza para centralizar las versiones que se utilizan de otras librerías de forma que, si se actualiza una versión de alguna librería como MariaDB, c3p0 o todas las herencias de Spring Framework, se actualiza simultáneamente en los tres proyectos que cuelgan de él.

Por otro lado, el proyecto «AIMML-core» es el proyecto donde se ha centralizado todo lo relacionado con la base de datos y las pequeñas utilidades que son comunes. Por tanto, los proyectos «AIMML-Admin» y «AIMML-API» necesitan de este.

En la Figura 3.1 se puede ver el árbol de directorios del proyecto «AIMML-Admin» que representa la arquitectura *software* de la plataforma del profesor.

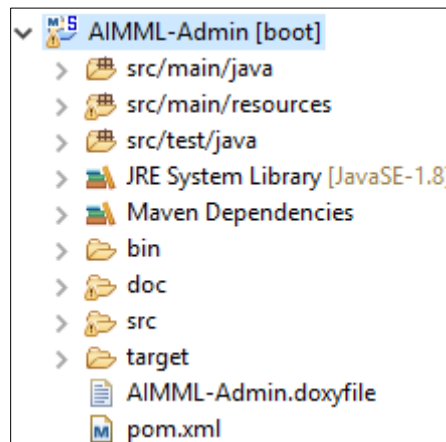


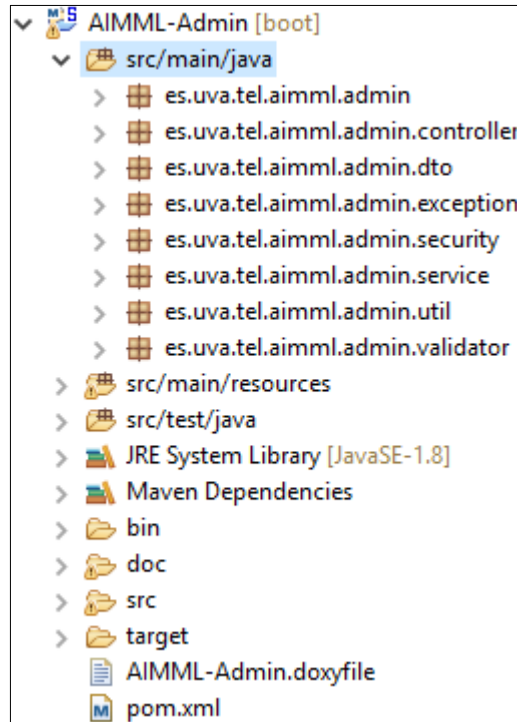
Figura 3.1. Árbol de directorios del proyecto «AIMML-Admin».

En el directorio «src/main/java» están los ficheros Java asociados a los datos y a la lógica de negocio, mientras que en el directorio «src/main/resources» están los ficheros asociados a la interfaz de usuario siguiendo el patrón MVC.

En la Figura 3.2 se muestran los paquetes Java dentro del directorio «src/main/java» que se describen a continuación:

- «es.uva.tel.aimml.admin»: en este paquete se desarrollan las clases relacionadas con el contexto de la aplicación gracias al *framework* Spring que facilita su configuración mediante el uso de anotaciones agilizando la configuración de la base de datos, la seguridad o los *beans* de la misma.
- «es.uva.tel.aimml.controller»: en este paquete, dentro de la lógica de negocio, se desarrolla la capa de control, encargada de gestionar las peticiones del usuario haciendo de intermediario entre la capa de negocio y las vistas.
- «es.uva.tel.aimml.dto»: en este paquete se desarrollan los DTO (*Data Transfer Object*, Objeto de Transferencia de Datos) para transportar los datos entre procesos independizando las entidades de la vista.
- «es.uva.tel.aimml.exception»: en este paquete se desarrollan las excepciones, mecanismo Java para la gestión de errores al ejecutar la aplicación.
- «es.uva.tel.aimml.security»: en este paquete se desarrollan las clases relacionadas con la autenticación y los roles de usuario.
- «es.uva.tel.aimml.service»: en este paquete, dentro de la lógica de negocio, se desarrolla la capa de negocio, encargada de interactuar con las entidades del modelo y llevar el control sobre cada operación con la finalidad de que se realicen de forma correcta.

- «es.uva.tel.aimml.util»: en este paquete se desarrollan simples utilidades para facilitar alguna que otra operación común más compleja.
- «es.uva.tel.aimml.validator»: en este paquete se desarrollan clases de validación de formularios para evitar errores y posibles conflictos con la base de datos.



**Figura 3.2.** Paquetes Java dentro del directorio «src/main/java» del proyecto «AIMML-Admin».

En segundo lugar, en la Figura 3.3 se muestran los directorios y ficheros dentro del directorio «src/main/resources». En este caso, los paquetes principales dentro del directorio serán los siguientes:

- «static»: directorio en el que se encuentran las hojas de estilo, fuentes, iconos, imágenes y ficheros JavaScript de la plataforma del profesor. Aquí se encuentra Bootstrap, jQuery, Summernote o Font Awesome.
- «templates»: directorio en el que se desarrollan todos los ficheros HTML junto con Thymeleaf.
- «i18n»: directorio en el que se desarrollan los ficheros de propiedades de internacionalización donde se configuran las cadenas de texto para los idiomas disponibles en la plataforma del profesor.
- «application.properties»: fichero de propiedades donde se configuran las variables de entorno de desarrollo.
- «log4j.properties»: fichero de propiedades donde se configuran los registros (*logs*) mediante la librería Log4j.

Y, por último, en la Figura 3.1 se puede ver un fichero importante, «pom.xml», en el que se configuran las dependencias o librerías del proyecto en Spring Tool Suite gracias a Maven.

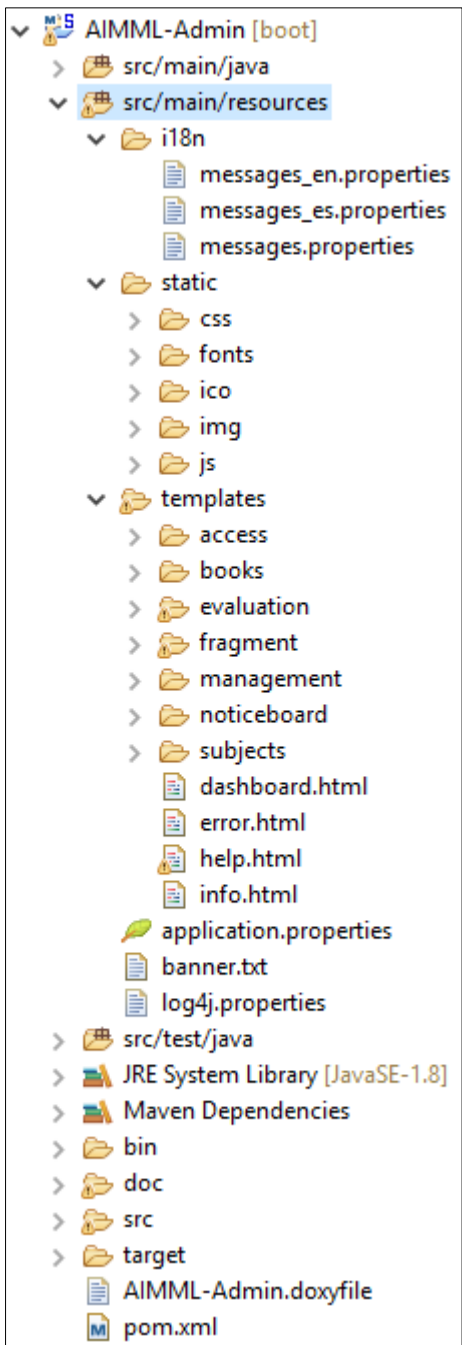


Figura 3.3. Ficheros dentro del directorio «src/main/resources» del proyecto «AIMML-Admin».

## 3.2.2. Funcionalidades de la plataforma del profesor

En los siguientes apartados se describen las diferentes funcionalidades implementadas en la plataforma del profesor, así como el controlador, el servicio y la interfaz de usuario asociados.

### 3.2.2.1. Acceso a la aplicación

Esta funcionalidad permite, por seguridad, impedir el acceso a la plataforma del profesor a usuarios no existentes en ella. Para autenticarse y poder acceder al resto de funcionalidades de la plataforma del profesor, el usuario tiene que introducir correctamente sus credenciales (nombre de usuario y contraseña) como se puede ver en la Figura 3.4.

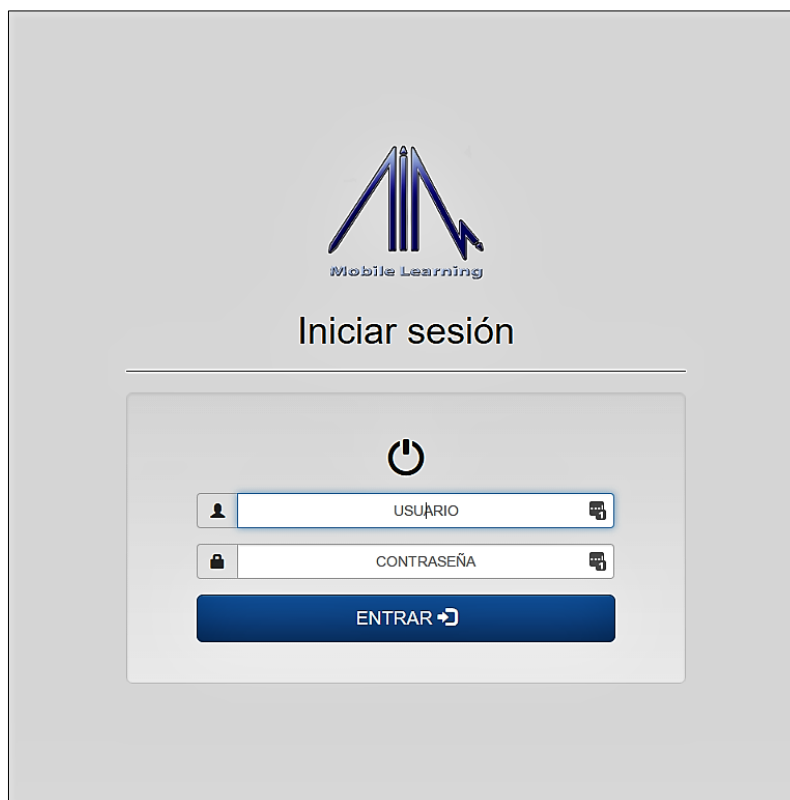


Figura 3.4. Acceso a la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «LoginController.java» se encarga de gestionar las peticiones de acceso. Además, en el paquete «es.uva.tel.aimml.admin.security», el fichero «RepositoryUserDetailsService.java» se encarga de la seguridad en la autenticación comprobando las credenciales de acceso e incluyendo el rol del usuario en la plataforma del profesor.

Los roles disponibles en la plataforma del profesor son: «regular» y «admin» que tiene acceso a la funcionalidad de administración en la plataforma del profesor. Finalmente, la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/access/login.html».

### 3.2.2.2. Menú principal

Tras acceder a la plataforma del profesor aparece un menú principal para acceder al resto de funcionalidades de la plataforma, como se puede ver en la Figura 3.5, bien a través del menú superior o bien a partir del tablero (*dashboard*) central.

Tanto en el menú superior como en el tablero central, en función del rol del usuario, se muestran o no algunos de los submenús. Es decir, si el usuario tiene un rol «admin» puede además acceder a las funciones propias de administración de la plataforma puesto que está activo y se muestra dicho submenú. Cabe mencionar que, en la esquina superior derecha del menú superior, Figura 3.5, se puede cerrar la sesión de forma segura, modificar el idioma de la plataforma del profesor (por defecto está en español, disponible en inglés) y también ver información sobre la plataforma del profesor y sobre sus diferentes funcionalidades.

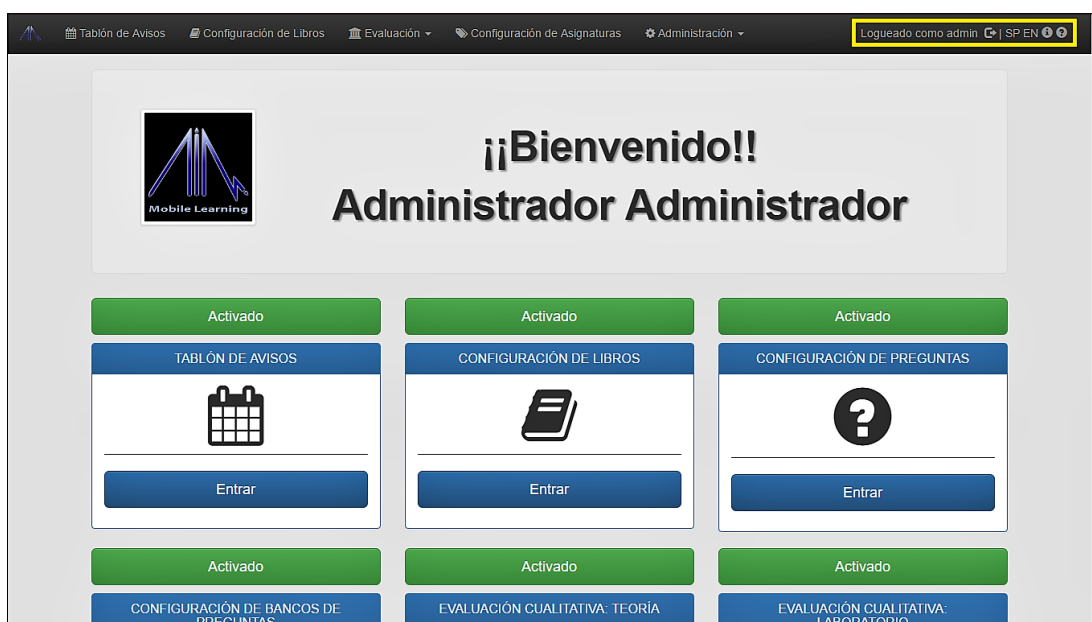


Figura 3.5. Menú principal de la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «HomeController.java» se encarga de gestionar este menú principal.

Finalmente, la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/dashboard.html».



### 3.2.2.3. Tablón de avisos

En esta sección se pueden gestionar en tiempo real los avisos de las diferentes asignaturas asociadas al usuario dentro de la plataforma del profesor. En la Figura 3.6 se pueden ver, en la parte superior, los avisos existentes («Avisos Existentes») separados por asignaturas en un sistema de pestañas y ordenados por su fecha de creación. Y junto a cada aviso, un botón de edición y otro de borrado. Finalmente, en la parte inferior, aparece un formulario de creación de un aviso («Nuevo Aviso») en el que el usuario puede editar la asignatura a la que pertenecerá, el título y la descripción del aviso.

Tras pulsar el botón de edición de un aviso, aparece, en la parte inferior, un formulario de edición para modificarlo. Este formulario contiene los mismos campos que el formulario de creación de un nuevo aviso.

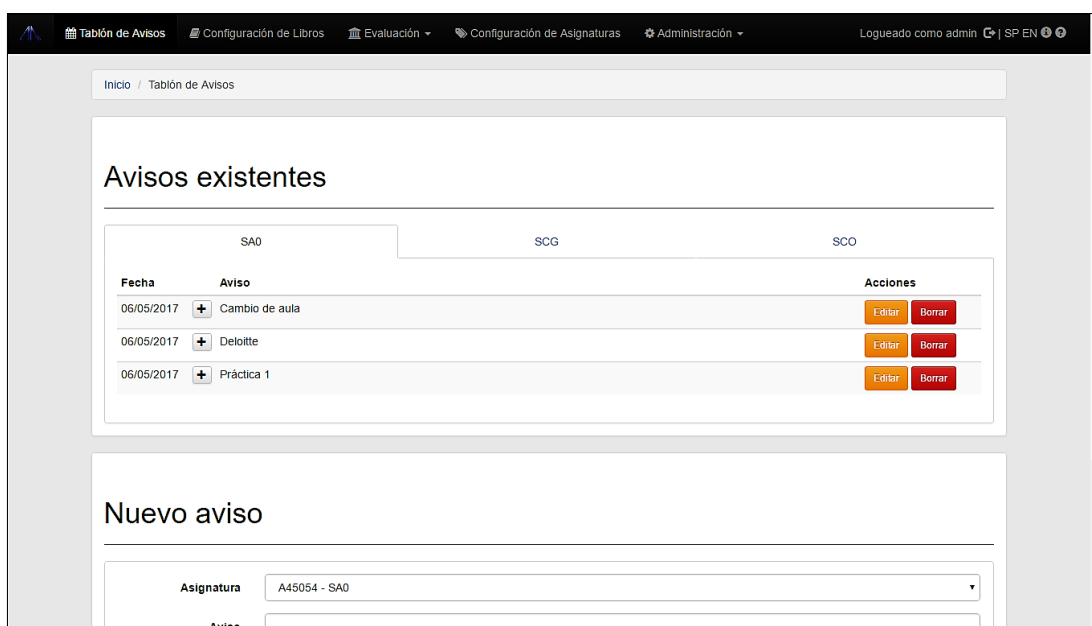


Figura 3.6. Tablón de avisos en la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «NoticeBoardController.java» se encarga de gestionar las peticiones en el tablón de avisos.

En el paquete «es.uva.tel.aimml.admin.service», el fichero «NoticeBoardService.java» se encarga de las funciones de la capa de negocio del tablón de avisos como crear, modificar o eliminar avisos.

Finalmente, la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/noticeboard/dashboard.html».

### 3.2.2.4. Configuración de libros

En esta sección se pueden gestionar los libros electrónicos de las diferentes asignaturas asociadas al usuario dentro de la plataforma del profesor. En la Figura 3.7 se pueden ver, en la parte superior, los libros electrónicos existentes separados («Libros Existentes») por asignaturas en un sistema de pestañas y ordenados por su fecha de creación. Y junto a cada libro electrónico, un botón de descarga del fichero con extensión PDF, uno de edición y otro de borrado del libro.

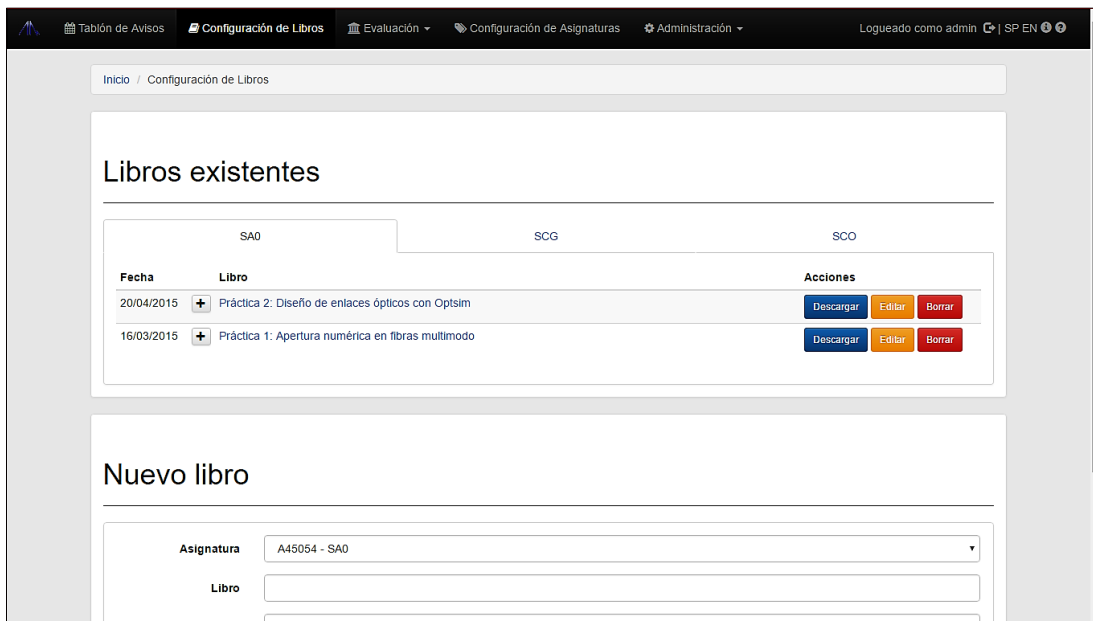


Figura 3.7. Configuración de libros en la plataforma del profesor.

Finalmente, en la parte inferior, se observa un formulario para la creación de un libro electrónico («Nuevo Libro») en el que el usuario puede editar la asignatura, el título y la descripción del libro electrónico. Tras pulsar el botón de edición de un libro electrónico (botón «Editar»), aparece, en una nueva vista, un formulario de edición para modificar el título y la descripción del mismo. Y, en la parte inferior de la nueva vista, las páginas existentes del libro electrónico («Páginas existentes»).

Junto al nombre de cada página (Figura 3.8), a la izquierda, se aprecian unos botones para modificar su orden de aparición en el libro. A la derecha de cada página, se observa un botón de edición y otro de borrado (Figura 3.8).

Cabe mencionar que bajo las páginas existentes del libro electrónico aparece un formulario de creación de una nueva página. En este formulario el usuario puede darle un título y añadir el contenido deseado, usando para ello un editor de texto, mediante el cual también se pueden insertar imágenes y vídeos.

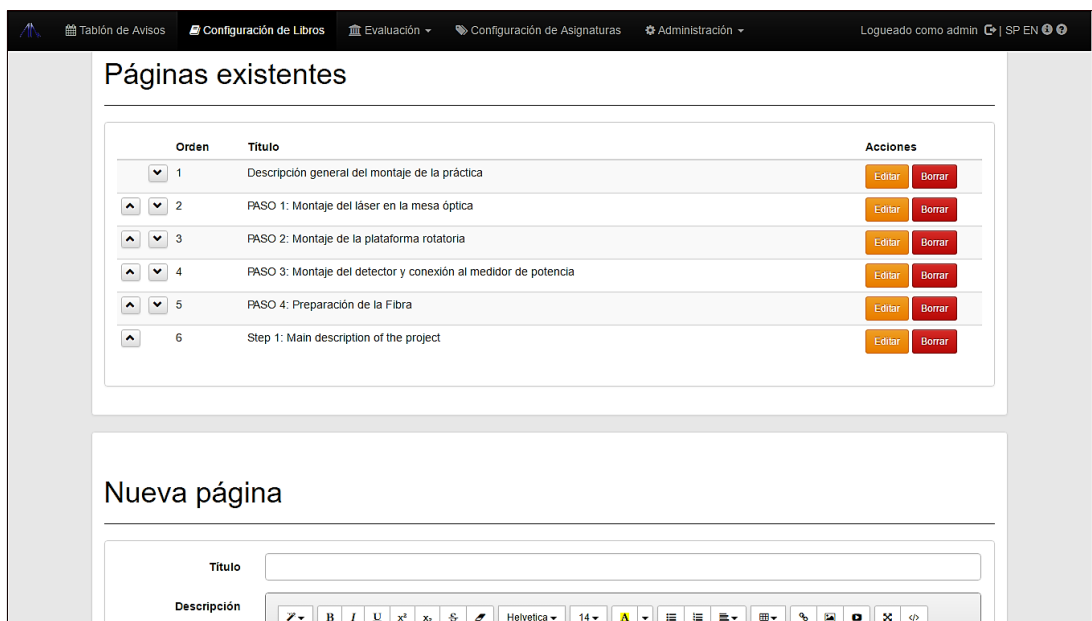


Figura 3.8. Edición de un libro electrónico en la plataforma del profesor.

Más abajo, en la página mencionada anteriormente, aparecen los vídeos existentes en la plataforma y la opción de añadir cuantos se quiera o borrar los ya existentes. La última funcionalidad existente dentro de esta funcionalidad de configuración de libros aparece bajo los vídeos del libro electrónico, en la que aparece un formulario con la opción de añadir el libro electrónico a otra asignatura.

A nivel de desarrollo *software*, en el paquete «es.uva.tel.aimml.admin.controller», el fichero «BookController.java» se encarga de gestionar las peticiones en la configuración de libros.

En el paquete «es.uva.tel.aimml.admin.service», el fichero «BookService.java» se encarga de gestionar las funciones de la capa de negocio de la configuración de libros, tales como crear, modificar o eliminar libros, páginas o vídeos.

Finalmente, la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/books/».

Cabe destacar que esta funcionalidad en la plataforma del profesor está implementada mediante un conjunto de herramientas entre las que destacan:

- Summernote [38]: editor WYSIWYG (*What You See Is What You Get*, Lo Que Ves Es Lo Que Obtienes) *online*. Está realizado como una librería JavaScript y permite ver el resultado final según se va editando. Entre otras características permite internalización, atajos de teclado, integrar imágenes dentro del contenido u online, incorporar vídeos, formatear el texto o introducir listas numeradas o de viñetas. Además, este editor ha sido modificado para permitir la incorporación de los vídeos subidos dentro del entorno virtual mediante un *plugin* llamado «summernote-ext-video.js» donde se incluye la inserción de vídeos y donde se debe realizar dicha modificación.
- Flying Saucer [39]: librería Java que, junto a las librerías iText [40] y JTidy [41], permite generar ficheros con extensión PDF desde un contenido XHTML.

- Para la subida de ficheros se han utilizado dos componentes del repositorio Apache Commons [42]: FileUpload [43] que proporciona un sencillo manejo más avanzado ficheros, y análogamente IO [44] que facilita y mejora la manipulación y lectura del contenido de ficheros.
- FFmpeg [45]: herramienta que permite convertir, grabar y transmitir audio y vídeo. Se utiliza para convertir los vídeos subidos al entorno virtual de aprendizaje. El objetivo de la conversión es obtener los videos en formatos admitidos en HTML5. Para ello, se convierte el vídeo subido en los formatos admitidos por las principales plataformas que son WebM, MP4 y OGV. Los videos convertidos son almacenados en la ruta indicada en el fichero «src/main/resources/application.properties» y catalogados según la asignatura y el libro donde han sido subidos.

## 3.2.2.5. Sistemas de evaluación continua del alumnado

### 3.2.2.5.1. Configuración de preguntas

En esta sección se pueden gestionar todas las preguntas de tipo test de las diferentes asignaturas asociadas al usuario dentro de la plataforma del profesor además de las diferentes preguntas publicadas y los resultados de las preguntas.

En la Figura 3.9 se puede ver, en la parte superior, las preguntas existentes separadas por asignaturas en un sistema de pestañas y ordenadas alfabéticamente. Y junto a cada pregunta, un botón de visualización, uno de publicación, uno de edición y otro de borrado.

Finalmente, en la parte inferior, se visualiza un formulario para la creación de una nueva pregunta en el que el profesor podrá editar la asignatura a la que va a pertenecer la pregunta, el enunciado de la misma y si desea mostrar o no las soluciones aleatoriamente desordenadas a cada alumno.

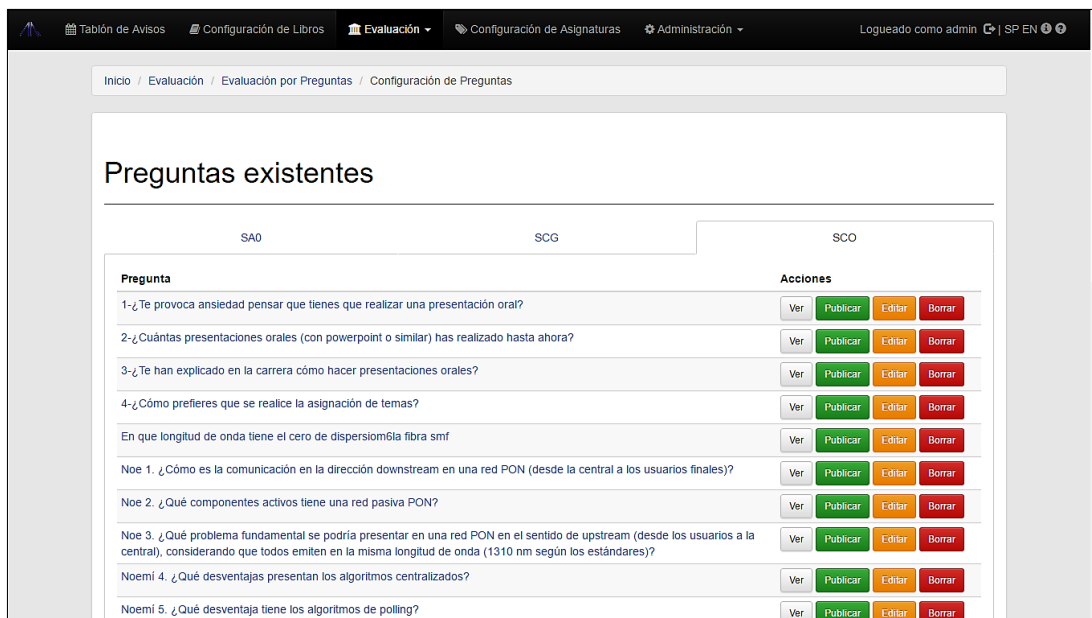


Figura 3.9. Configuración de preguntas en la plataforma del profesor.

Tras pulsar el botón de creación de una nueva pregunta o el botón de edición de una pregunta existente aparece un nuevo formulario de modificación de pregunta en el que se puede cambiar el enunciado y si mostrar o no las soluciones aleatoriamente desordenadas.

A su vez, justo debajo, aparecen las soluciones existentes junto a su valor (correcta o errónea) y un botón de edición y otro de borrado.

Bajo las soluciones existentes de la pregunta aparece un formulario de creación de una nueva solución asociada a la pregunta en el que el usuario puede elegir su valor (correcta o errónea) y el enunciado de la solución. Finalmente, bajo este formulario, aparece una opción que permite al profesor añadir la pregunta a otras asignaturas.

Por otro lado, en la pantalla principal de las preguntas, al pulsar el botón de visualización de una pregunta existente, aparece la lista de soluciones junto con su valor (correcta o errónea), así como los resultados de una publicación a elegir por el usuario y la descarga de estadísticas de las publicaciones de un curso académico concreto, tal y como se observa en la Figura 3.10. Dichas estadísticas se podrán descargar en un fichero con extensión de Microsoft Office Excel (XLS, XLSX).

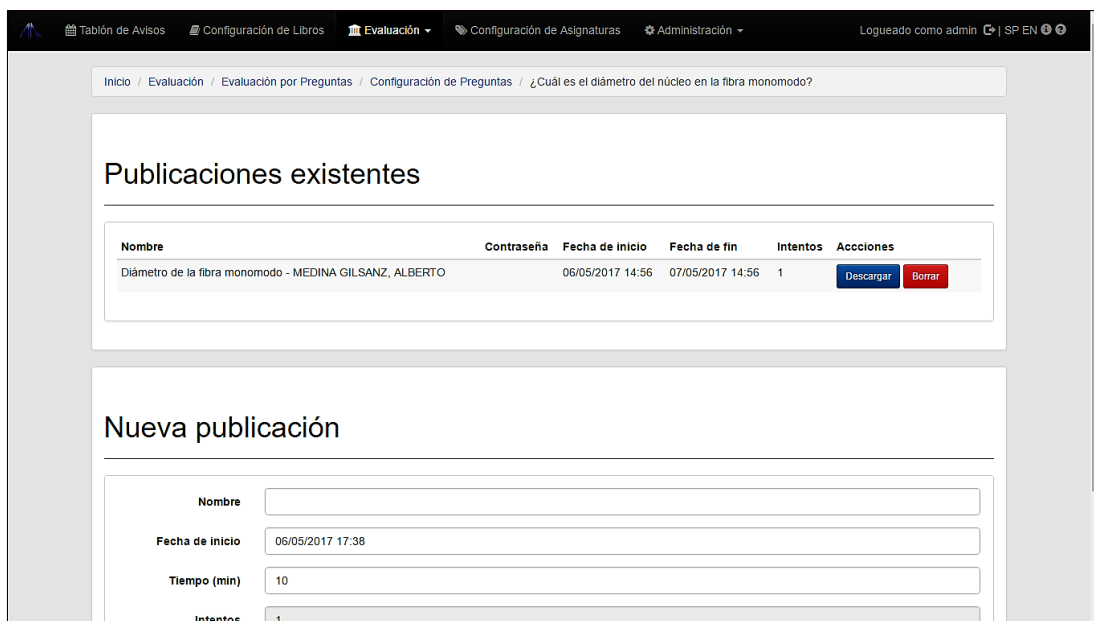


**Figura 3.10.** Visualización de una pregunta en la plataforma del profesor.

También desde la pantalla principal de las preguntas, se puede pulsar el botón de publicación de una pregunta existente, y aparece una lista de todas las publicaciones existentes a dicha pregunta.

Junto a cada publicación existente aparece un botón de descarga de resultados en un fichero con extensión de Microsoft Office Excel (XLS, XLSX) y otro de borrado, tal y como se aprecia en la parte superior de la Figura 3.11.

Finalmente cabe mencionar que bajo la lista de publicaciones existentes aparece un formulario de creación de una nueva publicación. El usuario profesor podrá dar un nombre a la publicación, una fecha y hora de activación de la publicación, una duración de dicho tiempo de activación (en minutos) y finalmente si la publicación está o no protegida por contraseña (Figura 3.11).



**Figura 3.11.** Publicación de una pregunta en la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «QuestionController.java» se encarga de gestionar las peticiones en la configuración de preguntas.

En el paquete «es.uva.tel.aimml.admin.service», el fichero «QuestionService.java» se encarga de las funciones de la capa de negocio como crear, modificar o eliminar preguntas; o crear, modificar o eliminar soluciones; o crear, modificar o eliminar publicaciones.

Finalmente, la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/evaluation/questions/individual/».

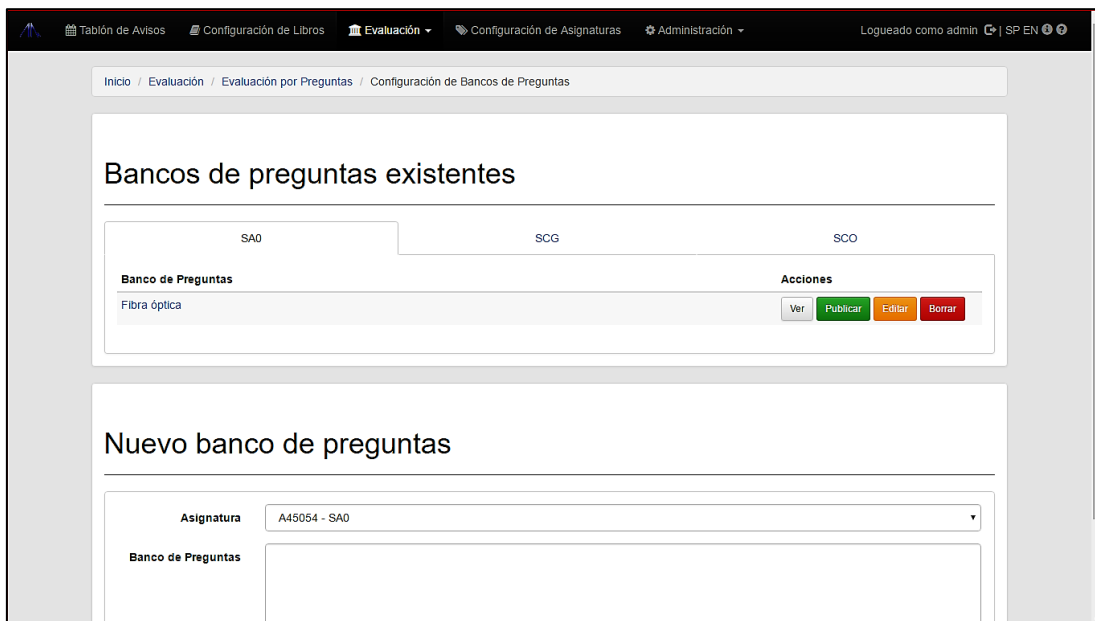
Esta funcionalidad en la plataforma del profesor está implementada mediante un conjunto de herramientas entre las que destacan:

- Morris.js [46]: librería JavaScript que permite visualizar datos en gráficos interactivos con un aspecto profesional.
- Apache POI [47]: librería Java que permite leer y escribir en ficheros con extensión de Microsoft Office Excel (XLS, XLSX) y Microsoft Office Word (DOC, DOCX).

### 3.2.2.5.2. Configuración de bancos de preguntas

En esta sección se gestionan los bancos de preguntas de las diferentes asignaturas asociadas al usuario en la plataforma del profesor. Se muestran separados por asignaturas en un sistema de pestañas y ordenados alfabéticamente. Los bancos de preguntas están pensados para que el profesor pueda enviar de forma simultánea un conjunto de preguntas a los alumnos, en vez de activarlas y enviarlas una a una. Junto a cada banco de preguntas, aparece un botón de visualización, uno de publicación, uno de edición y otro de borrado, tal y como se muestra en la parte superior de la Figura 3.12.

Bajo la lista de todos los bancos de preguntas existentes aparece un formulario de creación de un nuevo banco, donde el profesor puede introducir la asignatura y el nombre del banco de preguntas (Figura 3.12).



**Figura 3.12.** Configuración de bancos de preguntas en la plataforma del profesor.

Tras crear un banco de preguntas o tras pulsar el botón de edición aparece un nuevo formulario en el que el profesor puede editar el nombre del banco de preguntas. Además, bajo este formulario de edición se muestra la lista de preguntas existentes en la plataforma del profesor asociadas al banco de preguntas y ordenadas alfabéticamente.

Finalmente, bajo la lista de preguntas existentes se muestra un formulario que permite añadir una pregunta (de las existentes en la plataforma del profesor) al banco de preguntas.

Por otro lado, desde la pantalla principal del banco de preguntas, una vez pulsado el botón de visualización se muestran los resultados de una publicación a elegir por el usuario asociada a cada pregunta del banco de preguntas de la publicación. Además, aparece la opción de descarga de resultados (en fase de desarrollo). Finalmente, bajo los resultados se muestra el formulario de descarga de las estadísticas de un curso académico (en fase de desarrollo).

Finalmente, desde la pantalla principal y tras pulsar el botón de publicación se muestra la lista de publicaciones existentes en la plataforma del profesor asociadas al banco de preguntas. Además, bajo la lista de publicaciones existentes aparece un formulario de creación de una nueva publicación asociada a un banco de preguntas. En ella, el profesor introducirá el nombre, la fecha de inicio y la duración de la nueva publicación y si quiere o no que esté protegida por contraseña.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «BankController.java» se encarga de gestionar las peticiones en la configuración de bancos de preguntas.



En el paquete «es.uva.tel.aimml.admin.service», el fichero «BankService.java» se encarga de las funciones de la capa de negocio, tales como crear, modificar o eliminar bancos de preguntas y publicaciones.

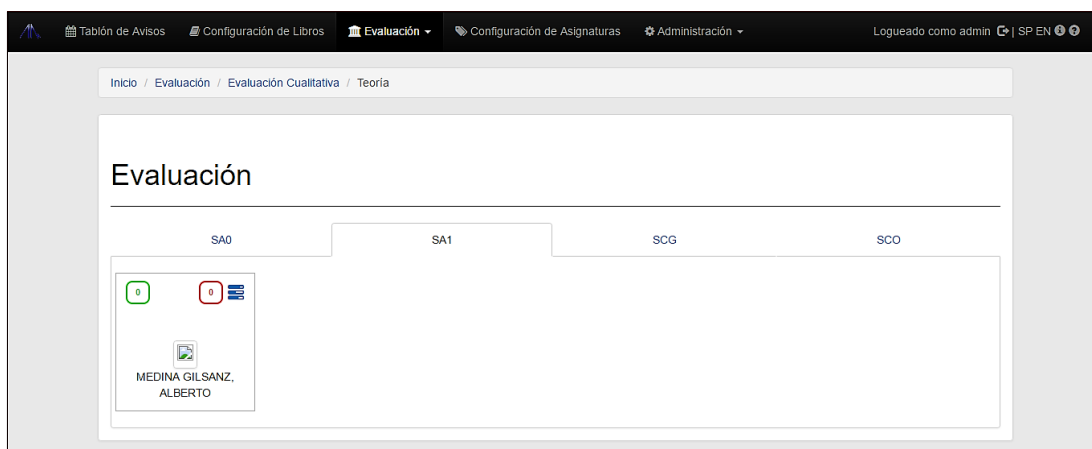
Finalmente, la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/evaluation/questions/bank/».

### 3.2.2.5.3. Sistema de evaluación continua individual basado en positivos y negativos

En esta sección el profesor podrá evaluar de forma individual a cada uno de los alumnos a partir de un sistema de positivos y negativos, en función de su nivel de implicación y participación dentro del aula. Dentro de esta sección se muestran los estudiantes de cada asignatura (en un sistema de pestañas). Cada estudiante se muestra con su fotografía, con el número de positivos (recuadro verde) y el número de negativos (recuadro rojo) asociados, como se puede ver en la Figura 3.13.

El usuario profesor puede añadir un positivo a un estudiante haciendo clic en el recuadro verde asociado y un negativo haciendo clic en el recuadro rojo. Del mismo modo, también puede quitar un positivo haciendo doble clic en su recuadro verde y un negativo haciendo doble clic en el rojo.

Además, el usuario profesor puede añadir un comentario a un estudiante haciendo clic en el icono existente junto al recuadro rojo asociado. El comentario se puede visualizar en la sección de configuración de asignaturas.



**Figura 3.13.** Evaluación individual basado en un sistema de positivos y negativos en la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «EvaluationController.java» se encarga de gestionar las peticiones en la configuración de la evaluación cualitativa de la teoría.

En el paquete «es.uva.tel.aimml.admin.service», el fichero «EvaluationService.java» se encarga de las funciones de la capa de negocio, tales como crear, modificar o eliminar positivos y negativos o escribir comentarios.

Y la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/evaluation/qualitative/theory.html».

### 3.2.2.5.4. Sistema de evaluación continua basado en positivos y negativos para grupos de laboratorio

En esta sección se muestran los estudiantes de una asignatura (en un sistema de pestañas) agrupados por grupos de laboratorio, con la finalidad de poder evaluarlos de forma grupal en función de su nivel de implicación y trabajo en las clases de laboratorio.

Además de añadir y quitar positivos y negativos a un estudiante se pueden añadir o quitar comentarios a un grupo de laboratorio, tal y como se observa en la Figura 3.14.

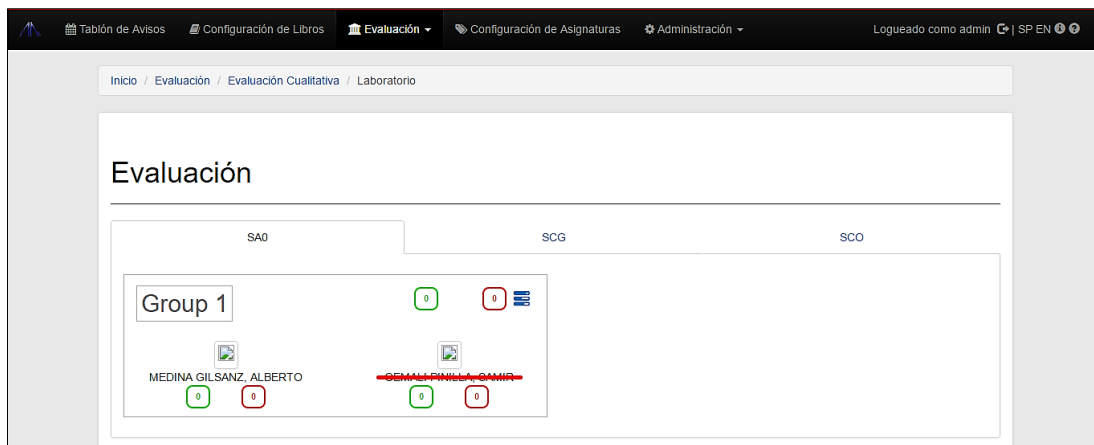


Figura 3.14. Evaluación de laboratorio en la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «EvaluationController.java» se encarga de gestionar las peticiones en la configuración de la evaluación cualitativa del laboratorio.

En el paquete «es.uva.tel.aimml.admin.service», el fichero «EvaluationService.java» se encarga de las funciones de la capa de negocio, tales como crear, modificar o eliminar positivos, negativos o comentarios.

Y la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/evaluation/qualitative/laboratory.html».

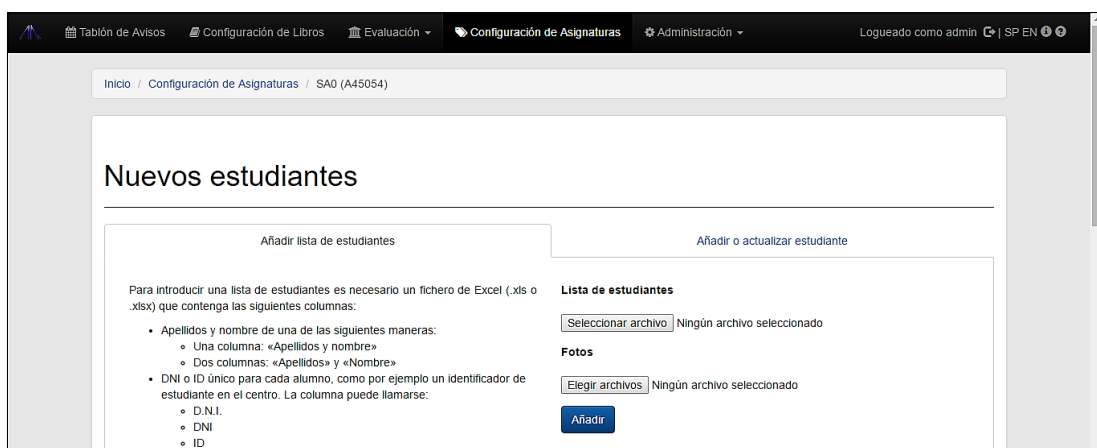
### 3.2.2.6. Configuración de asignaturas

En esta sección se pueden gestionar las diferentes asignaturas asociadas a cada profesor tal y como se muestra en la Figura 3.15.



**Figura 3.15.** Configuración de asignaturas en la plataforma del profesor.

Tras elegir una asignatura en concreto y pulsar sobre su número de código aparece una nueva vista donde se podrán añadir o actualizar estudiantes (Figura 3.16).



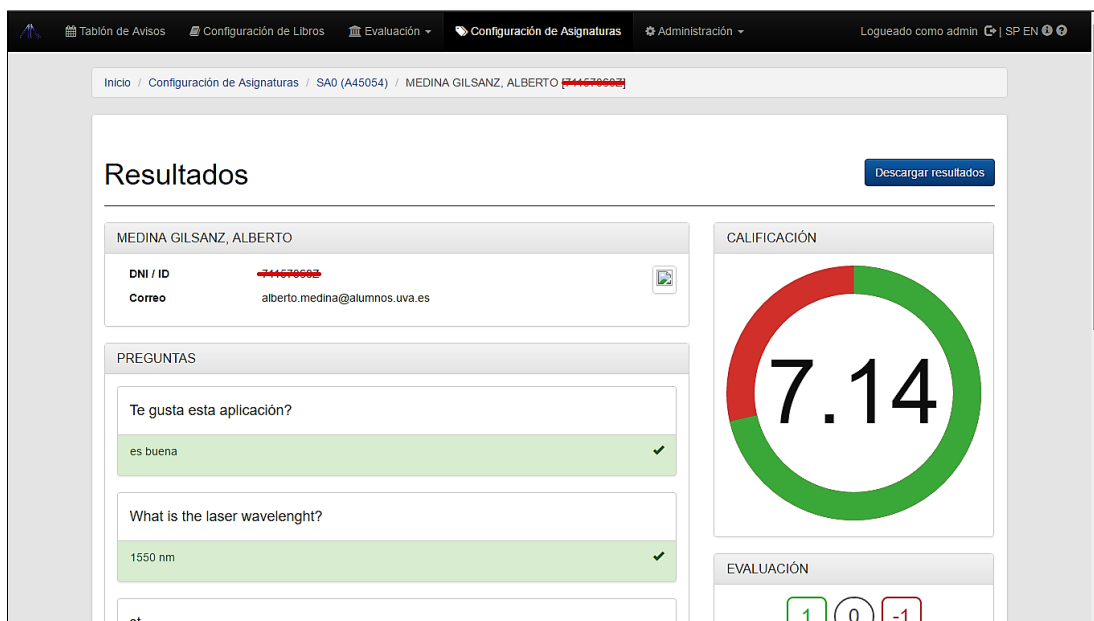
**Figura 3.16.** Creación de estudiantes en la plataforma del profesor.

La forma más común de añadir estudiantes, será subiendo un fichero con extensión de Microsoft Office Excel (XLS o XLSX)) con un formato concreto.

Por otro lado, bajo el formulario para añadir o modificar estudiantes aparece una nueva sección para crear grupos de laboratorio.

Debajo del formulario de creación de grupos de laboratorio aparece la lista de estudiantes y/o grupos de laboratorio de un curso académico a elegir, además de un botón para descargar las notas asociadas a la asignatura (fichero con extensión de Microsoft Office Excel (XLS o XLSX)).

A partir de este listado de estudiantes, se puede elegir un estudiante para visualizar sus resultados (calificaciones, preguntas, evaluaciones) además de un botón para descargar los resultados en un fichero Microsoft Office Excel (Figura 3.17).



**Figura 3.17.** Resultados de un estudiante de una asignatura en la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «SubjectController.java» se encarga de gestionar las peticiones en la configuración de asignaturas.

En el paquete «es.uva.tel.aimml.admin.service», el fichero «SubjectService.java» se encarga de las funciones de la capa de negocio como crear, modificar o eliminar asignaturas, etc.

Y la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/subjects/».

Esta funcionalidad en la plataforma del profesor está implementada mediante un conjunto de herramientas entre las que destacan:

- Circles [48]: librería JavaScript que permite generar gráficos circulares con animación.
- Componentes de Apache Commons: FileUpload e IO para la subida del fichero de estudiantes asociado a la asignatura.
- Apache POI se vuelve a utilizar para generar las notas en ficheros con extensión de Microsoft Office Excel (XLS o XLSX).

### 3.2.2.7. Tareas de administración de la plataforma

Desde esta sección se puede administrar la plataforma del profesor. El usuario tiene que tener el rol «administrador» para poder utilizar esta funcionalidad.

En esta sección se pueden administrar los usuarios, asignaturas, estudiantes, copias de seguridad y otras funcionalidades de la plataforma del profesor.

En el paquete «es.uva.tel.aimml.admin.controller», el fichero «ManagementController.java» se encarga de gestionar las peticiones en la administración de usuarios, asignaturas, estudiantes y copias de seguridad.

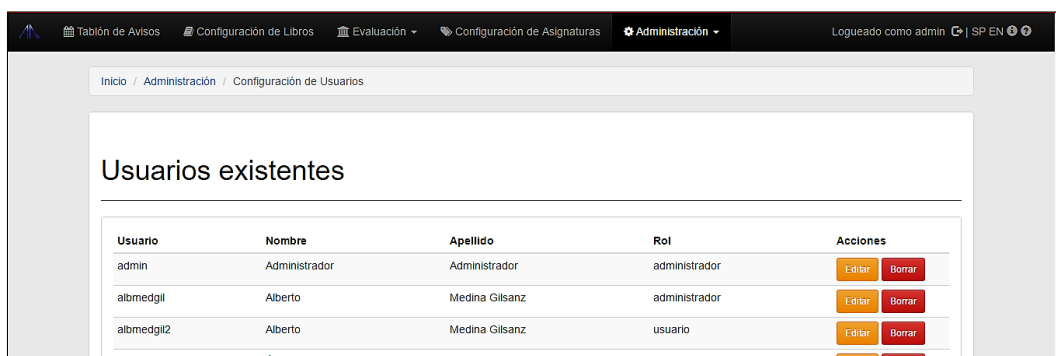
En el paquete «es.uva.tel.aimml.admin.service», el fichero «ManagementService.java» se encarga de las funciones de la capa de negocio, tales como crear, modificar o eliminar usuarios, asignaturas, estudiantes o copias de seguridad.

Y la interfaz de usuario de esta funcionalidad se encuentra en «src/main/resources/templates/management/».

#### 3.2.2.7.1. Configuración de usuarios

En esta sección aparece la lista de profesores dados de alta en la plataforma del profesor y junto a cada uno, un botón de edición y otro de borrado (Figura 3.18).

Bajo la lista de usuarios aparece una sección de creación de un usuario en la que el administrador puede introducir el nombre, el/los apellido/s, el nombre de usuario, la contraseña y el rol.



Usuario	Nombre	Apellido	Rol	Acciones
admin	Administrador	Administrador	administrador	<a href="#">Editar</a> <a href="#">Borrar</a>
albmedgil	Alberto	Medina Gilsanz	administrador	<a href="#">Editar</a> <a href="#">Borrar</a>
albmedgil2	Alberto	Medina Gilsanz	usuario	<a href="#">Editar</a> <a href="#">Borrar</a>
alvaro	Álvaro	Cabezas	administrador	<a href="#">Editar</a> <a href="#">Borrar</a>

Figura 3.18. Configuración de usuarios en la plataforma del profesor.

Tras pulsar el botón de edición de un usuario dado de alta en la plataforma, bajo la lista de usuarios aparece una sección de edición del usuario en el que el administrador puede editar todos los datos del usuario.

### 3.2.2.7.2. Configuración de asignaturas

En esta sección aparece la lista de asignaturas creadas y junto a cada asignatura un botón de edición y otro de borrado, tal y como se muestra en la Figura 3.19.



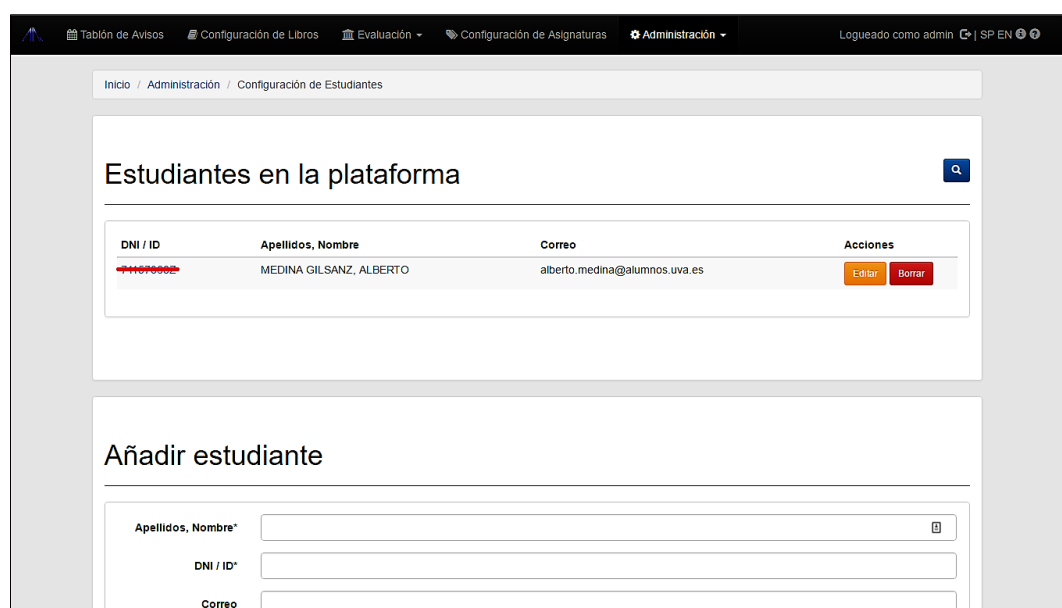
Código	Siglas	Nombre	Acciones
A46626	CO	Comunicaciones Ópticas	Editar Borrar
40470	CSE	Curriculum y Sistema Educativo	Editar Borrar
40164	DG	Didáctica General	Editar Borrar
51316	DSCO	Diseño de Comunicaciones Ópticas	Editar Borrar
42522	GRAI	Gestión del ruido ambiental y la industria	Editar Borrar
45673	OC	Optical Communications	Editar Borrar

Figura 3.19. Configuración de asignaturas en la plataforma del profesor.

Bajo la lista de asignaturas aparece una sección para crear una asignatura en la que el administrador puede introducir el código de la asignatura, las siglas asignadas y el nombre completo de la nueva asignatura. Tras pulsar el botón de edición de una asignatura en la plataforma del profesor, aparece un formulario de edición en el que el administrador puede editar el código, las siglas y el nombre. Bajo el formulario de edición aparece la lista de profesores asociados a ella. Y bajo esta lista, también se visualiza un formulario en el que el administrador puede añadir a un nuevo profesor asociado a la asignatura.

### 3.2.2.7.3. Configuración de estudiantes

En esta sección aparece la lista de estudiantes dados de alta en la plataforma del profesor y junto a cada estudiante un botón de edición y otro de borrado (Figura 3.20).



DNI / ID	Apellidos, Nombre	Correo	Acciones
<del>XXXXXXXXXX</del>	MEDINA GILSANZ, ALBERTO	alberto.medina@alumnos.uva.es	Editar Borrar

**Añadir estudiante**

Apellidos, Nombre\*

DNI / ID\*

Correo

Figura 3.20. Configuración y creación de nuevos estudiantes dados de alta en la plataforma del profesor.

Bajo la lista de estudiantes se visualiza un formulario de creación de un nuevo estudiante en la plataforma del profesor (Figura 3.20) en el que el administrador puede introducir el/los apellido/s y el nombre (obligatorio), el DNI (Documento Nacional de identidad) o el identificador (obligatorio), el correo electrónico y una fotografía del nuevo estudiante.

Tras pulsar el botón de edición de un estudiante en la plataforma del profesor, bajo la lista de estudiantes aparece un formulario de edición del estudiante. En dicho formulario el profesor puede introducir el/los apellido/s y el nombre (obligatorio), el DNI o identificador (obligatorio), el correo electrónico y una fotografía del nuevo estudiante.

Y tras elegir un estudiante en la lista de estudiantes, haciendo clic en su DNI, se visualiza un formulario en el que el administrador puede elegir una asignatura a la que añadir al estudiante, así como visualizar la lista de asignaturas asociadas al estudiante e información sobre él.

Hay que destacar que el aspa al lado del nombre y los apellidos del estudiante (Figura 3.21) indica que este se ha registrado en la aplicación del alumno (respecto a la cruz que indica lo contrario).

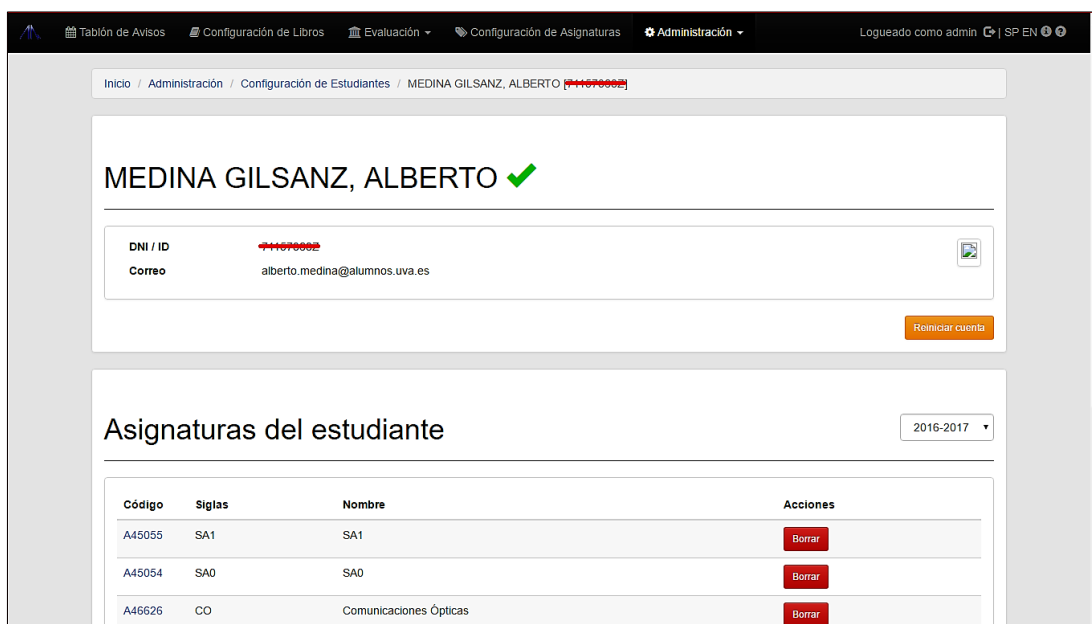


Figura 3.21. Detalle de un estudiante en la plataforma del profesor.

También se puede observar en dicha figura que existe un botón de reinicio de cuenta («Reiniciar cuenta») en la que el profesor podrá borrar el registro realizado por el alumno. Esta funcionalidad resulta de gran interés, puesto que esta versión anterior, el alumno no es capaz de recuperar su contraseña de acceso a la aplicación, por lo que el profesor puede resetear su contraseña pulsando dicho botón. En este caso, el alumno podrá volver a realizar de nuevo el registro y configurar una nueva contraseña en el sistema.

### 3.2.2.7.4. Copias de seguridad

En esta sección aparece la lista de copias de seguridad de la base de datos existentes en el entorno virtual y que son creadas a diario. Dichas copias de seguridad aparecen ordenadas por su fecha de creación, tal y como muestra la Figura 3.22.

Junto a cada copia de seguridad aparece un botón para descargar la copia de seguridad (con extensión SQL), otro para restaurar la base de datos a partir de la copia de seguridad y otro para borrar la copia de seguridad (Figura 3.22). Hay que destacar también el botón de respaldo («Respaldo ahora») que permite crear de forma automática una copia de seguridad de la base de datos.



Figura 3.22. Copias de seguridad de la plataforma del profesor.

### 3.2.2.7.5. Actuator

En esta sección aparecen otras funcionalidades para visualizar registros sobre la plataforma del profesor para administrarla mejor (Figura 3.23). Actualmente se encuentra en fase de desarrollo.



Figura 3.23. Otras funcionalidades de administración en la plataforma del profesor.



## 3.3. Descripción de la plataforma del alumno

En este apartado, en primer lugar, se describe la arquitectura *software* asociada a la plataforma del alumno siguiendo el patrón MVC que separa los datos y la lógica de negocio de la aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.

A continuación, se describen las funcionalidades implementadas en la aplicación del alumno en el entorno virtual de aprendizaje.

### 3.3.1. Arquitectura *software* de la aplicación del alumno

En la Figura 3.24 se puede ver el árbol de directorios que representa la arquitectura *software* de la aplicación del alumno. En la aplicación del alumno hay que destacar los siguientes directorios y ficheros:

- «app»: directorio en el que se almacena la aplicación desarrollada en Sencha Touch siguiendo el patrón MVC. En este directorio hay que destacar los siguientes subdirectorios:
  - «model»: directorio que contiene los ficheros asociados con el modelo de datos.
  - «store»: directorio que contiene los ficheros asociados con los modelos para proporcionar funciones de ordenación, filtrado y consulta de registros.
  - «view»: directorio que contiene los ficheros asociados con la presentación del modelo de datos asociados a la interfaz de usuario.
  - «controller»: directorio que contiene los ficheros asociados con la lógica de la aplicación.
- «build»: directorio en el que se almacenan las diferentes compilaciones generadas mediante Sencha Cmd.
- «cordova»: directorio en el que se almacenan los directorios y ficheros relacionados con Cordova.
- «resources»: directorio que contiene subdirectorios con ficheros CSS, Sass y otros activos de la aplicación como iconos e imágenes.
- «touch»: directorio que contiene una copia del SDK de Sencha Touch con todos los ficheros del *framework* necesarios para la aplicación.
- «app.js»: fichero que contiene el punto de inicio de la aplicación siguiendo el patrón MVC.
- «app.json»: fichero que contiene ajustes e instrucciones para construir y distribuir la aplicación.

- «index.html»: fichero de inicio de la aplicación que contiene las referencias al *framework* de Sencha Touch, el punto de inicio JavaScript («app.js») y las hojas de estilo.

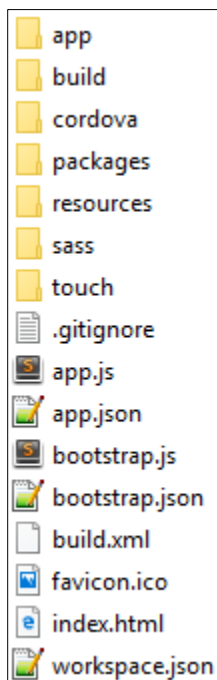


Figura 3.24. Árbol de directorios de la plataforma del alumno.

## 3.3.2. Funcionalidades de la aplicación del alumno

En los siguientes apartados se describen las diferentes funcionalidades implementadas en la aplicación del alumno siguiendo el patrón MVC, así como el controlador, el servicio y la interfaz de usuario asociados.

### 3.3.2.1. Registro en la aplicación

Esta funcionalidad permite, por seguridad, impedir el acceso a la aplicación a usuarios no registrados. El usuario alumno tiene que ser añadido en la aplicación del profesor (más concretamente en la base de datos) previamente para poder registrarse y posteriormente acceder al resto de funcionalidades del alumno.

Para registrarse en la plataforma del alumno, el usuario tiene que introducir su nombre de usuario (DNI o identificador) y una contraseña (doblemente validada), como se puede ver en la Figura 3.25 (versión web) y en la Figura 3.26 (versión Android).



The screenshot shows the registration form in the web version. At the top is the 'Mobile Learning' logo, which consists of three stylized vertical bars of varying heights and a blue arrow pointing downwards to the right. Below the logo, the word 'Mobile Learning' is written in a blue, sans-serif font. The registration form is contained within a light gray rounded rectangle. It has a title 'Registro' in blue. There are three input fields: 'Usuario', 'Contraseña', and 'Repetir Contraseña'. At the bottom of the form is a green button with the text 'registrarse' in white.

**Figura 3.25.** Registro en la aplicación del alumno (versión web).



The screenshot shows the registration form in the Android version. It features the same 'Mobile Learning' logo and text as the web version. The form is titled 'REGISTRO' in blue. It contains three input fields: 'Usuario', 'Contraseña', and 'Repetir Contraseña'. At the bottom is a green button with the text 'Registrarse' in white.

**Figura 3.26.** Registro en la aplicación del alumno (versión Android).

El fichero «Login.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Mientras, el fichero «Register.js» en la ruta «app/view/» se encarga de la interfaz de usuario de esta funcionalidad.

### 3.3.2.2. Acceso a la aplicación

Esta funcionalidad permite, por seguridad, impedir el acceso a la aplicación a usuarios no registrados en ella previamente. Para acceder a las funcionalidades de la aplicación del alumno, el usuario tiene que autenticarse introduciendo correctamente sus credenciales (nombre de usuario y contraseña creada).

El usuario alumno tiene que ser añadido en la aplicación del profesor (en la base de datos) y haberse registrado en la aplicación del alumno para acceder a las funcionalidades de la aplicación.

Además, el usuario tiene la opción de guardar sus credenciales de acceso durante su sesión activando la casilla de verificación, como muestra la Figura 3.27 (versión web) y Figura 3.28 (versión Android).

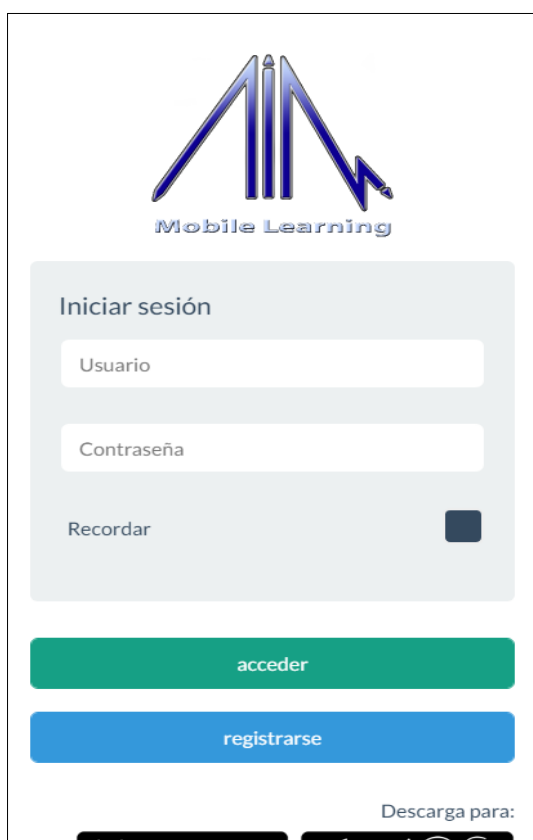


Figura 3.27. Acceso a la aplicación del alumno (versión web).

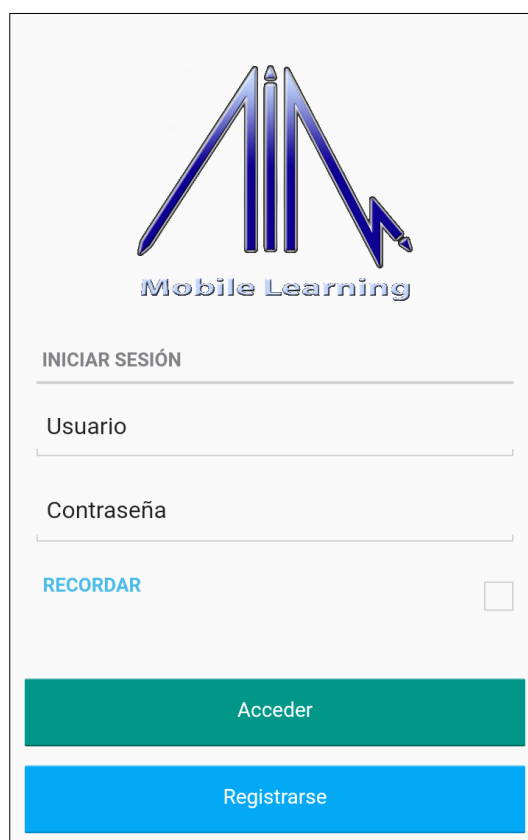


Figura 3.28. Acceso a la aplicación del alumno (versión Android).

El fichero «Login.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Además de cargar el resto de vistas, modelos y almacenes asociados en la aplicación.

Finalmente, el fichero «Login.js» en la ruta «app/view/» se encarga de la interfaz de usuario de esta funcionalidad.

### 3.3.2.3. Menú principal de la aplicación del alumno

Esta funcionalidad permite navegar al usuario de forma rápida y sencilla a las funcionalidades de la aplicación del alumno, tal y como se muestra en la Figura 3.29 (versión web) y en la Figura 3.30 (versión Android). En la barra de navegación superior se muestra, de izquierda a derecha, un botón para acceder a un pequeño manual de ayuda del alumno, un título de sección y un botón para cerrar la sesión de forma segura.

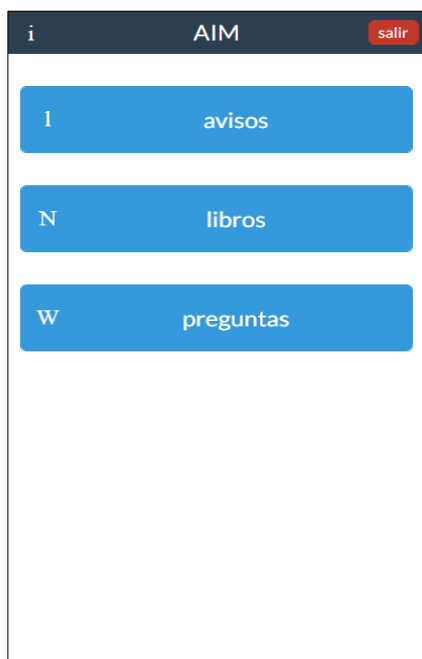


Figura 3.29. Menú principal en la aplicación del alumno (versión web).



Figura 3.30. Menú principal en la aplicación del alumno (versión Android).

El fichero «Main.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Por otro lado, el fichero «MainMenu.js» en la ruta «app/view/» se encarga de la interfaz de usuario de esta funcionalidad.

### 3.3.2.4. Tablón de avisos en la aplicación del alumno

Esta funcionalidad permite visualizar los diferentes avisos, cronológicamente ordenados, de las diferentes asignaturas (asociadas al usuario alumno) separadas mediante un sistema de pestañas, como se puede ver en la Figura 3.31 (versión web) y en la Figura 3.32 (versión Android). Cuando el alumno pulsa en un aviso concreto se muestra una descripción del mismo. Tal y como se puede apreciar en la Figura 3.33 (versión web) y en la Figura 3.34 (versión Android), se visualiza la fecha de creación del aviso, así como el nombre y una descripción más profunda del mismo.

El fichero «Notices.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Y los ficheros «Notices.js» y «Notice.js» en la ruta «app/view/» se encargan de la interfaz de usuario de esta funcionalidad.

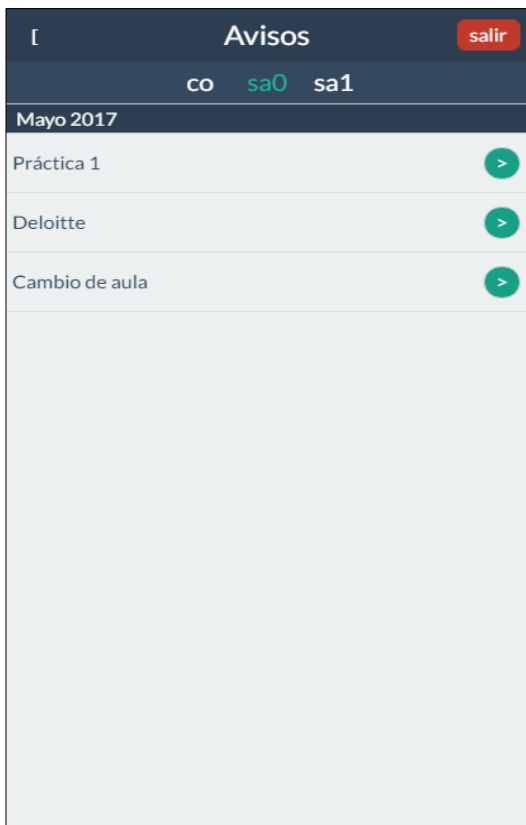


Figura 3.31. Tablón de avisos en la aplicación del alumno (versión web).

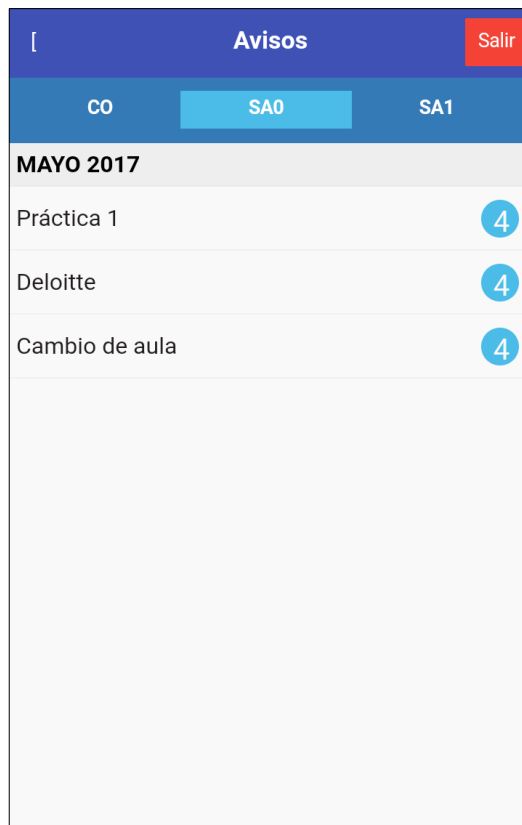


Figura 3.32. Tablón de avisos en la aplicación del alumno (versión Android).

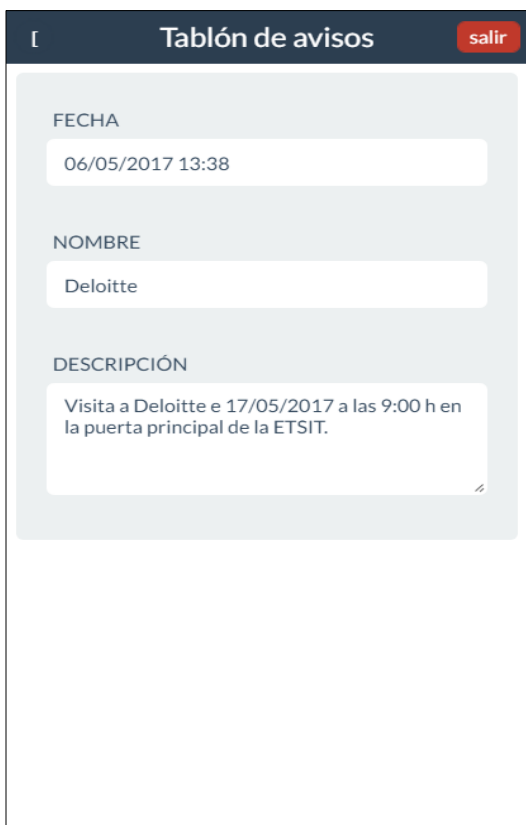


Figura 3.33. Detalle de un aviso en la plataforma del alumno (versión web).

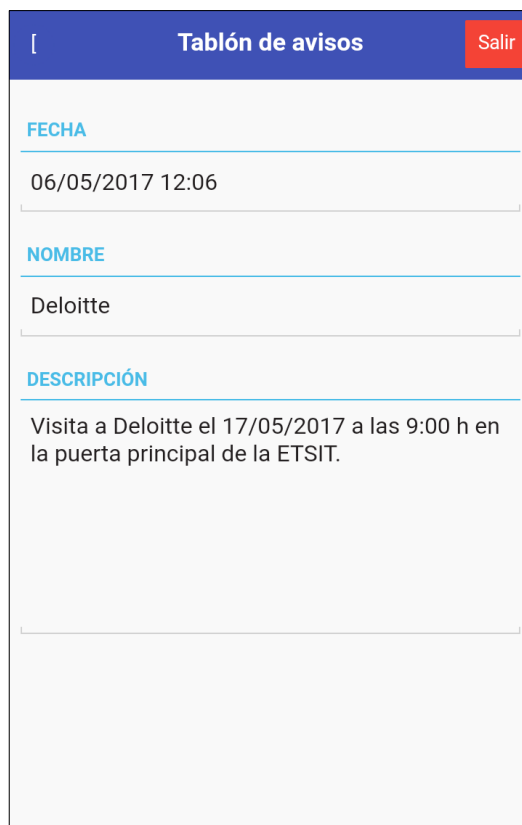
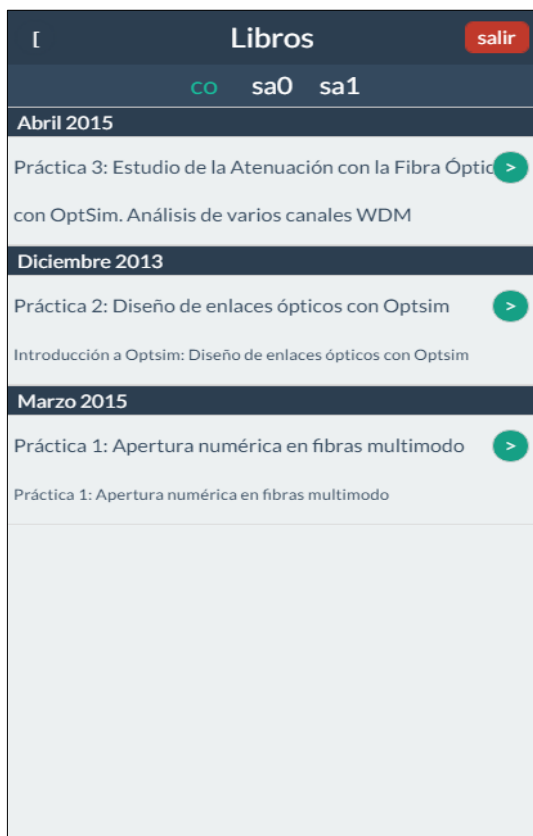


Figura 3.34. Detalle de un aviso en la plataforma del alumno (versión Android).

### 3.3.2.5. Libros electrónicos en la aplicación del alumno

Esta funcionalidad permite visualizar los diferentes libros electrónicos, cronológicamente ordenados, de las diferentes asignaturas en las que está matriculado el alumno.

Estas asignaturas están separadas mediante un sistema de pestañas, tal como se puede ver en la Figura 3.35 (versión web) y en la Figura 3.36 (versión Android). Tal y como se observan en ambas gráficas, en la vista principal se muestra el nombre principal de los libros electrónicos.



**Figura 3.35.** Libros electrónicos en la aplicación del alumno (versión web).



**Figura 3.36.** Libros electrónicos en la aplicación del alumno (versión Android).

Una vez que se pulsa en un libro electrónico aparece una nueva vista, donde se muestra una lista de las diferentes páginas asociadas al libro electrónico, es decir, un índice con todos los contenidos del libro electrónico, como se observa en la Figura 3.37 (versión web) y en la Figura 3.38 (versión Android).

A continuación, tras pulsar dentro de una página del libro electrónico, se muestra en una nueva vista todos los contenidos de la misma, tal y como se muestra en la Figura 3.39 (versión web) y en la Figura 3.40 (versión Android).

El fichero «Books.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Y finalmente, los ficheros «Books.js», «Book.js» y «Page.js» en la ruta «app/view/» se encargan de la interfaz de usuario de esta funcionalidad.

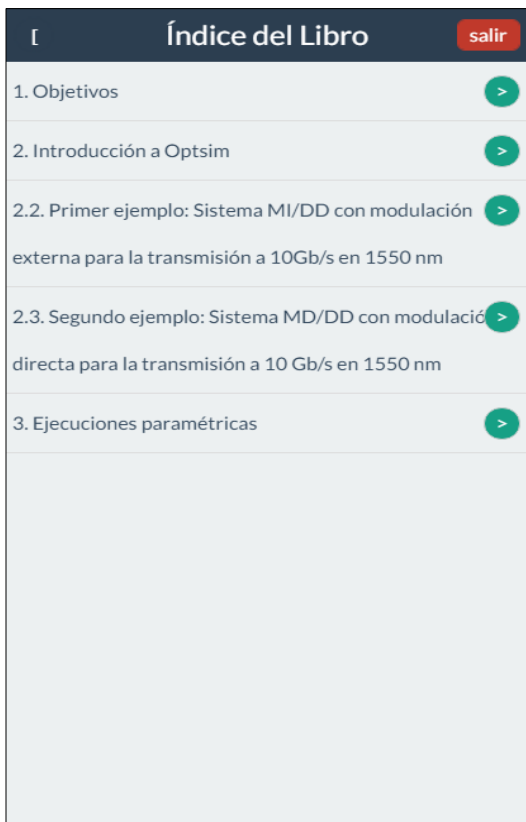


Figura 3.37. Páginas de un libro electrónico en la aplicación del alumno (versión web).

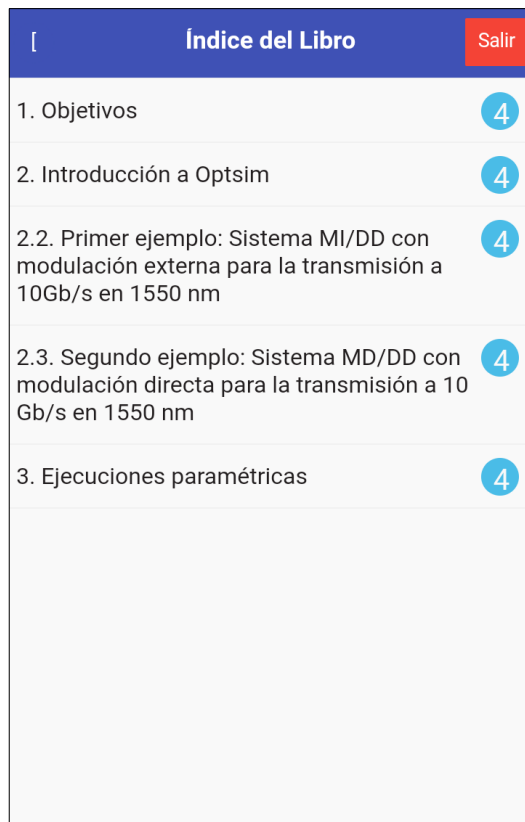


Figura 3.38. Páginas de un libro electrónico en la aplicación del alumno (versión Android).

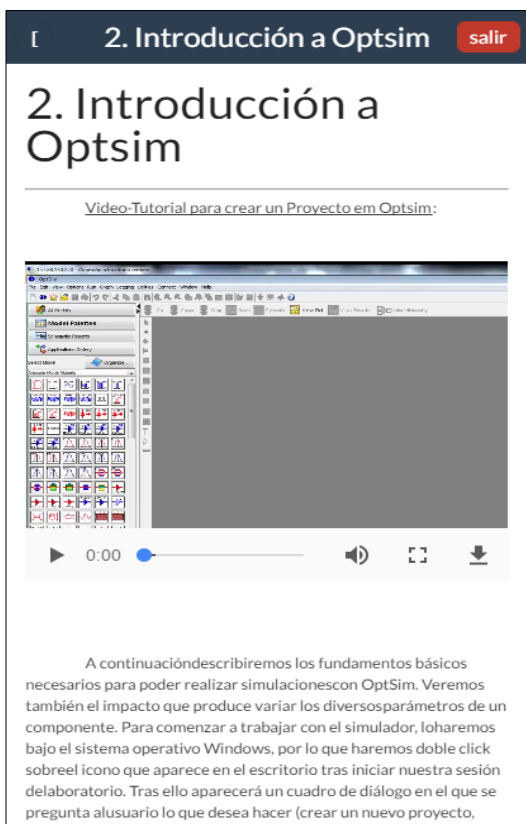


Figura 3.39. Página de un libro electrónico en la aplicación del alumno (versión web).

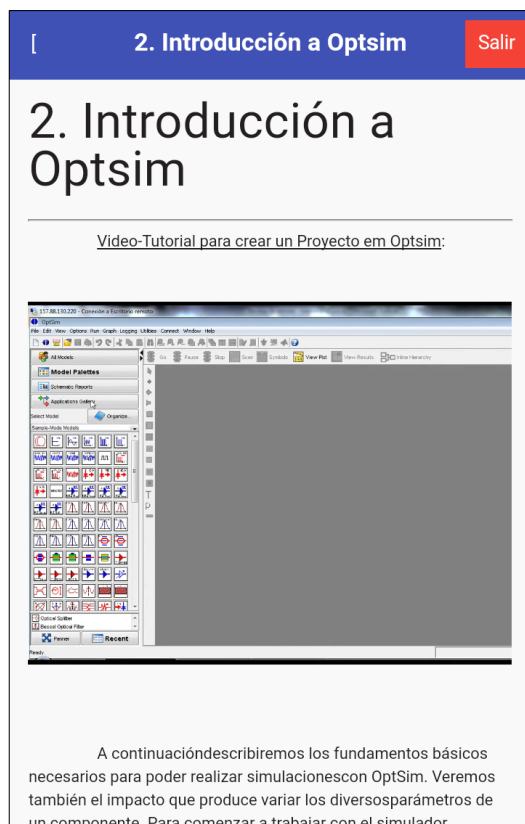


Figura 3.40. Página de un libro electrónico en la aplicación del alumno (versión Android).



### 3.3.2.6. Evaluación continua en la aplicación del alumno

Esta funcionalidad permite visualizar las diferentes preguntas publicadas y activas en ese momento, cronológicamente ordenadas, de las diferentes asignaturas (asociadas al usuario). Estas preguntas están separadas mediante un sistema de pestañas, tal y como se observa en la Figura 3.41 (versión web) y en la Figura 3.42 (versión Android). Además, bajo el enunciado de una pregunta se muestra el tiempo durante el que se puede responder.

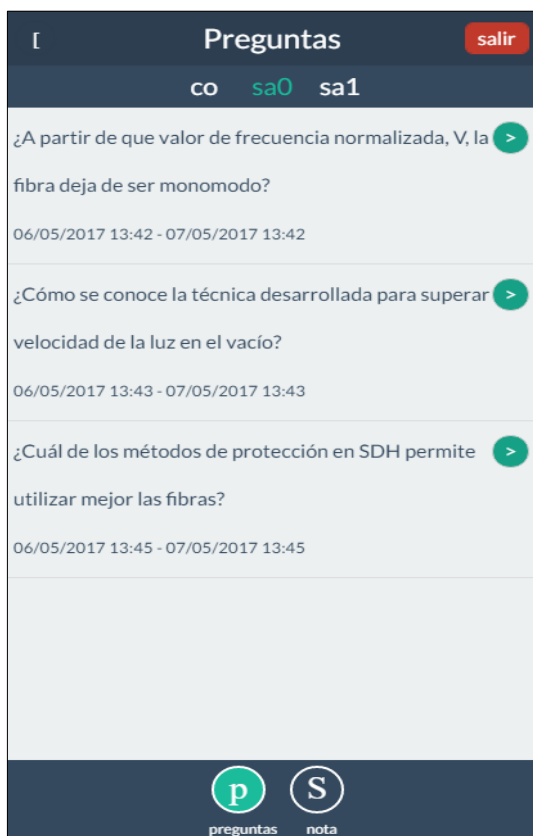


Figura 3.41. Preguntas publicadas en la aplicación del alumno (versión web).

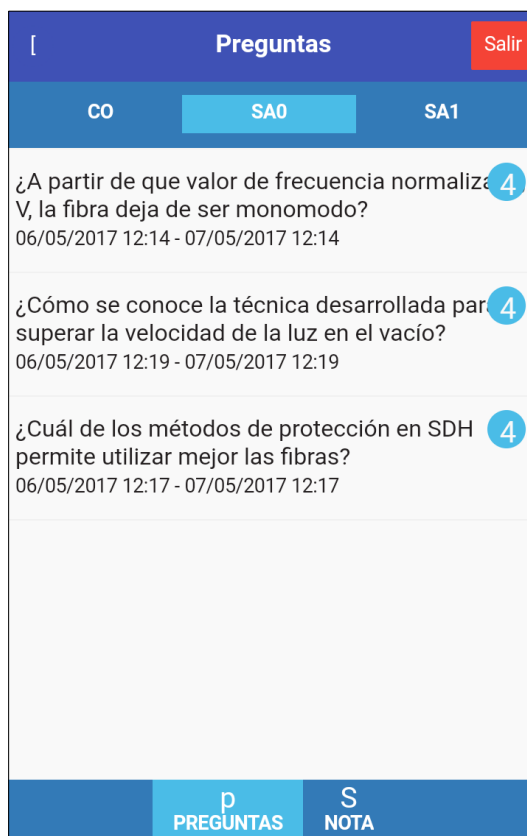


Figura 3.42. Preguntas publicadas en la aplicación del alumno (versión Android).

Tras pulsar en una de las preguntas publicadas (activadas), en ocasiones, se muestra una nueva vista en el que el usuario tiene que introducir una contraseña que el profesor da al alumno, siempre que el profesor active esta opción en su aplicación. En caso de no estar activada esta opción, no aparecerá dicha vista. Así pues, una vez introducida correctamente la contraseña (o tras pulsar en una pregunta) se muestra el enunciado de la pregunta, sus posibles soluciones y el tiempo de publicación, tal y como se puede ver en la Figura 3.43 (versión web) y en la Figura 3.44 (versión Android).

El alumno solo puede seleccionar una posible solución (correcta o incorrecta) y enviar la respuesta para su evaluación en tiempo real. Si el alumno no envía la respuesta durante el tiempo de activación de la pregunta, se considera la pregunta sin responder (solución incorrecta).



Figura 3.43. Pregunta publicada en la aplicación del alumno (versión web).

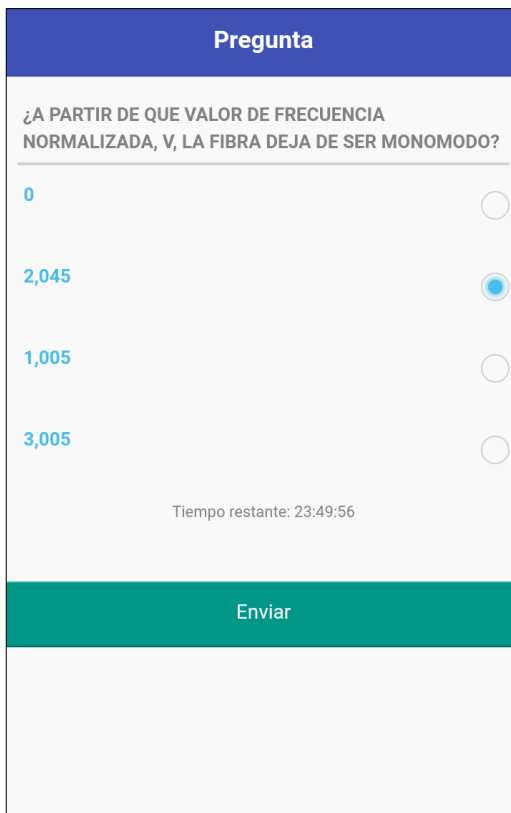


Figura 3.44. Pregunta publicada en la aplicación del alumno (versión Android).

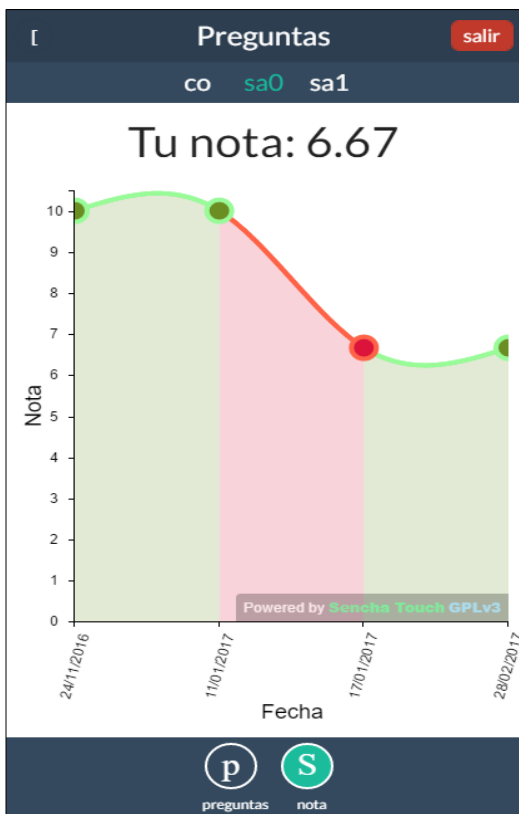


Figura 3.45. Evolución de las notas en la plataforma del alumno (versión web).

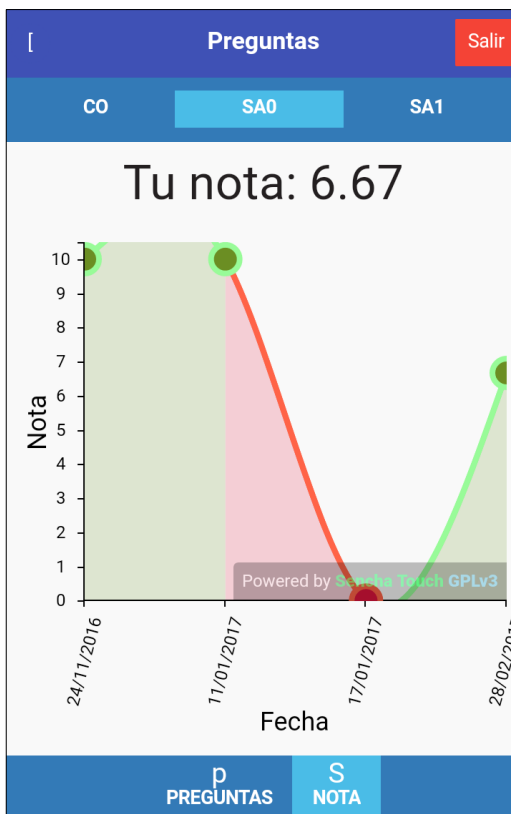


Figura 3.46. Evolución de las notas en la plataforma del alumno (versión Android).

Esta funcionalidad también permite visualizar gráficamente las notas de las diferentes asignaturas asociadas al alumno, separadas mediante un sistema de pestañas (Figura 3.45 (versión web) y Figura 3.46 (versión Android)).

Para cada asignatura, en la parte superior de la vista se muestra la nota media acumulada hasta el día actual y en la parte inferior se puede ver un gráfico con las notas medias obtenidas al finalizar un día. En ese día, el alumno podrá haber respondido una o varias preguntas. Finalmente, el fichero «Questions.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Los ficheros «Questions.js», «QuestionPassword.js» y «Page.js» en la ruta «app/view/» se encargan de la interfaz de usuario de esta funcionalidad.

### 3.3.2.7. Información sobre la versión de la aplicación del alumno

Esta funcionalidad permite visualizar información sobre la aplicación del alumno, una vez que se pulsa el botón izquierdo asociado en la barra de navegación superior del menú principal, tal y como en la Figura 3.47 (versión web) y en la Figura 3.48 (versión Android)

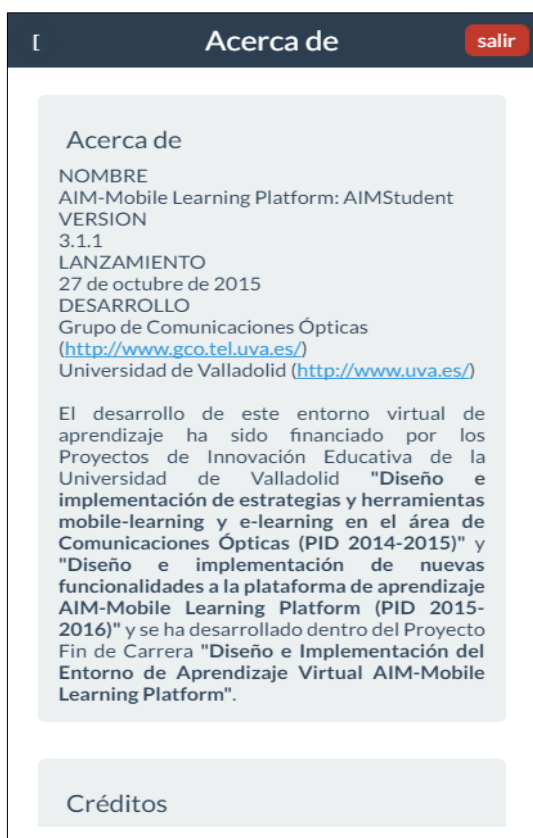


Figura 3.47. Información en la aplicación del alumno (versión web).

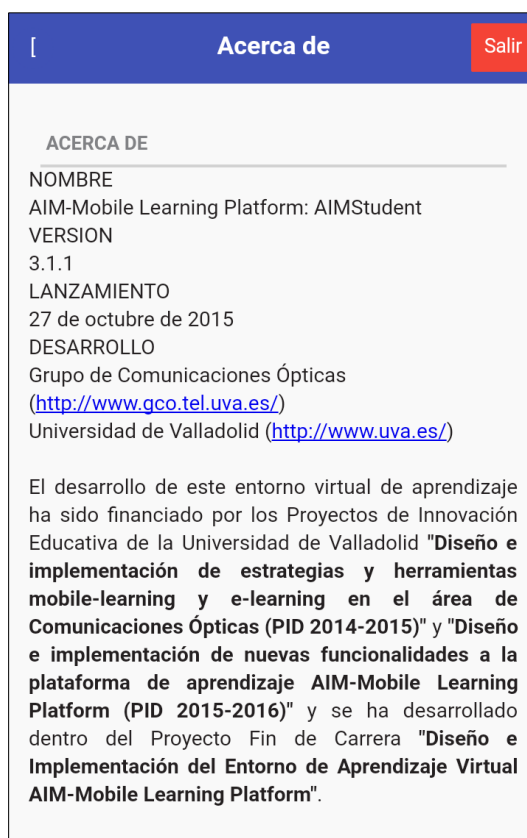


Figura 3.48. Información en la aplicación del alumno (versión Android).

Por otro lado, el fichero «Main.js» en la ruta «app/controller» se encarga de gestionar esta funcionalidad. Finalmente, el fichero «Info.js» en la ruta «app/view/» se encarga de la interfaz de usuario de esta funcionalidad.

## 3.4. Descripción de la API

El fin del desarrollo de la aplicación del alumno es una aplicación móvil o web que, para implementar sus funciones, necesita intercambiar información con una base de datos. Para ello, se desarrolla una API.

Una API sencillamente es un conjunto de métodos a utilizar por otro *software*. En AIM-Mobile Learning Platform se utiliza como intermediaria para comunicar la plataforma del alumno con la base de datos.

En este apartado se describe la arquitectura *software* asociada a la API siguiendo en parte el patrón MVC.

### 3.4.1. Arquitectura *software* de la API

En la Figura 3.49 se muestra el árbol de directorios del proyecto «AIMML-API» que representa la arquitectura *software* de la API.

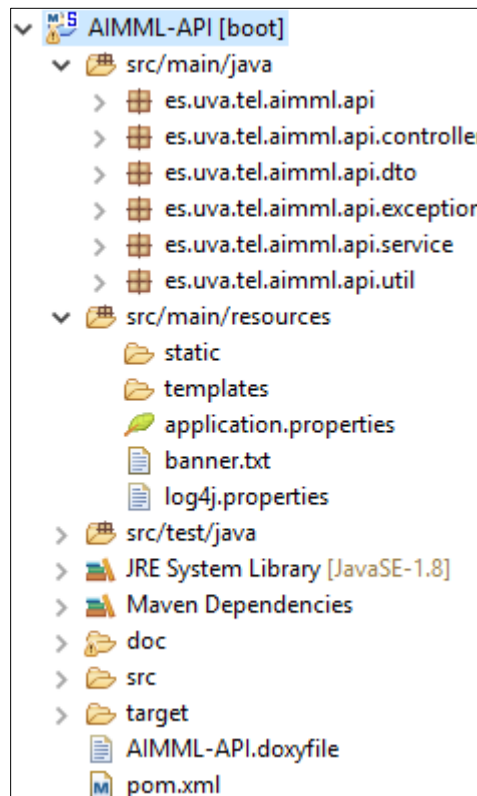


Figura 3.49. Árbol de directorios de la API.

La API tiene una arquitectura *software* similar a la aplicación del profesor puesto que también se desarrolla utilizando el *framework* Spring. La principal diferencia con respecto a la aplicación del profesor es que no necesita desarrollar las vistas asociadas a la interfaz de usuario siguiendo el patrón MVC, puesto que únicamente se utiliza de intermediaria entre la aplicación del alumno y la base de datos.

En el directorio «src/main/java» se encuentran los ficheros Java asociados a la lógica de negocio siguiendo el patrón MVC. De este modo, la lógica de negocio

relacionada con el tablón de avisos se encarga de gestionar los diferentes avisos asociados a un alumno, es decir, obtener los diferentes avisos asociados.

La lógica de negocio relacionada con los libros electrónicos se encarga de gestionar los diferentes libros electrónicos asociados alumno, es decir, obtener los diferentes libros electrónicos asociados, así como las páginas y videos asociados al libro electrónico.

Finalmente, la lógica de negocio relaciona con las preguntas se encarga de gestionar las diferentes preguntas asociadas a un alumno. Así, es el encargado de obtener las preguntas activas asociadas, las diferentes notas, comprobar la contraseña, obtener las diferentes soluciones asociadas a una pregunta, enviar la respuesta, etc.

### 3.5. Descripción de la base de datos

Una base de datos se define en [49] como una «*colección de datos integrados, con redundancia controlada y con una estructura que refleje las interrelaciones y restricciones existentes en el mundo real; los datos, que han de ser compartidos por diferentes usuarios y aplicaciones, deben mantenerse independientes de éstas, y su definición y descripción, únicas para cada tipo de datos, han de estar almacenadas junto con los mismos. Los procedimientos de actualización y recuperación, comunes y bien determinados, habrán de ser capaces de conservar la integridad, seguridad y confidencialidad del conjunto de los datos*».

El SGBD (Sistema de Gestión de Bases de Datos) es el *software* que permite al usuario la utilización y/o la actualización de los datos contenidos en una o varias bases de datos. Su objetivo fundamental consiste en proporcionar al usuario las herramientas necesarias para manipular los datos, de forma que éste se vea abstraído del modo en el que se almacenan de los datos en la máquina y del método de acceso empleado [50].

En el entorno de aprendizaje AIM-Mobile Learning Platform se utiliza MariaDB [51] como SGBD. MariaDB es un *software* derivado como *fork* de MySQL con más funcionalidades y un mejor rendimiento. Surge en 2009 a raíz de la compra de Sun Microsystems (MySQL) por parte de Oracle para mantener una versión bajo licencia GPL (GNU *General Public License*, Licencia Pública General GNU). Es un SGDB relacional, multiusuario, multihilo que, entre otras características, añade nuevos mecanismos de almacenamiento como Aria y XtraDB, y también aumenta la velocidad al guardar datos temporales en memoria mientras que MySQL utiliza el disco duro. Además, tiene tablas de bases de datos con estadísticas para optimizar las conexiones a las bases de datos e implementa un sistema de conexiones *pool-of-threads* en vez del tradicional *thread-per-connection*.

La persistencia de datos es la necesidad de almacenar los contenidos en un sistema de almacenamiento, normalmente una base de datos relacional, para preservar los datos para su uso posterior. En Java, estos datos se conocen como entidades, objetos Java que necesitan ser almacenados. La transformación entre

entidades y tablas de una base de datos relacional se realiza a través del mapeo objeto-relacional.

Existen cuatro tipos de relaciones entre instancias de entidades:

1. «OneToOne»: una instancia de una entidad se puede relacionar con una instancia de la otra entidad.
2. «OneToMany»: una instancia de una entidad se puede relacionar con varias instancias de la otra entidad.
3. «ManyToOne»: varias instancias de una entidad se pueden relacionar con una instancia de la otra entidad.
4. «ManyToMany»: varias instancias de una entidad se pueden relacionar con varias instancias de la otra entidad.

Para implementar la persistencia de datos y el mapeo objeto-relacional en AIM-Mobile Learning Platform, se utiliza conjuntamente Hibernate y Spring Data JPA.

El módulo Spring Data JPA, del proyecto Spring Data, permite implementar repositorios basados en JPA (*Java Persistence API*, API de Persistencia Java). Además, facilita el acceso a los datos en bases de datos relacionales a través de la JPA. El mapeo objeto-relacional se puede realizar mediante ficheros XML o mediante anotaciones en el código.

### 3.5.1. Arquitectura *software* de la base de datos

En la Figura 3.50 se muestra el árbol de directorios del proyecto «AIMML-core» que representa la arquitectura *software* de la base de datos. En el directorio «src/main/java», en el paquete «es.uva.tel.aimml.core.domain», se definen las clases Java asociadas a las entidades, como se observa en la Figura 3.51.

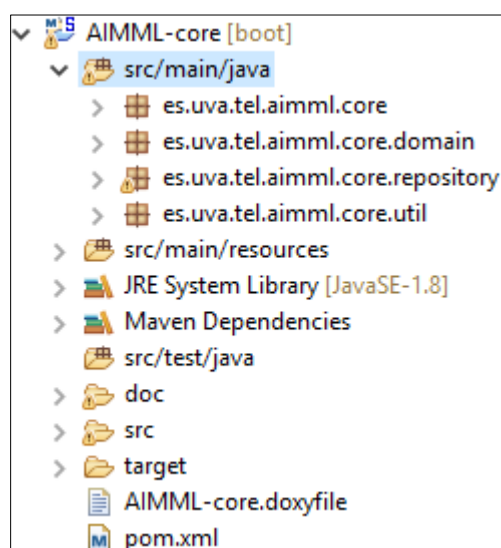
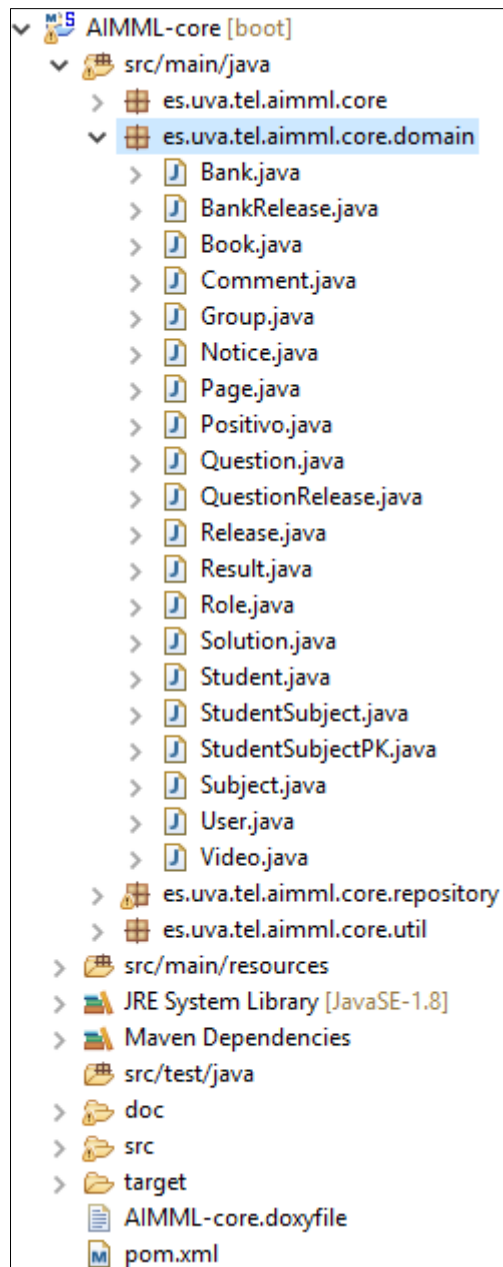
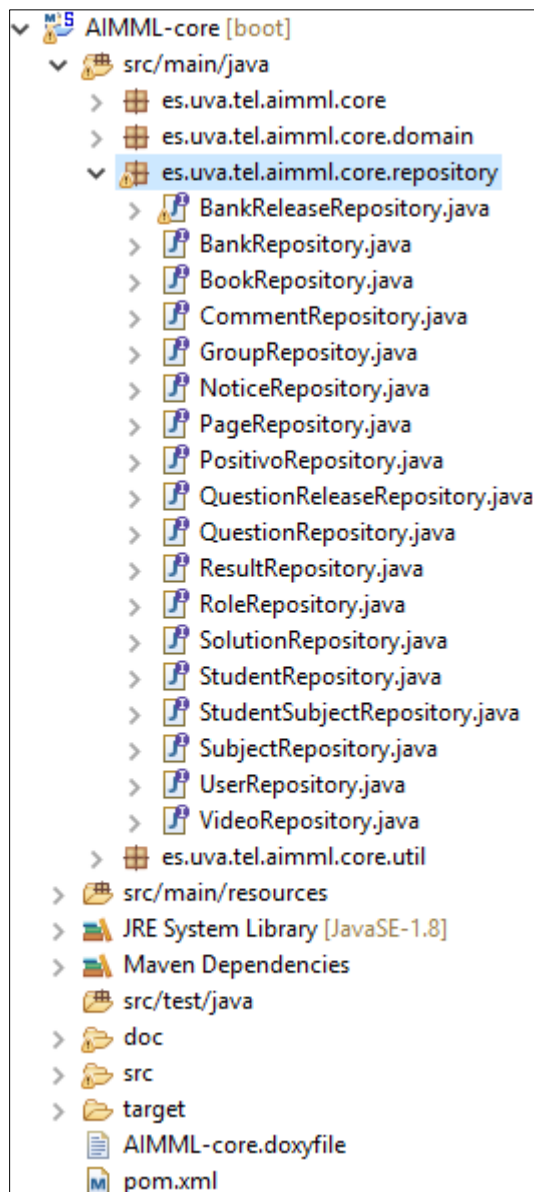


Figura 3.50. Paquetes Java dentro del directorio «src/main/java» del proyecto «AIMML-core».



**Figura 3.51.** Ficheros Java dentro del paquete «es.uva.tel.aimml.core.domain» del directorio «src/main/java» del proyecto «AIMML-core».

Por otro lado, en el directorio «src/main/java», en el paquete «es.uva.tel.aimml.core.repository», se definen las interfaces Java asociadas (Figura 3.52).



**Figura 3.52.** Ficheros Java dentro del paquete «es.uva.tel.aimml.core.repository» del directorio «src/main/java» del proyecto «AIMML-core».

En el desarrollo de nuestro entorno virtual, para el mapeo objeto-relacional se utilizan anotaciones en el código, creando una estructura de la base de datos como (Figura 3.53). En dicha figura se puede ver, en primer lugar, las entidades de AIM-Mobile Learning Platform:

- «users»: almacena los datos de los profesores.
- «roles»: almacena los diferentes roles de la plataforma.
- «students»: almacena los datos de los alumnos.
- «subjects»: almacena los datos de las asignaturas.
- «notices»: almacena los datos de los avisos.



- «books»: almacena los datos de los libros electrónicos.
- «pages»: almacena los datos de las páginas de los libros electrónicos.
- «videos»: almacena los datos de los vídeos.
- «questions»: almacenar los datos de las preguntas.
- «solutions»: almacena los datos de las soluciones de las preguntas.
- «question\_releases»: almacena los datos de las preguntas publicadas.
- «results»: almacena los datos de los resultados de las preguntas.
- «banks»: almacena los datos de los bancos de preguntas.
- «bank\_releases»: almacena los datos de los bancos de preguntas publicados.
- «groups»: almacena los datos de los grupos de laboratorio.
- «positivo»: almacena los datos de los positivos y negativos en la evaluación cualitativa de los alumnos.
- «comments»: almacena los datos de los comentarios en la evaluación cualitativa de los alumnos.
- «modules»: almacena los datos de los módulos de la plataforma del profesor.

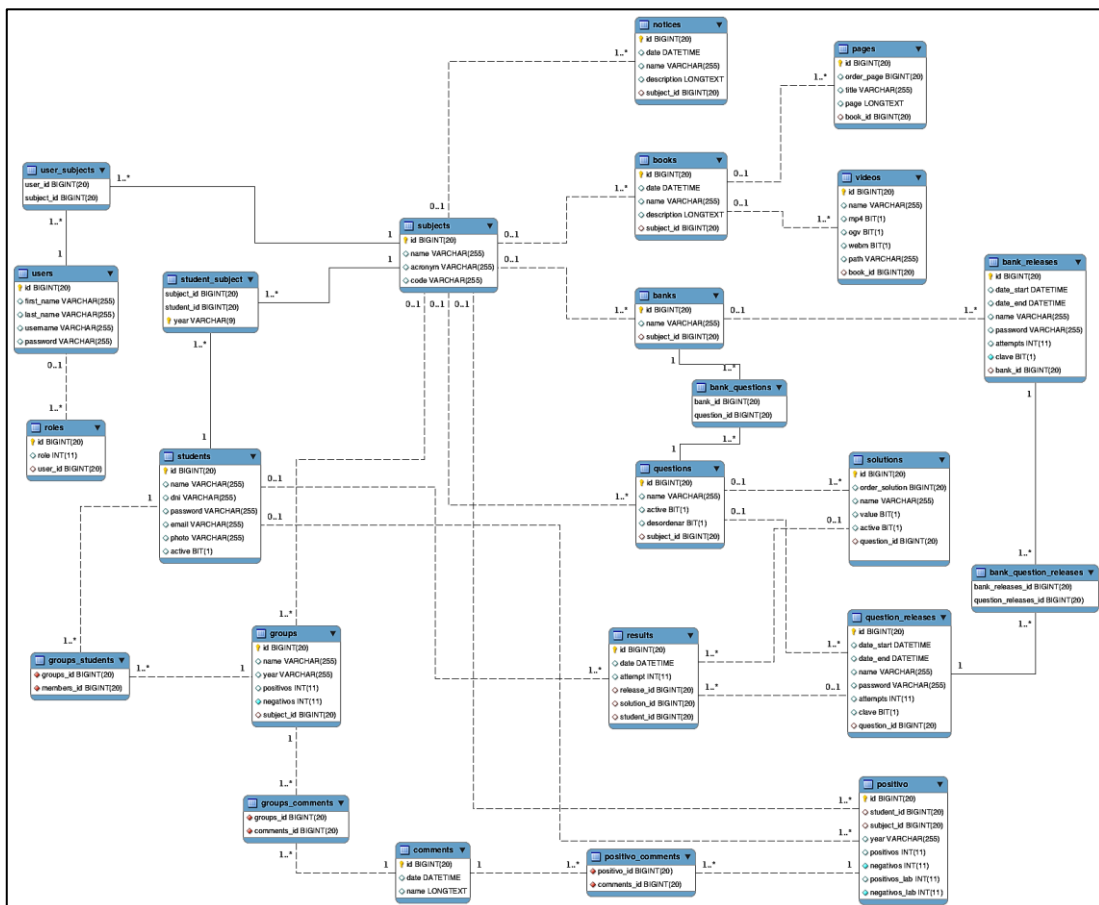


Figura 3.53. Diagrama de la base de datos de AIM-Mobile Learning Platform.

Y, en segundo lugar, en esa misma figura, se pueden observar las relaciones entre las instancias de las entidades de AIM-Mobile Learning Platform:

- «users» con «roles»: de tipo «OneToOne» en ambos sentidos, es decir, una instancia de «users» se puede relacionar con una instancia de «roles» y viceversa.
- «users» con «subjects»: de tipo «ManyToMany» en ambos sentidos, es decir, varias instancias de «users» se pueden relacionar con varias instancias de «subjects» y viceversa.
- «students» con «subjects»: de tipo «OneToMany» en ambos sentidos, es decir, una instancia de «students» se puede relacionar con varias instancias de «subjects» y una instancia de «subjects» se puede relacionar con varias instancias de «students».
- «students» con «results»: de tipo «OneToMany» y «OneToOne» en sentido opuesto, es decir, una instancia de «students» se puede relacionar con varias instancias de «results» y una instancia de «results» se puede relacionar con una instancia de «students».
- «students» con «groups»: de tipo «ManyToMany» en ambos sentidos, es decir, varias instancias de «students» se pueden relacionar con varias instancias de «groups» y viceversa.
- «students» con «positivo»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «students» se puede relacionar con varias instancias de «positivo» y varias instancias de «positivo» se pueden relacionar con una instancia de «students».
- «subjects» con «notices»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «subjects» se puede relacionar con varias instancias de «notices» y varias instancias de «notices» se pueden relacionar con una instancia de «subjects».
- «subjects» con «books»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «subjects» se puede relacionar con varias instancias de «books» y varias instancias de «books» se pueden relacionar con una instancia de «subjects».
- «subjects» con «questions»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «subjects» se puede relacionar con varias instancias de «questions» y varias instancias de «questions» se pueden relacionar con una instancia de «books».
- «subjects» con «banks»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «subjects» se puede relacionar con varias instancias de «banks» y varias instancias de «banks» se pueden relacionar con una instancia de «subjects».
- «subjects» con «student\_subject»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «subjects» se puede relacionar con varias instancias de «student\_subject» y varias instancias de «student\_subject» se pueden relacionar con una instancia de «subjects».
- «books» con «pages»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «books» se puede relacionar con varias

instancias de «pages» y varias instancias de «pages» se pueden relacionar con una instancia de «books».

- «books» con «videos»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «books» se puede relacionar con varias instancias de «videos» y varias instancias de «videos» se pueden relacionar con una instancia de «books».
- «questions» con «solutions»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «questions» se puede relacionar con varias instancias de «solutions» y varias instancias de «solutions» se pueden relacionar con una instancia de «questions».
- «questions» con «question\_releases»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «questions» se puede relacionar con varias instancias de «question\_releases» y varias instancias de «question\_releases» se pueden relacionar con una instancia de «questions».
- «questions» con «banks»: de tipo «ManyToMany» en ambos sentidos, es decir, varias instancias de «questions» se pueden relacionar con varias instancias de «banks» y viceversa.
- «questions\_releases» con «results»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «question\_releases» se puede relacionar con varias instancias de «results» y varias instancias de «results» se pueden relacionar con una instancia de «question\_releases».
- «results» con «solutions»: de tipo «OneToOne», es decir, una instancia de «results» se puede relacionar con una instancia de «solutions».
- «banks» con «bank\_releases»: de tipo «OneToMany» y «ManyToOne» en sentido opuesto, es decir, una instancia de «banks» se puede relacionar con varias instancias de «bank\_releases» y varias instancias de «bank\_releases» se pueden relacionar con una instancia de «banks».
- «bank\_releases» con «question\_releases»: de tipo «OneToMany», es decir, una instancia de «bank\_releases» se puede relacionar con varias instancias de «question\_releases».
- «groups» con «subjects»: de tipo «ManyToOne», es decir, varias instancias de «groups» se pueden relacionar con una instancia de «subjects».
- «groups» con «students»: de tipo «ManyToMany» en ambos sentidos, es decir, varias instancias de «groups» se pueden relacionar con varias instancias de «students» y viceversa.
- «groups» con «comments»: de tipo «OneToMany», es decir, una instancia de «groups» se pueden relacionar con varias instancias de «comments».
- «positivo» con «students»: de tipo «ManyToOne», es decir, varias instancias de «positivo» se puede relacionar con una instancia de «students».
- «positivo» con «subjects»: de tipo «ManyToOne», es decir, varias instancias de «positivo» se puede relacionar con una instancia de «subjects».
- «positivo» con «comments»: de tipo «OneToMany», es decir, una instancia de «positivo» se puede relacionar con varias instancias de «comments».

## 3.6. Conclusiones

En este Capítulo 3 se ha descrito la versión *software* del entorno virtual de aprendizaje AIM-Mobile Learning Platform previa al inicio de este TFM.

Para ello, se ha dividido la explicación del entorno virtual de aprendizaje en cuatro partes: la aplicación del profesor, la aplicación del alumno, la API y la base de datos. En este Capítulo 3 se han descrito cada una de ellas por separado sin perder de vista el entorno virtual de aprendizaje de forma global.

Por otro lado, también se dividió la descripción de la aplicación del profesor y la del alumno en dos partes: desde el punto de vista del desarrollador *software* y desde el punto de vista del usuario. Sin embargo, la descripción de la API y la base de datos solo se realiza desde el punto de vista del desarrollador *software*, puesto que no disponen de una interfaz de usuario.

De este modo, se elige describir el primer punto de vista, el del desarrollador *software*, para facilitar el desarrollo de futuras versiones. Del mismo modo, el segundo punto de vista, el del usuario, es necesario para obtener una visión global de cada una de las partes del entorno virtual de aprendizaje.

En la aplicación del profesor, o *back-end* desde el punto de vista del desarrollo web, se dispone de una interfaz de usuario en la que un profesor puede utilizar la aplicación del profesor de forma sencilla. En primer lugar, accediendo de forma segura, después, creando y gestionando contenidos a visualizar por el alumno, y finalmente, evaluando de forma continua a los alumnos y gestionando sus resultados. Además, un administrador puede gestionar todos los profesores dados de alta en el entorno virtual, así como asignaturas, estudiantes y copias de seguridad de la plataforma.

En la aplicación del alumno, o *front-end* desde el punto de vista del desarrollo web o aplicación cliente desde el punto de vista de la arquitectura web, también se dispone de una interfaz de usuario sencilla e intuitiva, tanto en un navegador web como en una aplicación nativa en Android e iOS gracias a la combinación de Sencha Touch y Apache Cordova. Un alumno puede utilizar la aplicación, primero, accediendo de forma segura, después, visualizando los contenidos creados por el profesor desde su aplicación correspondiente, y finalmente, respondiendo a las preguntas creadas por el profesor para su evaluación continua y visualizando su progreso a lo largo de la asignatura.

En tercer lugar, se dispone de una API REST (*REpresentational State Transfer*, Transferencia de Estado Representacional) que permite el acceso a la base de datos desde la aplicación del alumno para realizar sus funcionalidades.

Finalmente, y en cuarto lugar se dispone de una base de datos relacional que permite la persistencia de los datos del entorno virtual de aprendizaje, tanto en la parte de la aplicación del profesor como en la parte del alumno, creando, actualizando, borrando o leyendo los datos. Desde el punto de vista del desarrollo *software*, dispone de un conjunto de entidades con diseño escalable y un potente SGBD, como MariaDB, para evitar errores en la conexión con la base de datos y cuellos de botella.

# 4

## Nuevas funcionalidades de la plataforma del alumno

### 4.1. Introducción

En el Capítulo 4 se describen las nuevas funcionalidades que se han desarrollado en la aplicación del alumno, aplicación cliente, como parte de este TFM.

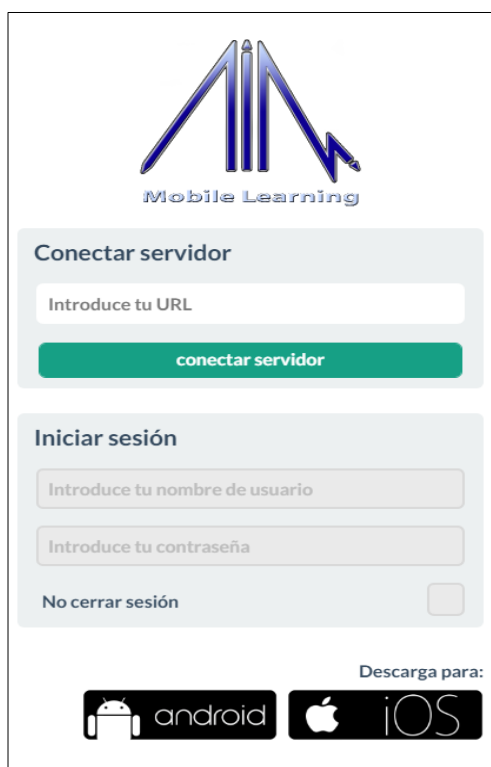
### 4.2. Inicio de conexión con la API y el servidor

En esta primera nueva funcionalidad de la aplicación del alumno, se ha modificado el flujo de inicio de la aplicación. En primer lugar, se inicia una conexión con el servidor web de aplicaciones, es decir, se comprueba que se puede acceder a la API en el servidor web de aplicaciones antes de que el usuario introduzca sus credenciales para iniciar la sesión. Permite que cualquier centro de enseñanza o institución educativa pueda tener su entorno virtual con su propia base de datos y API desplegada en su propio servidor web de aplicaciones. Por lo tanto, esta nueva funcionalidad le otorga mayor potencialidad y flexibilidad al entorno de aprendizaje virtual.

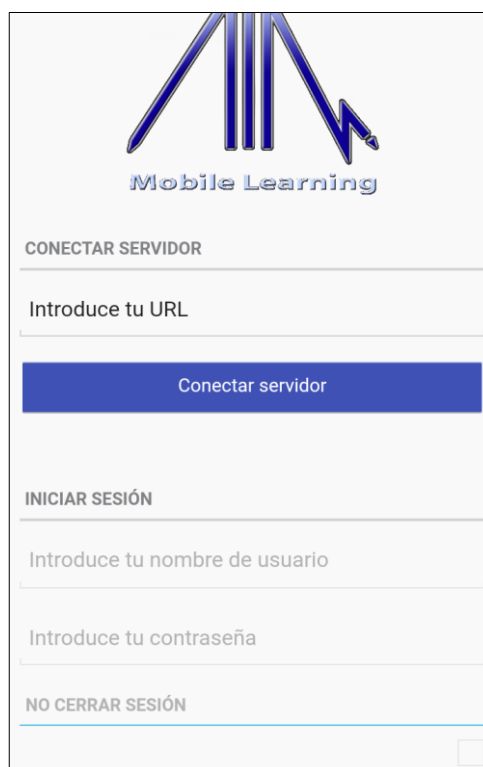
Gracias a esta nueva funcionalidad no se necesita que el fichero WAR (*Web application ARchive*, Fichero de Aplicación Web) del proyecto «AIMStudent» correspondiente a la aplicación del alumno se despliegue en el mismo servidor web de aplicaciones que los ficheros WAR del proyecto «AIMML-Admin» correspondiente a la aplicación del profesor y del proyecto «AIMML-API» correspondiente a la API.

Para ello, se han modificado los ficheros «app.js» y el fichero «Login.js» en la ruta «app/controller» del proyecto en Sencha Touch que se encarga de gestionar esta nueva funcionalidad, así como el fichero «Login.js» en la ruta «app/view/» del

proyecto en Sencha Touch, que se encarga de la interfaz de usuario de esta nueva funcionalidad. En concreto, se ha añadido una caja de texto en la vista principal en el que el usuario tiene que introducir un fragmento de la URL (*Uniform Resource Locator*, Localizador Uniforme de Recursos) del directorio de despliegue del fichero WAR del proyecto «AIMML-API» correspondiente a la API: «http://su.servidor.com/AIMML-API». Cabe destacar que no se necesita introducir el protocolo de comunicación («http://») ni el directorio de despliegue del fichero WAR del proyecto «AIMML-API» correspondiente a la API («/AIMML-API»). También se ha añadido un botón de inicio de conexión, tal y como se observa en la Figura 4.1 (versión web) y en la Figura 4.2 (versión Android) que, una vez pulsado realiza una petición a la API para comprobar que existe conexión con ella.



**Figura 4.1.** Inicio de conexión con la API y el servidor en la aplicación del alumno (versión web).



**Figura 4.2.** Inicio de conexión con la API y el servidor en la aplicación del alumno (versión Android).

Si existe conexión con la API, esta responde indicando que la petición al servicio web es correcta. A continuación, se visualiza un mensaje de alerta a una nueva ventana emergente que informa al usuario sobre el éxito de la conexión, como se muestra en la Figura 4.3 (versión web) y en la Figura 4.4 (versión Android).

En segundo lugar, se visualiza otro mensaje de alerta que permite al usuario elegir si quiere guardar en su almacenamiento local el fragmento de URL previamente introducido (Figura 4.5 (versión web) y en la Figura 4.6 (versión Android)). De esta manera, el usuario alumno no tiene que volver a introducir el nombre del servidor al que se quiere conectar cuando desee establecer una nueva sesión. Para llevar a cabo esta nueva funcionalidad, se ha creado el fichero «Url.js» en la ruta «app/model/» y el fichero «Urls.js» en la ruta «app/store/» del proyecto en Sencha Touch. Estos nuevos ficheros permiten almacenar localmente el fragmento de URL (nombre del servidor de aplicaciones).



Figura 4.3. Conexión con la API y el servidor iniciada en la aplicación del alumno (versión web).

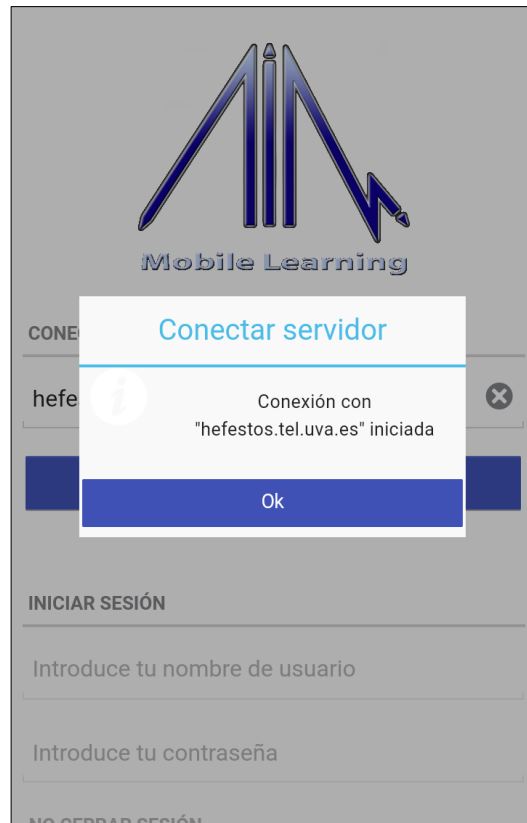


Figura 4.4. Conexión con la API y el servidor iniciada en la aplicación del alumno (versión Android).

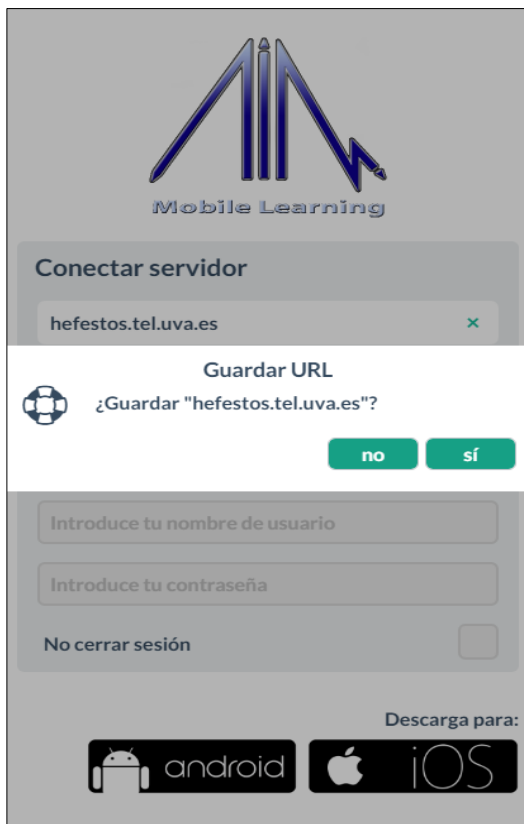


Figura 4.5. Opción de almacenar localmente la URL (versión web).

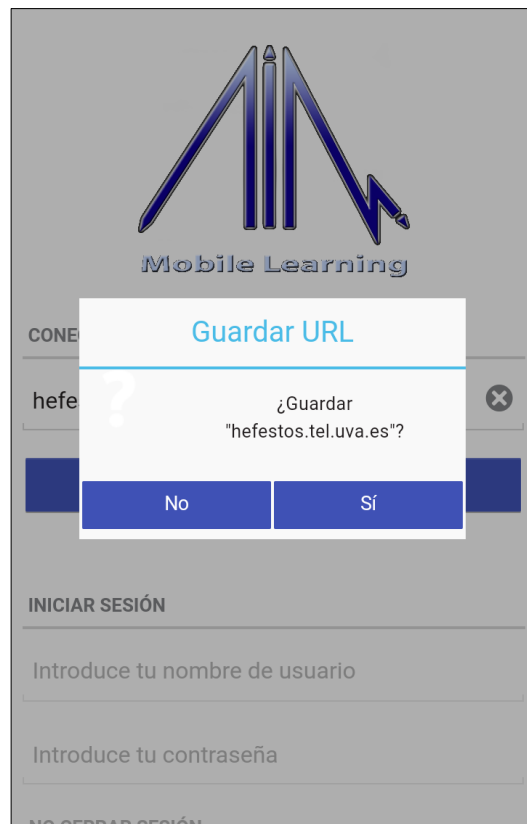


Figura 4.6. Opción de almacenar la URL localmente (versión Android).

En último lugar, se visualiza otro mensaje de alerta en una nueva ventana emergente que informa al usuario alumno sobre su elección. Por ejemplo, como se puede ver en la Figura 4.7 (versión web) y en la Figura 4.8 (versión Android) el usuario alumno ha elegido que se guarde el nombre del servidor de aplicaciones (URL) para futuras sesiones.

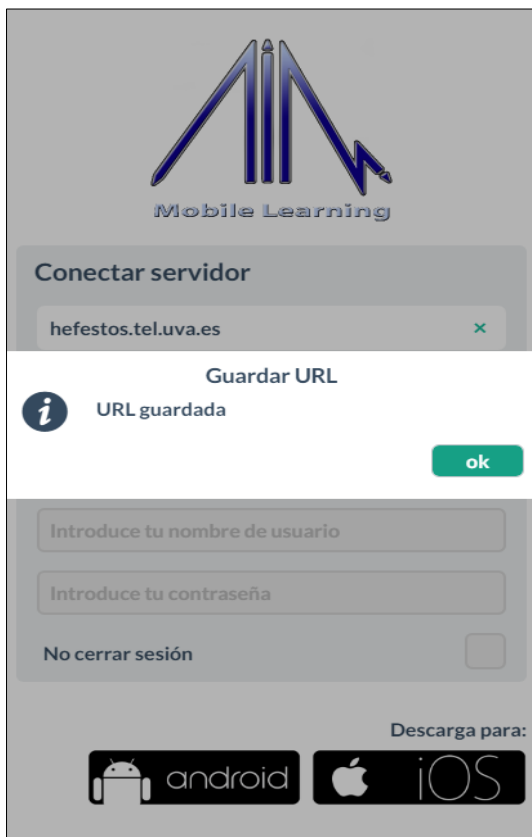


Figura 4.7. URL localmente almacenada (versión web).

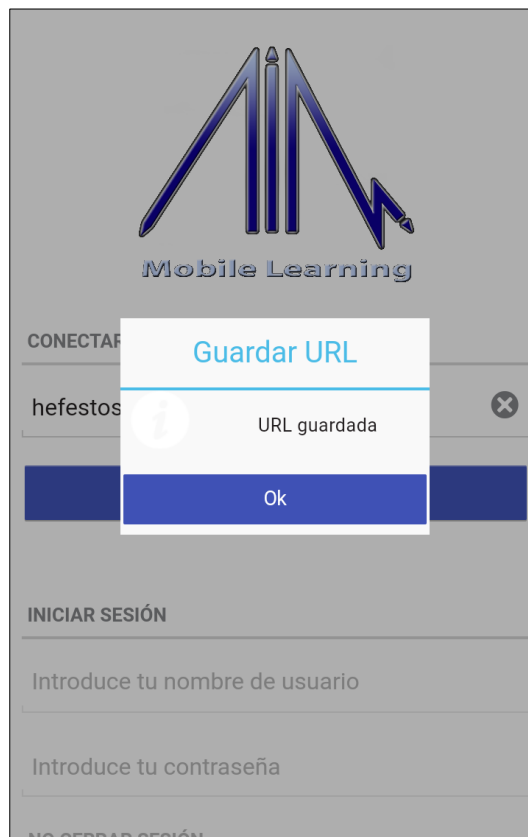


Figura 4.8. URL localmente almacenada (versión Android).

Tras iniciar la conexión con la API, se habilitan los campos de inicio de sesión (en la parte inferior de la vista principal de acceso) en los que el usuario tiene que introducir sus credenciales.

En concreto, aparecen los botones de inicio de sesión (para acceder al resto de funcionalidades de la aplicación del alumno), de re-inicialización de contraseña y de registro, tal y como se muestra en la Figura 4.9 (versión web) y en la Figura 4.10 (versión Android).

Finalmente, también se deshabilita el campo de formulario en el que el usuario introdujo el fragmento de URL de la API. En caso de no poder realizarse una conexión con la API, se muestra un mensaje de alerta en forma de ventana emergente que informa al usuario sobre el fracaso en la conexión, tal y como se muestra en la Figura 4.11 (versión web) y en la Figura 4.12 (versión Android).



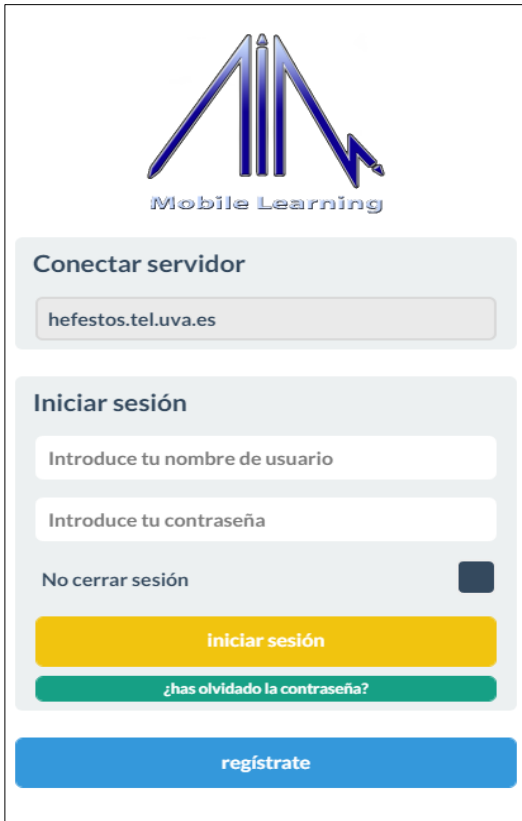


Figura 4.9. Conexión con la API y el servidor iniciada (versión web).

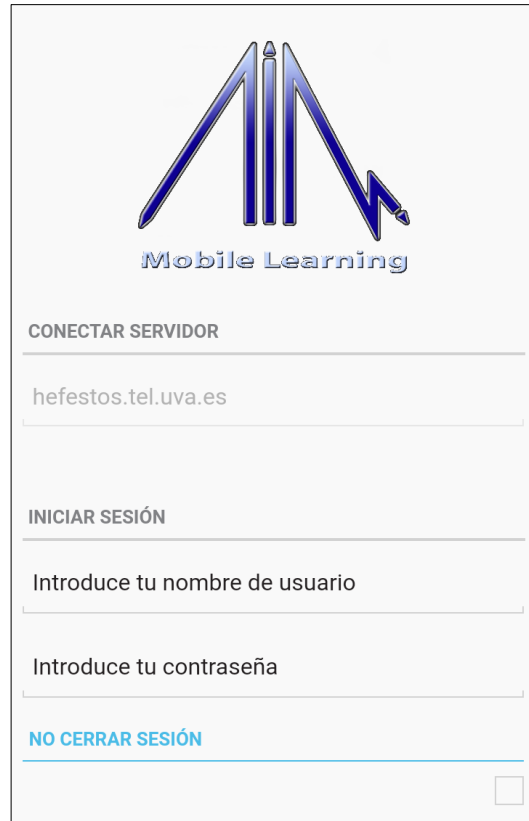


Figura 4.10. Conexión con la API y el servidor iniciada (versión Android).



Figura 4.11. Conexión con la API y el servidor no iniciada (versión web).

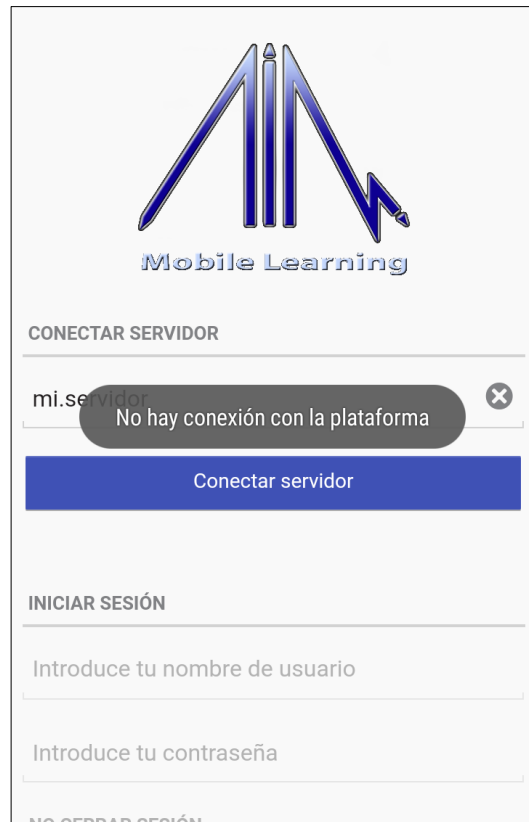


Figura 4.12. Conexión con la API y el servidor no iniciada (versión Android).

Finalmente, antes de realizar la petición a la API para comprobar que hay conexión con ella se valida si el campo de formulario en el que el usuario tiene que introducir el fragmento de URL de la API está vacío. En el caso en el que el usuario no hay introducir ninguna dirección, se muestra un mensaje de alerta en forma de ventana emergente que informa de dicho error al usuario (Figura 4.13 (versión web) y Figura 4.14 (versión Android)).

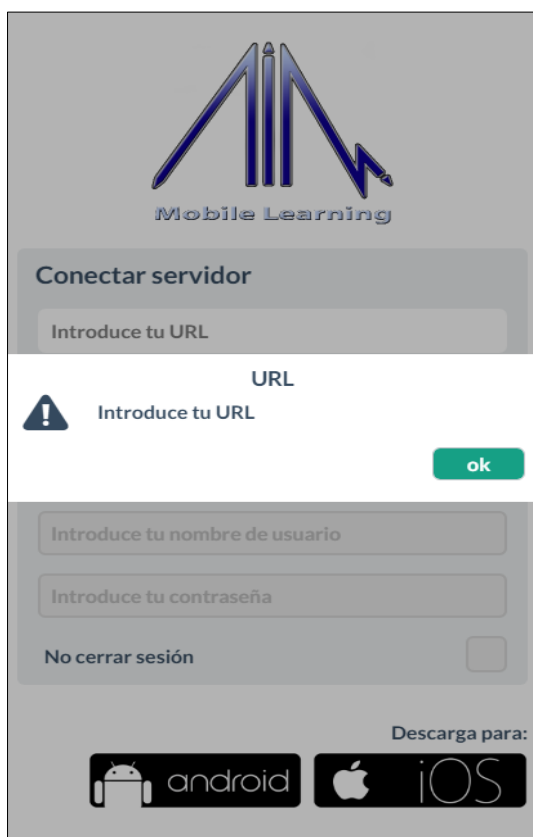


Figura 4.13. URL vacía (versión web).

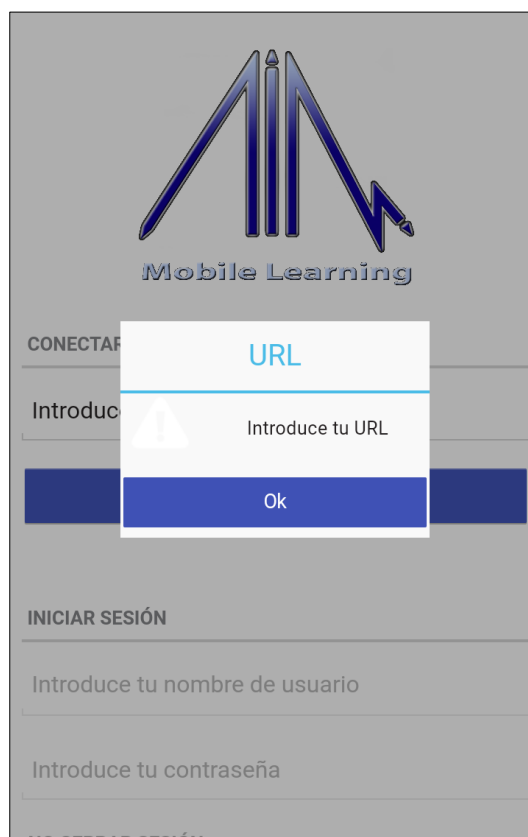


Figura 4.14. URL vacía (versión Android).

La opción de guardar en el almacenamiento local del usuario el fragmento de URL de la API permite agilizar el proceso de conexión con la API en futuras sesiones puesto que el usuario ya no tendrá que introducirlo en el campo del formulario.

En este caso, simplemente tendrá que pulsar el botón de inicio de conexión, tal y como se muestra en la Figura 4.15 (versión web) y en la Figura 4.16 (versión Android). Además, si el fragmento de URL es el mismo que el guardado en el almacenamiento local, ya no se preguntará de nuevo si quiere guardarlo. Si previamente también se guardaron las credenciales de usuario, se indicará de la misma manera (Figura 4.17 (versión web) y Figura 4.18 (versión Android)). En caso contrario, tendrá que volver a introducirlas. Por lo tanto, la aplicación del alumno también introduce la funcionalidad de guardar el *login* y contraseña del alumno para recordarlas en futuras sesiones.

Lógicamente, el fragmento de URL de la API se tiene que corresponder con las credenciales de usuario localmente almacenadas, si no, se tendrán que volver a introducir e iniciar la sesión.

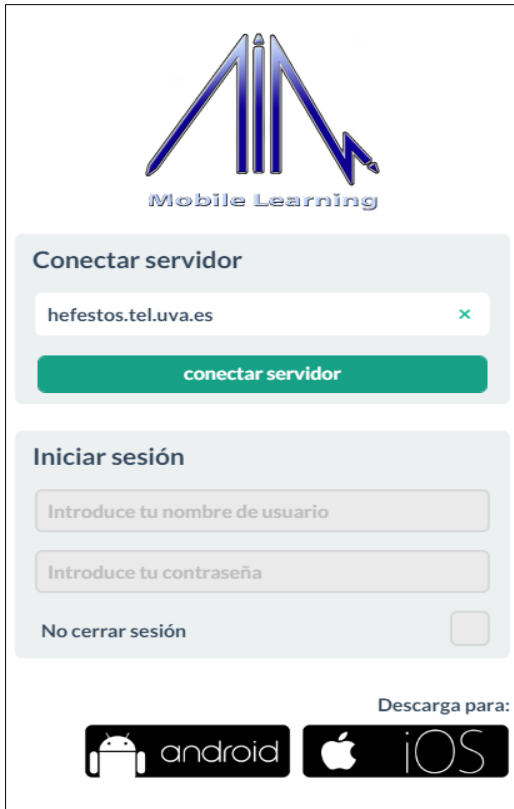


Figura 4.15. Inicio de conexión con la API y el servidor con la URL localmente almacenada (versión web).

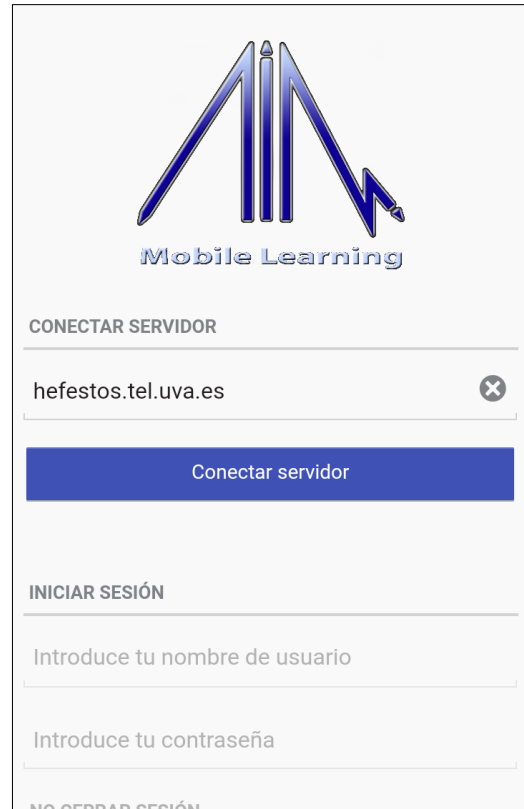


Figura 4.16. Inicio de conexión con la API y el servidor con la URL localmente almacenada (versión Android).

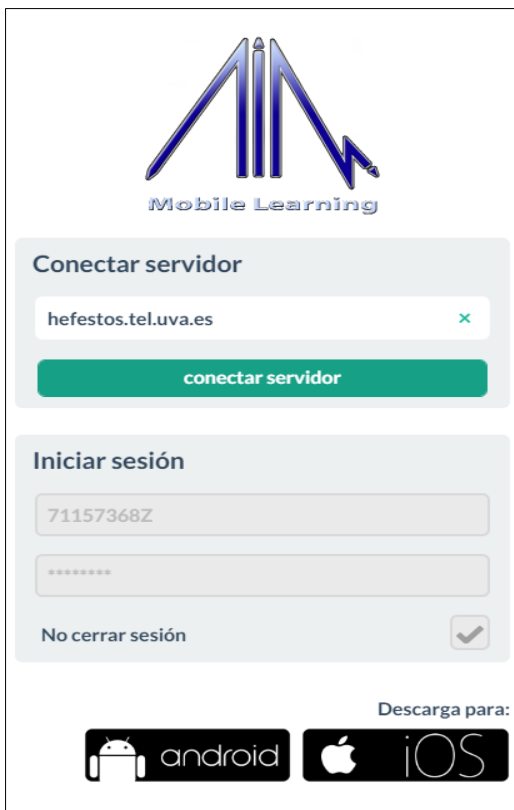


Figura 4.17. Inicio de conexión con la API y el servidor con la URL y las credenciales localmente almacenada (versión web).

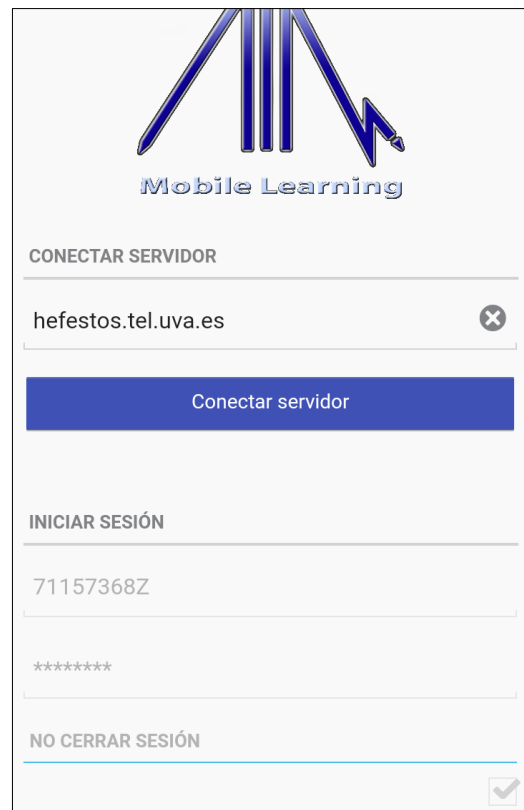


Figura 4.18. Inicio de conexión con la API y el servidor con la URL y las credenciales localmente almacenada (versión Android).

## 4.3. Re-inicialización de la contraseña

Esta nueva funcionalidad permite re-inicializar la contraseña de inicio de sesión en la aplicación del alumno. Es decir, permite al alumno obtener una nueva contraseña vía correo electrónico con la que acceder al resto de funcionalidades de la aplicación en caso de que no recuerde la contraseña que introdujo cuando se registró.

En la versión anterior del entorno virtual de aprendizaje AIM-Mobile Learning Platform, si un estudiante olvidaba la contraseña de inicio de sesión, un usuario administrador en la aplicación del profesor tenía que reiniciar la cuenta del alumno. Para ello, el administrador accedía a la sección de configuración de estudiantes del menú de administración pulsando el botón correspondiente («Reiniciar cuenta»), tal y como se muestra en Figura 4.19, para que el estudiante pudiera volver a registrarse con una nueva contraseña.

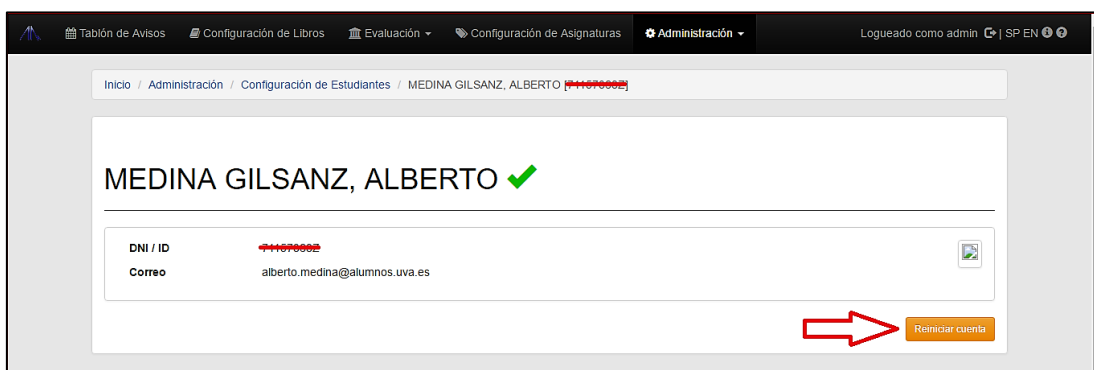


Figura 4.19. Proceso de reinicio de cuenta en la aplicación del profesor.

La opción anterior se mantiene en la nueva versión del entorno de aprendizaje, pero gracias a esta nueva funcionalidad, un estudiante puede re-inicializar su contraseña de inicio de sesión sin tener que ponerse en contacto con el administrador de la plataforma para tal fin. Esta nueva funcionalidad agiliza el proceso y se delegan responsabilidades en el propio entorno virtual de aprendizaje, y más concretamente en el propio usuario alumno.

Para ello, se ha modificado el fichero «Login.js» en la ruta «app/view/» del proyecto en Sencha Touch en el que se ha añadido un botón de re-inicialización de contraseña, tal y como se observa en la Figura 4.20 (versión web) y en Figura 4.21 (versión Android) tras el inicio de conexión con la API.

Tras pulsar dicho botón se visualiza un mensaje de alerta que informa al usuario de que se ha enviado un correo electrónico con su nueva contraseña de inicio de sesión (Figura 4.22 (versión web) y Figura 4.23 (versión Android)).

Tras pulsar el botón de validación se muestra otro mensaje de alerta en el que el usuario tiene que introducir su nombre de usuario (DNI), tal y como se observa en la Figura 4.24 (versión web) y Figura 4.25 (versión Android).

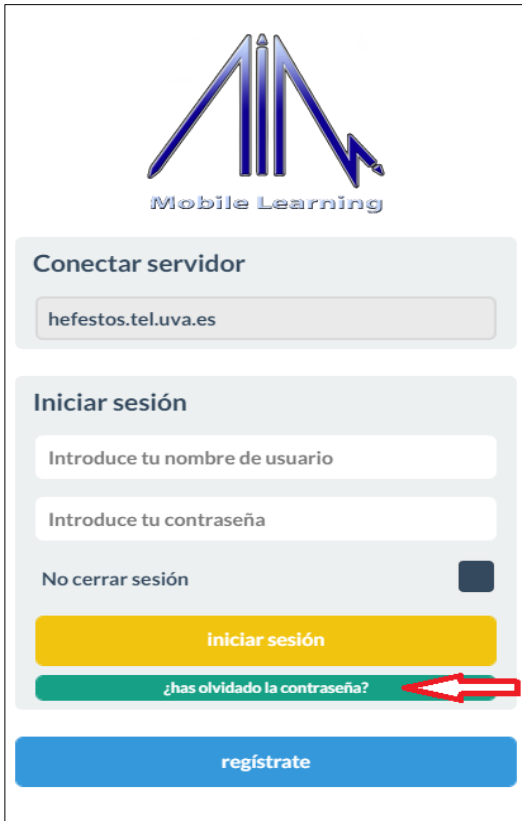


Figura 4.20. Re-inicializar contraseña (versión web).

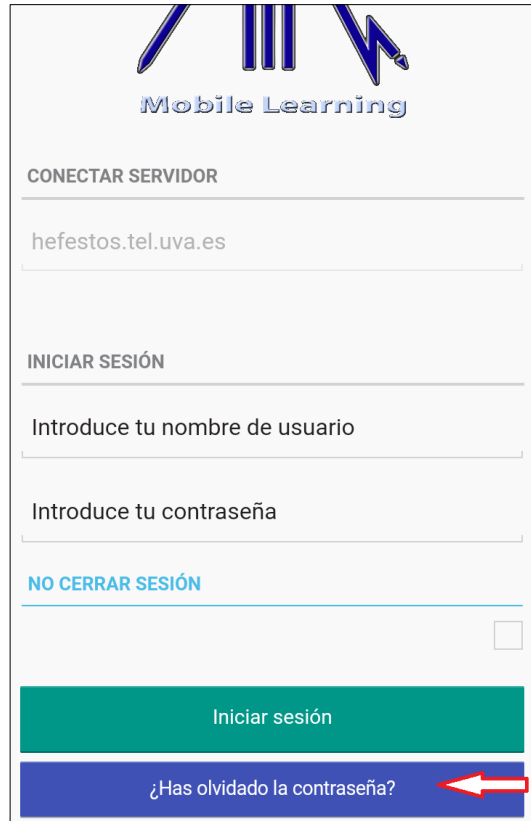


Figura 4.21. Re-inicializar contraseña (versión Android).

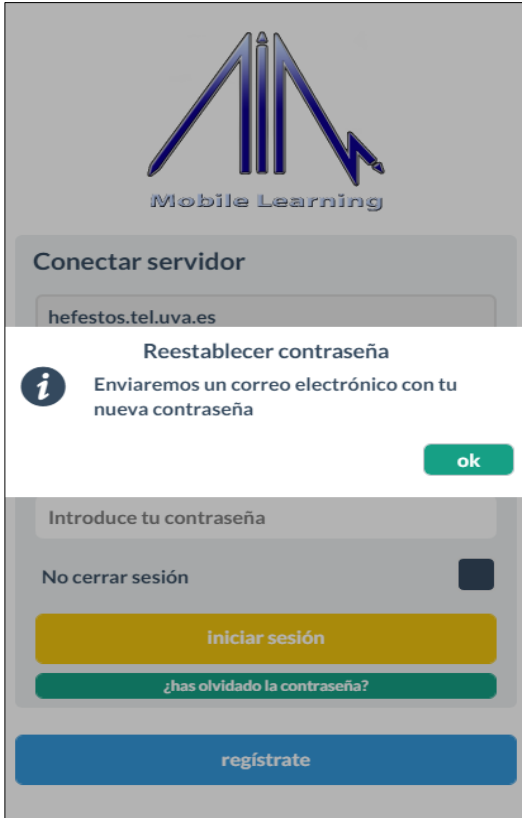


Figura 4.22. Mensaje de re-inicialización de contraseña (versión web).



Figura 4.23. Mensaje de re-inicialización de contraseña (versión Android).

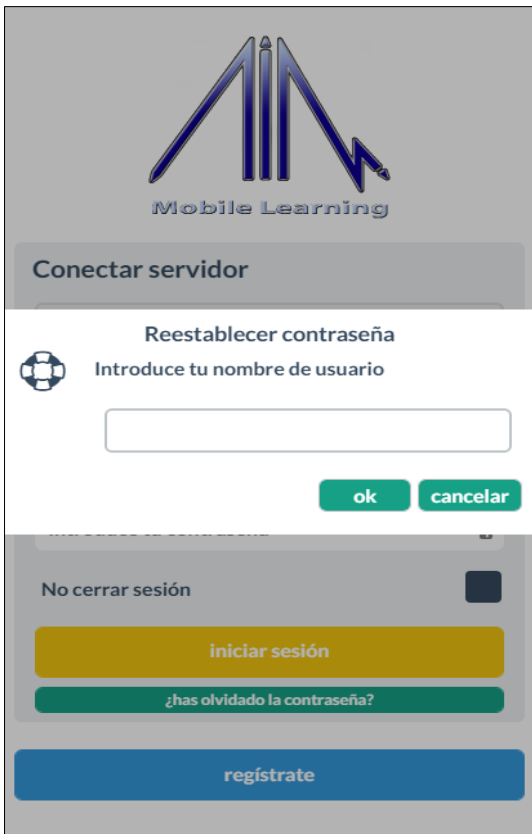


Figura 4.24. Introducción de nombre de usuario para re-inicializar contraseña (versión web).

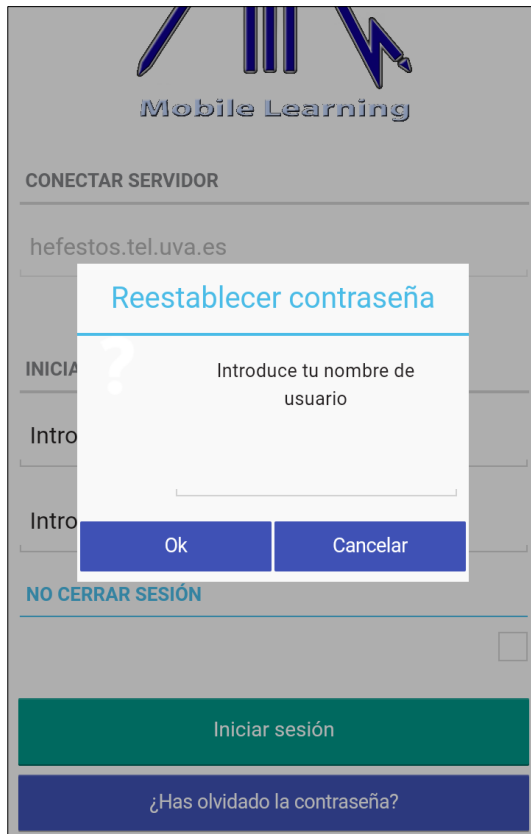


Figura 4.25. Introducción de nombre de usuario para re-inicializar contraseña (versión Android).

Tras pulsar el botón de validación se envía una petición a la API que comprueba si el alumno existe y si se ha registrado previamente en el entorno virtual a través de la aplicación alumno. A continuación, se genera una nueva contraseña actualizando la anterior en la base de datos y se envía un correo electrónico al alumno con la nueva contraseña generada.

Para poder llevar a cabo esta funcionalidad, se necesita que a la hora de añadir un estudiante en la aplicación del profesor se introduzca su dirección de correo electrónico. Si el alumno no tiene asignado una dirección de correo electrónico o si se produce algún problema en el envío del correo electrónico con la nueva contraseña generada, la API responde indicando que la petición al servicio web no es correcta. A continuación, se muestra un mensaje de alerta en la aplicación del alumno que informa al usuario sobre el error como se puede ver en la Figura 4.26 (versión web) y en la Figura 4.27 (versión Android) por un lado, y en la Figura 4.28 (versión web) y en la Figura 4.29 (versión Android) por otro lado.

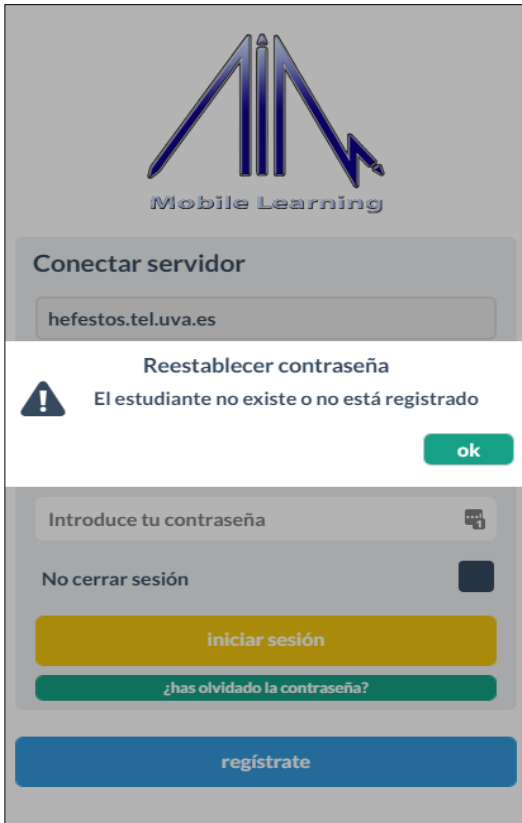


Figura 4.26. Mensaje de alerta en la re-inicialización de contraseña (estudiante) (versión web).



Figura 4.27. Mensaje de alerta en la re-inicialización de contraseña (estudiante) (versión Android).

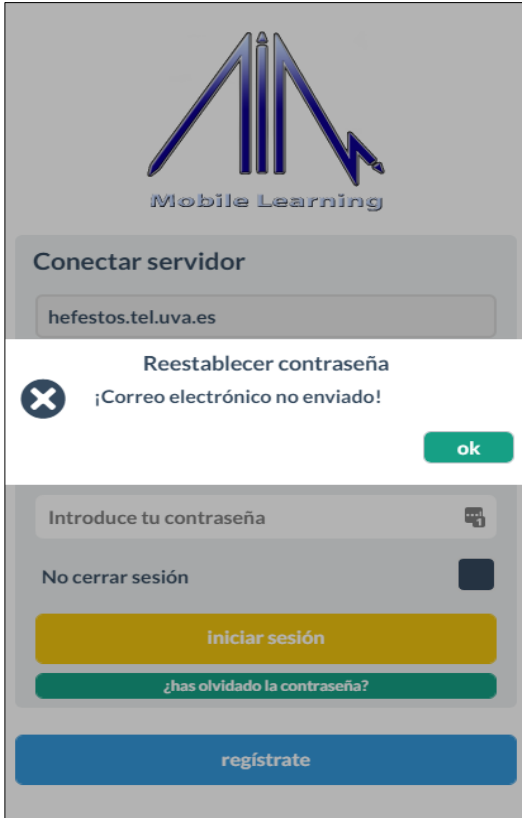


Figura 4.28. Mensaje de alerta en la re-inicialización de contraseña (correo electrónico) (versión web).



Figura 4.29. Mensaje de alerta en la re-inicialización de contraseña (correo electrónico) (versión Android).

Para llevar a cabo esta nueva funcionalidad, en el proyecto en Sencha Touch, se ha modificado el fichero «Login.js» en la ruta «app/controller/». En concreto, se ha añadido el método «resetPassword» que gestiona esta nueva funcionalidad.

Por otro lado, en el proyecto «AIMML-API» en Spring Tool Suite, se ha modificado el fichero «StudentController.java» del paquete Java «es.uva.tel.aimml.api.controller» en el directorio «src/main/java». En concreto, se ha añadido el método «resetPassword», y el fichero «StudentService.java» del paquete Java «es.uva.tel.aimml.api.service» en el mismo directorio que gestionan esta nueva funcionalidad. También se ha creado el fichero «RandomStringGenerator.java» en el paquete Java «es.uva.tel.aimml.api.util» en el directorio. Además, se ha modificado el fichero «SecurityConfig.java» del paquete Java «es.uva.tel.aimml.api» en el mismo directorio. De este modo, se ha modificado el método «configure» para permitir las peticiones a la URL asociada. Finalmente, se ha modificado el fichero «application.properties» en el directorio «src/main/resources» para configurar el envío del correo electrónico.

## 4.4. Manual de ayuda de la aplicación del estudiante

Esta simple funcionalidad permite visualizar un pequeño manual de ayuda al alumno a través de ventanas de información emergentes que aparecen tras pulsar en el símbolo de información.

Este símbolo de información, tal y como se muestra en la Figura 4.30 (versión web) y en la Figura 4.31 (versión Android) aparece a la derecha de cada uno de los botones de las diferentes funcionalidades principales de la aplicación del alumno.

De este modo, cuando se pulsa el botón de información en cada funcionalidad aparece una nueva ventana emergente se muestra información sobre la funcionalidad correspondiente. Por ejemplo, la Figura 4.32 (versión web) y Figura 4.33 (versión Android) muestra un pequeño manual de manejo y descripción de la funcionalidad de avisos.

Para ello, se ha modificado el fichero «MainMenu.js» en la ruta «app/view/» en el proyecto de Sencha Touch. En dicho fichero se han modificado los botones de las funcionalidades principales de la aplicación del alumno, de modo que se han segmentado para añadir los botones de información de las funcionalidades principales de la aplicación.

La gestión de esta nueva funcionalidad se encuentra repartida entre los ficheros «Notices.js», «Book.js», «Questions.js» y «MyQuestions.js» en la ruta «/app/controller/» del proyecto en Sencha Touch.



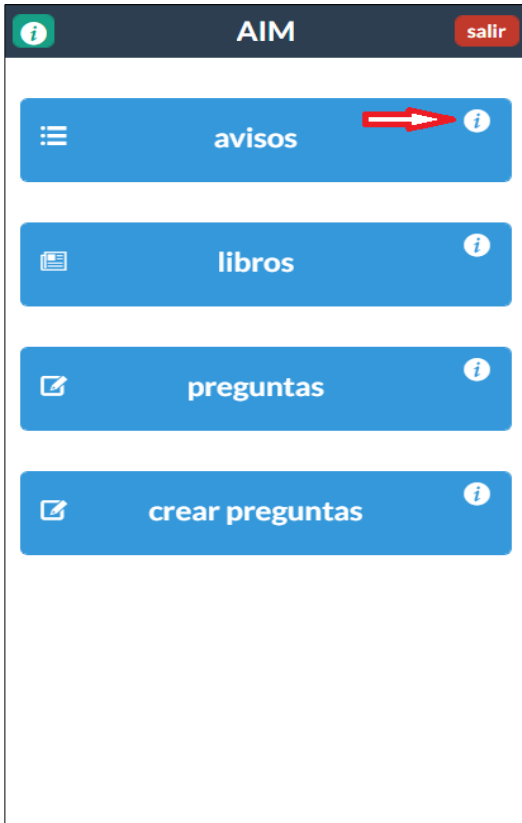


Figura 4.30. Manual de ayuda de la aplicación (versión web).

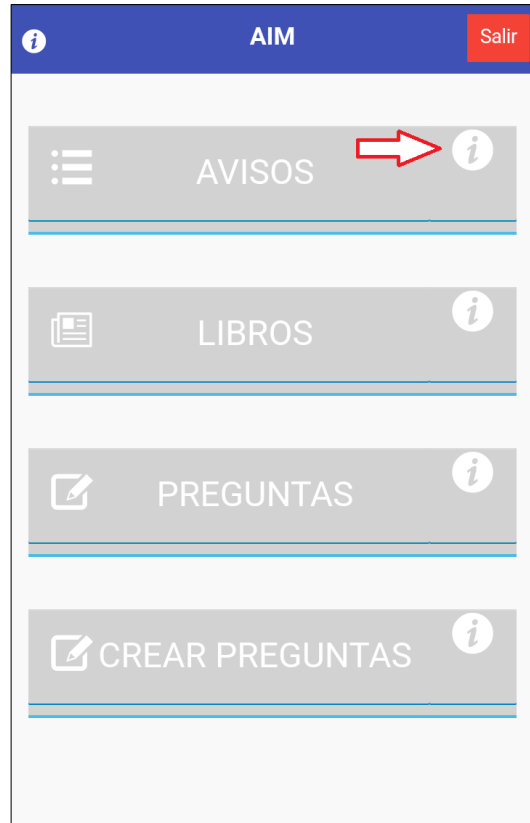


Figura 4.31. Manual de ayuda de la aplicación (versión Android).



Figura 4.32. Ventana emergente de información sobre una funcionalidad principal (versión web).

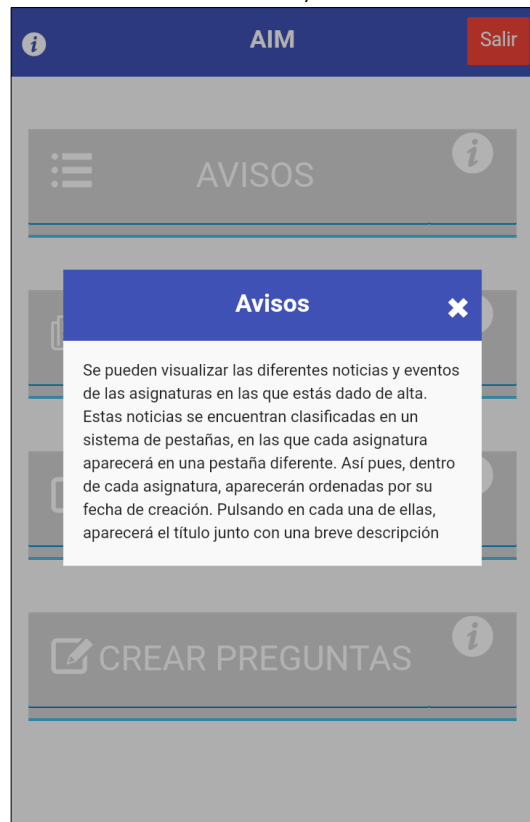


Figura 4.33. Ventana emergente de información sobre una funcionalidad principal (versión Android).

## 4.5. Nuevas funcionalidades en el sistema de evaluación continua

En la funcionalidad principal de preguntas, en la pestaña correspondiente a las notas conseguidas por el alumno a lo largo de la evaluación continua realizada por el profesor a partir de las preguntas lanzadas, en la versión anterior de la aplicación del alumno se mostraba una gráfica con la evolución de la nota media del alumno a lo largo de los días en los que el alumno respondía a las preguntas creadas y lanzadas por el profesor.

Sin embargo, en la versión actual de la aplicación del alumno, se muestra una gráfica con la evolución de la nota diaria a lo largo de los días en los que se ha respondido a las preguntas lanzadas. Es decir, la gráfica no muestra la nota media obtenida hasta la fecha en concreto si no la nota media obtenida en la fecha en concreto escogida, no teniendo en cuenta las fechas anteriores para hallar la nota media acumulada.

Para ello, se ha modificado el fichero «Questions.js» en la ruta «app/controller/» en el proyecto de Sencha Touch y el fichero «QuestionController.java» del paquete Java «es.uva.tel.aimml.api.controller» en el directorio «src/main/java» del proyecto «AIMML-API» en Spring Tool Suite. En concreto, se ha modificado el método «getScoreDashboard».

Además, bajo la gráfica de la evaluación continua del alumno a lo largo de la asignatura, también se muestra una lista con las preguntas respondidas a lo largo del curso académico para la asignatura concreta. En cada pregunta respondida se muestra la fecha y la hora de la respuesta junto al enunciado de la pregunta y debajo, las posibles soluciones de la pregunta. Además, cada solución aparece junto a un aspa si dicha solución es correcta y junto a una cruz si es errónea, resaltando la solución que el alumno seleccionó. En concreto, en verde si seleccionó la solución correcta y en rojo si seleccionó la solución errónea, tal y como se muestra en la Figura 4.34 (versión web) y en la Figura 4.35 (versión móvil).

Esta funcionalidad permite al alumno visualizar las preguntas respondidas y si respondió correcta o erróneamente, observando las soluciones correctas y erróneas de manera visual para continuar con el aprendizaje a lo largo de la asignatura. De este modo se hace más consciente de su aprendizaje y obtiene una retroalimentación a lo largo de la asignatura de forma continua.

Hay que mencionar que al igual que en la gráfica no se visualizarán todas las preguntas respondidas en tiempo real, por motivos de cálculo y homogeneidad con la misma, solo se muestran las respuestas hasta el día anterior. Es decir, si el alumno responde a una o varias preguntas, estas no se mostrarán hasta que no finalice el día.

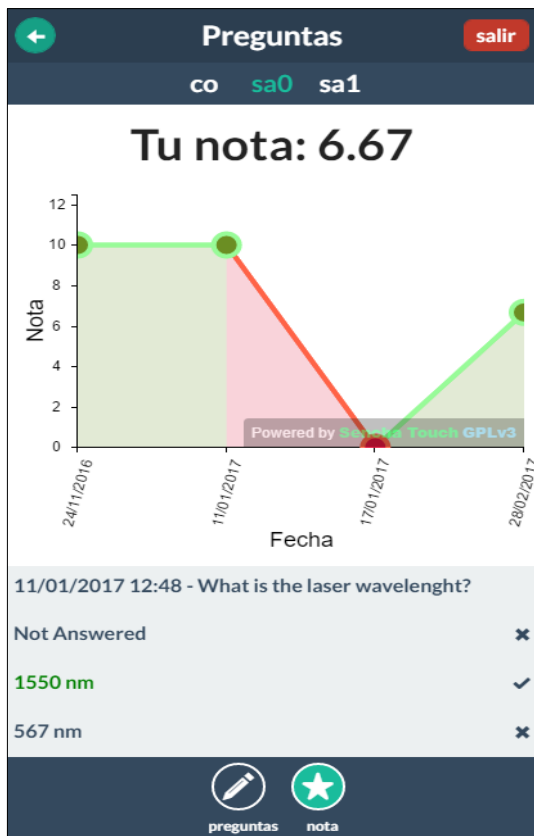


Figura 4.34. Lista de preguntas respondidas (versión web).

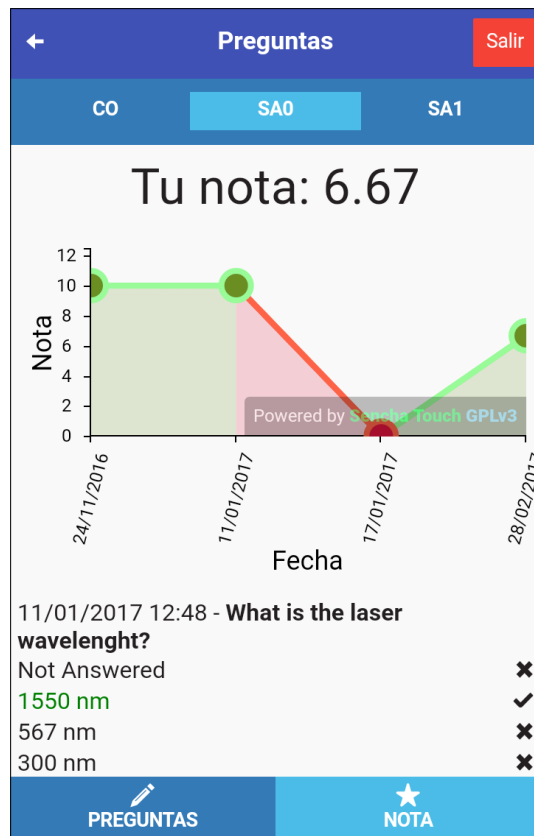


Figura 4.35. Lista de preguntas respondidas (versión Android).

Para implementar esta funcionalidad, se han creado los ficheros «Reply.js» en la ruta «app/model/» y «Replies.js» en la ruta «app/store/» y se ha modificado el fichero «Questions.js» en la ruta «app/controller/», dentro del proyecto de Sencha Touch. Por otro lado, en el proyecto «AIMML-API» de Spring Tool Suite, se ha modificado el fichero «QuestionController.java» del paquete Java «es.uva.tel.aimml.api.controller» en el directorio «src/main/java». En concreto se ha añadido el método «getReplies» que, tras la correspondiente petición a la API, devuelve una lista de objetos en formato JSON con la información asociada a las respuestas. Esta información estará filtrada en el primer fichero mencionado anteriormente para ordenar la lista cronológicamente y mostrar las respuestas del curso académico actual por asignaturas. Además, se han creado los ficheros «ReplyDTO.java» y «RepliedSolutionDTO.java» en el paquete Java «es.uva.tel.aimml.api.dto» en el directorio «src/main/java».

Finalmente, si el usuario de la plataforma del alumno pulsa en cualquier punto destacado de la gráfica, se muestra una ventana emergente con información detallada sobre la nota de la fecha en cuestión junto con la lista de preguntas respondidas (con el mismo formato que en la lista inferior) en la misma fecha como se observa en la Figura 4.36 (versión web) y en la Figura 4.37 (versión móvil).

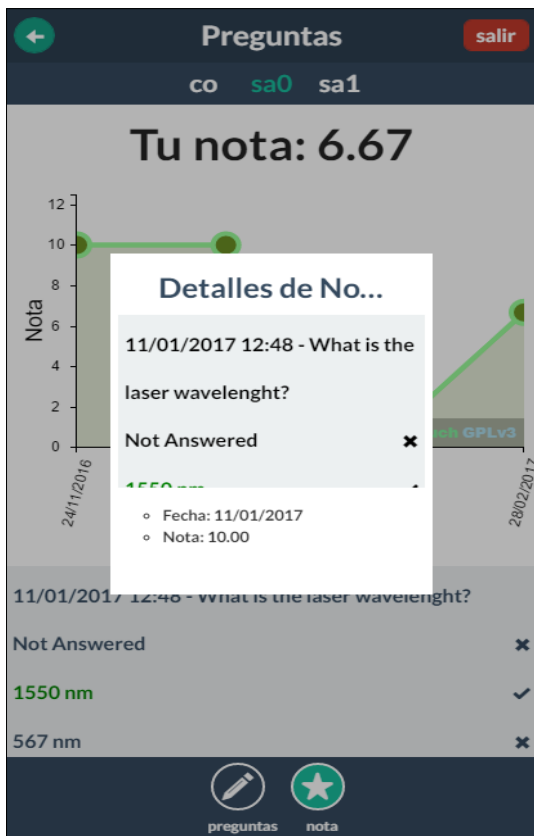


Figura 4.36. Detalles de nota (versión web).



Figura 4.37. Detalles de nota (versión Android).

Para ello, se ha modificado el fichero «Questions.js» en la ruta «app/controller/» en el proyecto de Sencha Touch filtrando de manera más exhaustiva la lista de respuestas obtenidas mediante la petición a la API mencionada anteriormente; es decir, mostrando únicamente las respuestas correspondientes a la fecha en cuestión.

## 4.6. Funcionalidad para crear preguntas

Esta funcionalidad permite crear preguntas desde la aplicación del alumno para que el estudiante participe activamente en el aprendizaje.

Tras pulsar en el botón correspondiente del menú principal en la aplicación del alumno (Figura 4.38 (versión web) y Figura 4.39 (versión Android)), en cada una de las pestañas de las asignaturas en las que el estudiante está matriculado, se muestra un formulario para crear una nueva pregunta y lanzarla.

Se muestra un primer apartado para crear el enunciado de la pregunta, un segundo apartado en el que se muestran las soluciones de la pregunta y un tercer apartado deshabilitado para crear una solución de la pregunta, tal y como se muestra en la Figura 4.40 (versión web) y en la Figura 4.41 (versión Android).

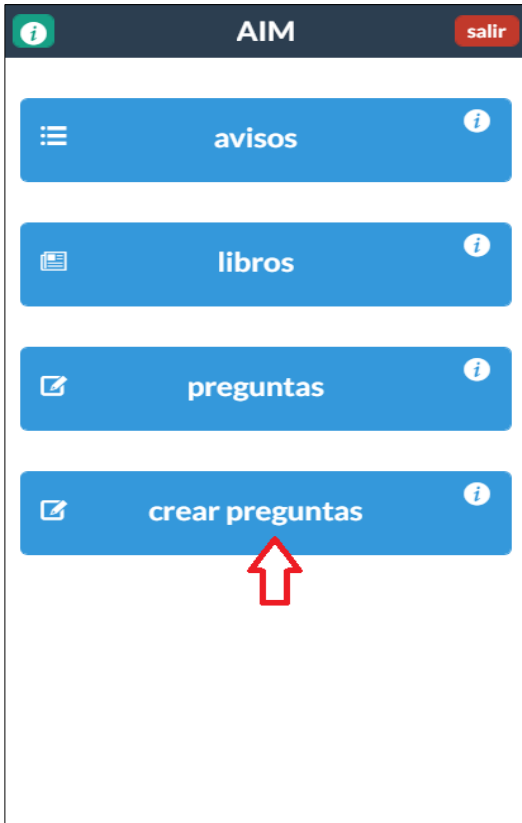


Figura 4.38. Funcionalidad para crear preguntas (versión web).



Figura 4.39. Funcionalidad para crear preguntas (versión Android).

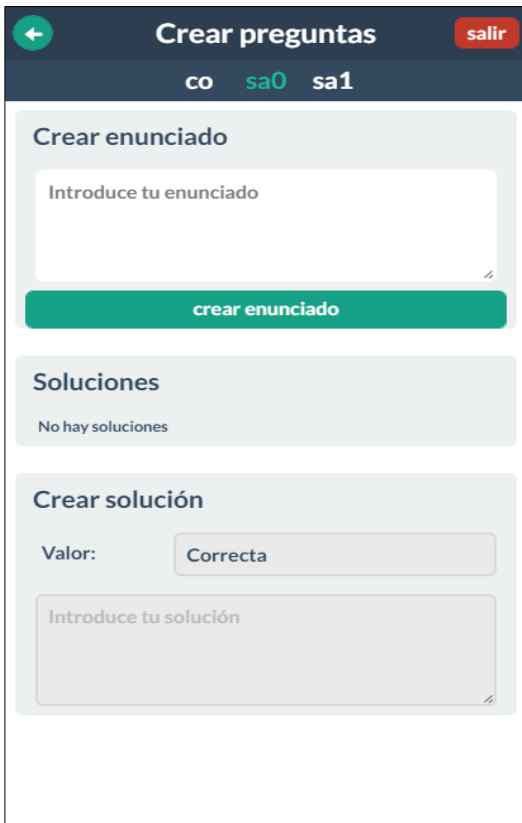


Figura 4.40. Crear pregunta (versión web).

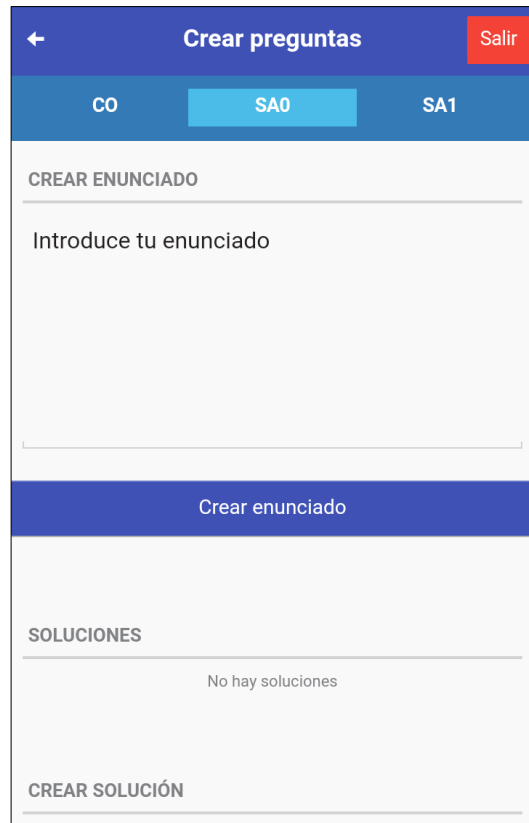


Figura 4.41. Crear pregunta (versión Android).

Tal y como se observa, en primer lugar, se introduce el enunciado en la caja de texto del apartado de creación de enunciado y se pulsa el botón «crear enunciado» del mismo apartado. Tras pulsar este botón se deshabilita el apartado de creación de enunciado y se oculta el botón de este apartado. Además, se habilita el apartado de creación de las soluciones tal y como se puede ver en la Figura 4.42 (versión web) y en la Figura 4.43 (versión Android). En el caso de querer modificar el enunciado introducido se tiene que salir de la funcionalidad de crear pregunta y volver a entrar de nuevo.

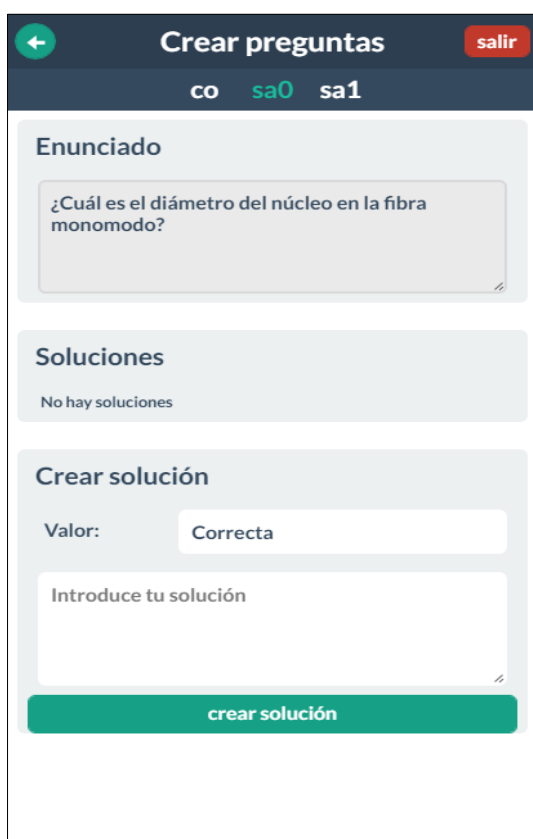


Figura 4.42. Enunciado creado (versión web).

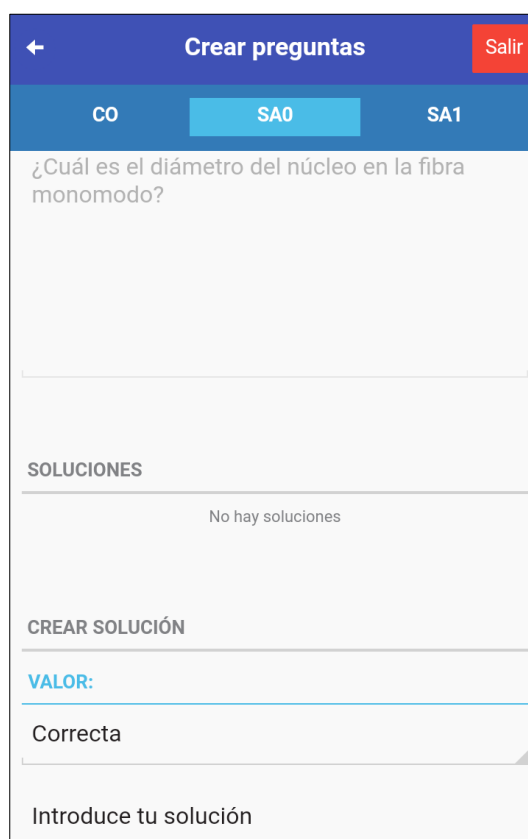


Figura 4.43. Enunciado creado (versión Android).

En segundo lugar, se debe crear el enunciado de la primera solución en la caja de texto correspondiente del apartado de creación de solución junto a su valor, correcta o errónea, y se pulsa el botón del mismo apartado. Tras pulsar este botón se visualiza la solución introducida junto a su valor en el apartado de soluciones (Figura 4.44 (versión web) y Figura 4.45 (versión Android)).

Cabe destacar que no se pueden introducir dos soluciones iguales y si se quiere modificar la solución introducida se tiene que salir de la funcionalidad de crear preguntas y volver a entrar de nuevo. Finalmente, tras introducir como mínimo dos soluciones, se muestra un cuarto apartado destinado a crear la publicación, es decir, lanzarla al resto de alumnos. En dicho apartado el alumno deberá rellenar el nombre de la publicación junto con la fecha de activación y la duración en activo (en minutos) de la pregunta. Finalmente, aparece un último botón de creación de la publicación («Crear pregunta»), como se puede ver en la Figura 4.46 (versión web) y en la Figura 4.47 (versión Android), momento en el cual se enviará la pregunta a la base de datos.

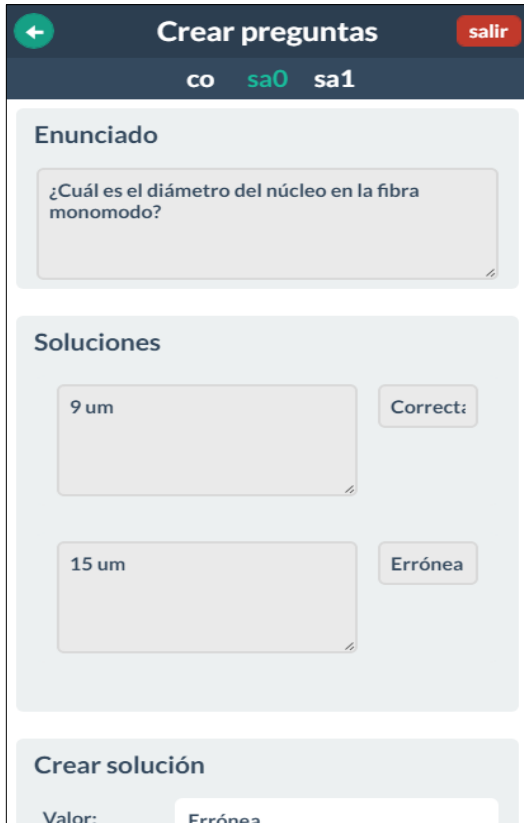


Figura 4.44. Solución creada (versión web).

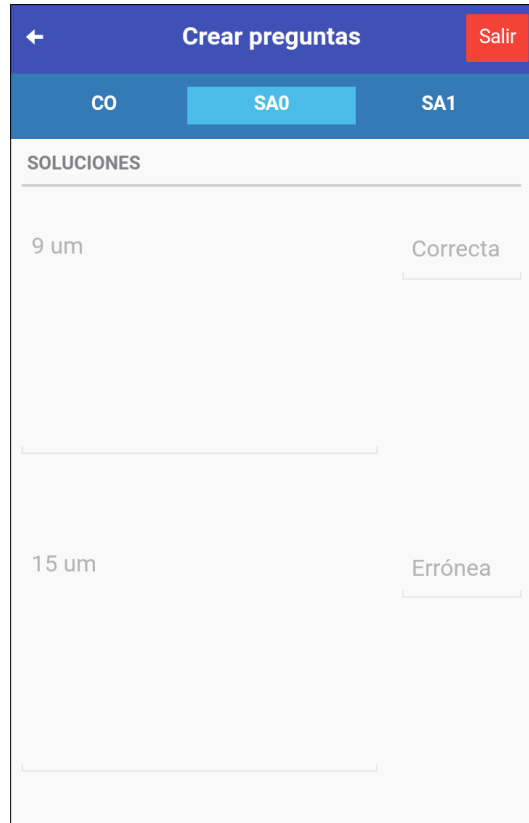


Figura 4.45. Solución creada (versión Android).



Figura 4.46. Creación de publicación y pregunta (versión web).

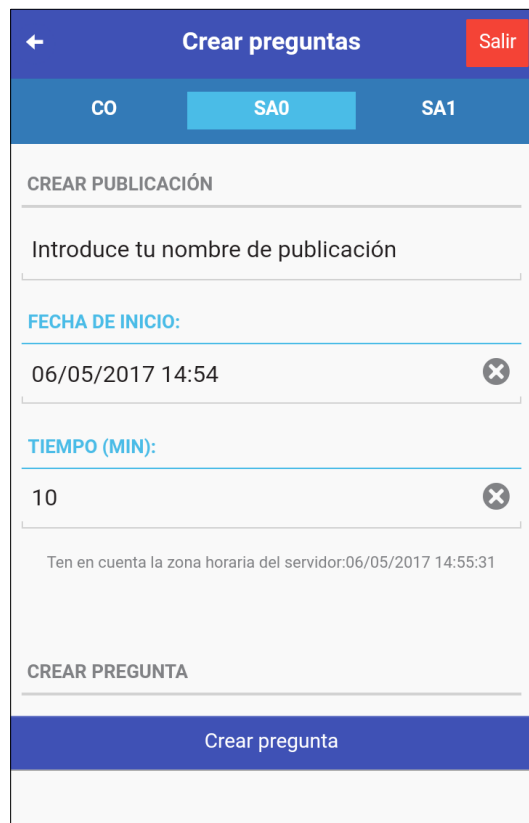


Figura 4.47. Creación de publicación y pregunta (versión Android).

Cabe destacar que se debe introducir de forma obligatoria el nombre de la publicación en la caja de texto del apartado de creación de publicación. A continuación, se deberá introducir o verificar la fecha y hora de inicio en la segunda caja de texto del mismo apartado y el tiempo en minutos en la tercera caja de texto del mismo apartado.

Se tiene que introducir una fecha y una hora con el mismo formato que se muestra en el cuarto apartado de creación de publicación. Además, se tiene que introducir una fecha y una hora correspondientes a un momento posterior al tiempo actual del servidor web de aplicaciones en el que se despliega la API y pertenecientes al curso académico en curso. En este sentido, en la parte inferior de este apartado, se muestra la fecha y la hora del servidor web de aplicaciones. También se tiene que introducir un tiempo en minutos positivo sin valores decimales.

En último lugar, se debe pulsar el botón del apartado final para lanzar la pregunta y enviarla a la base de datos. Tras pulsar este botón y comprobar que los valores introducidos en las cajas de texto tienen el formato correcto, se envía una petición a la API con un objeto JSON con los valores introducidos en los campos del formulario. En caso de éxito, se muestra un mensaje de alerta a través de una ventana emergente que informa al estudiante, como se aprecia en la Figura 4.48 (versión web) y Figura 4.49 (versión Android). Además, la publicación creada se puede visualizar, tanto en la aplicación del profesor como en la aplicación del alumno, en el momento y con la duración configurados.



Figura 4.48. Publicación creada (versión web).



Figura 4.49. Publicación creada (versión Android).



De forma transparente al alumno, al nombre de publicación creado se le añade el nombre y los apellidos o el DNI del alumno que ha creado la publicación, tal y como se observa en la parte izquierda de la Figura 4.50. Esto permite al profesor, identificar al alumno que generó la publicación, tanto para bien como para mal. Además, en última instancia, el profesor siempre puede eliminar la publicación de la pregunta creada si así lo desea (Figura 4.50).



**Figura 4.50.** Autoría y opción de borrado de la publicación creada desde la plataforma del alumno en la plataforma del profesor.

Para desarrollar esta funcionalidad se han creado los ficheros «MyQuestions.js», «MyQuestion.js» en la ruta «app/view/», además del fichero «MyQuestions.js» en la ruta «app/controller/», el fichero «Offset.js» en la ruta «app/model/» y finalmente el fichero «Offsets.js» en la ruta «app/store/» (en el proyecto de Sencha Touch). Los dos primeros se asocian a la interfaz de usuario de esta funcionalidad principal, el tercero controla la lógica, y el cuarto y el quinto se asocian al *offset* horario respecto a GMT (*Greenwich Mean Time*, Tiempo Medio de Greenwich) del servidor web de aplicaciones y permite almacenar localmente dicho *offset*.

En el proyecto «AIMML-API» en Spring Tool Suite, se ha modificado el fichero «QuestionController.java» en el paquete Java «es.uva.tel.aimml.api.controller» del directorio «src/main/java». En concreto, se han añadido los métodos «createQuestions» y «getTime». El primero está relacionado con la petición a la API para crear preguntas desde la plataforma del alumno y el segundo está relacionado con la petición a la API para obtener el *offset* horario del servidor web de aplicaciones. Además, se han creado los ficheros «MyQuestionDTO.java» y «MySolutionDTO.java» en el paquete Java «es.uva.tel.aimml.api.dto» del mismo directorio relacionados con la petición a la API para crear preguntas desde la plataforma del alumno.

## 4.7. Otras modificaciones

En este apartado se describen algunas de las pequeñas modificaciones realizadas en el desarrollo de la plataforma del alumno.

- Se ha modificado el fichero «app/view/Info.js» en el proyecto en Sencha Touch para modificar y añadir algunas cadenas de texto en la funcionalidad que muestra un manual de ayuda de la aplicación del alumno. Principalmente, se ha modificado la versión de desarrollo *software* de la plataforma del

alumno, se han añadido los nuevos desarrolladores *software* de la plataforma y los navegadores web soportados por la aplicación para disfrutar de una mejor visualización y funcionamiento de esta.

- Se ha modificado el fichero «app/util/I18n.js» en el proyecto en Sencha Touch, modificando y añadiendo cadenas de texto tanto en inglés como en español para según el idioma del navegador web, o del dispositivo móvil, se muestren en un idioma o en otro las cadenas de texto en las vistas de la aplicación del alumno.
- Se han modificado algunos ficheros en el proyecto en Sencha Touch en los que aparece código fuente relativo al método que muestra mensajes de alerta en la interfaz de usuario. Se ha reemplazado por otro método que realiza la misma función, pero menos propenso a errores de ejecución debido al comportamiento asíncrono de ambos. Además, tiene la opción de personalizar las cadenas de texto que se muestran en los botones de los mensajes de alerta.
- Se ha modificado el fichero «app/util/CheckLogin.js» en el proyecto en Sencha Touch debido al cambio de flujo en el inicio de la aplicación de la plataforma del alumno por la inclusión de la funcionalidad de inicio de conexión con la API.
- Se ha creado el fichero «app/útil/FlexBox.js» en el proyecto en Sencha Touch en el que se define una clase que sobrescribe la clase «Ext.layout.FlexBox». El objetivo es mejorar el comportamiento de las vistas en las que se utiliza la propiedad CSS «flex» en navegadores web que no tienen un motor de «renderizado» Webkit, tales como Firefox.
- Se ha intentado documentar el código fuente, tanto el de versiones anteriores como el relativo al desarrollo relacionado con este TFM.

## 4.8. Conclusiones

En el Capítulo 4 se han descrito las nuevas funcionalidades desarrolladas en la aplicación del alumno, a raíz de este TFM. El desarrollo de estas nuevas funcionalidades ha sido el núcleo de este TFM, son el fin tangible de un proyecto académico con el que también se adquieren otras competencias transversales.

Entre estas nuevas funcionalidades hay que destacar el desarrollo de la funcionalidad de inicio de conexión con la API y de la de crear preguntas por parte del alumno. Con la primera funcionalidad se abre la posibilidad de que cualquier institución educativa pueda alojar el entorno virtual de aprendizaje virtual en sus propios servidores, personalizando su uso y dotando de mayor seguridad el acceso externo al mismo. Con la segunda funcionalidad se abre la posibilidad de que el estudiante participe activamente en el aprendizaje de cada una de las asignaturas creando sus propias preguntas. De este modo, podrá afianzar conceptos y colaborar con otros estudiantes. Además, podrá poner a prueba sus conocimientos de una forma continua a lo largo de la asignatura.

También se han resuelto pequeños problemas que había en versiones anteriores, siempre teniendo en cuenta que el *software* está en continua evolución, tanto el *framework* Sencha Touch como los diferentes navegadores web y sistemas operativos móviles.



# 5

## Conclusiones

### 5.1. Introducción

En este último Capítulo 5 se describen las principales conclusiones extraídas sobre este TFM, así como los posibles trabajos futuros a realizar sobre el entorno virtual de aprendizaje AIM-Mobile Learning Platform para ampliar sus funcionalidades y mejorar su funcionamiento en futuras versiones. Estas mejoras se podrán realizar tanto en la aplicación del alumno como en la aplicación servidora, así como en la API y en la base de datos.

### 5.2. Conclusiones

En este TFM se ha desarrollado una nueva versión *software* del entorno virtual de aprendizaje AIM-Mobile Learning Platform, a partir de su versión 3.1.4, ampliando principalmente las funcionalidades de la aplicación del alumno desarrollada con la ayuda del *framework* Sencha Touch y Apache Cordova. Sin embargo, también se ha modificado pequeñas partes del código fuente de la aplicación servidora, principalmente en la API (desarrollada con la ayuda de Spring Framework), relacionadas con la aplicación cliente mediante peticiones y respuestas a través de Internet como cualquier servicio electrónico.

También se han realizado pequeñas tareas relacionadas con la etapa de mantenimiento y actualización del ciclo de vida *software* de la plataforma de *e-learning*. Tareas destinadas a resolver los problemas derivados de las actualizaciones producidas en las herramientas de desarrollo como los *framework*, sistemas operativos o navegadores empleados.

Para ello, en primer lugar, se implementó la plataforma de *e-learning* AIM-Mobile Learning Platform en un entorno de pruebas local en Windows 10 con la ayuda de Spring Tool Suite, Sublime Text, Sencha Touch, XAMPP y Chrome principalmente. En segundo lugar, con la ayuda de guías, tutoriales, ejemplos, referencias de los sitios web oficiales de las herramientas de desarrollo empleadas y con los comentarios de desarrolladores *software* en foros web, se implementó el código necesario para cumplir con los requisitos de diseño fijados en conjunción con la tutora de este TFM para implementar las nuevas funcionalidades.

En último lugar, se trasladó lo desarrollado al entorno de producción, alojado en un servidor de la ETSIT (Escuela Técnica Superior de Ingenieros de Telecomunicación), tanto los ficheros a desplegar mediante aplicaciones web como el código fuente, usando para ello un sistema de control de versiones en un repositorio remoto en otro servidor de la ETSIT. Mediante ensayos prueba-error en la etapa de depuración del *software* se han ido resolviendo pequeños problemas que iban surgiendo, sobre todo los relacionados con las nuevas funcionalidades desarrolladas en este TFM para finalmente obtener una nueva versión de AIM-Mobile Learning Platform plenamente operativa siguiendo los principios con los que se inició su desarrollo en proyectos universitarios anteriores. Estos principios básicos se centran principalmente en la modularidad, sencillez, rapidez y usabilidad, tanto en su versión web como en su aplicación nativa, en este caso para Android, cumpliendo así el objetivo general y los objetivos específicos descritos al inicio de este TFM.

Una de las principales tareas se centró también en documentar de la mejor manera posible el código fuente para ayudar a futuros desarrolladores, puesto que cómo casi cualquier proyecto de desarrollo *software*, en él intervienen e intervendrán varios desarrolladores a lo largo del ciclo de vida del desarrollo *software* del entorno virtual de aprendizaje AIM-Mobile Learning Platform.

De esta forma, se han desarrollado varias nuevas funcionalidades destinadas a mejorar la relación de la aplicación del alumno con la aplicación servidora y, por tanto, a mejorar la experiencia del alumno en el proceso de aprendizaje. En concreto, se ha desarrollado un nuevo flujo de conexión con el servidor de la aplicación posibilitando que cada centro educativo puede desplegar su propio entorno virtual de aprendizaje AIM-Mobile Learning Platform, con su propia base de datos, dotándolo de una mayor flexibilidad y encapsulamiento de la información. También se ha desarrollado la opción de obtener una nueva contraseña a través del *email* en caso de que el alumno olvidó su contraseña de acceso a la aplicación.

Se puede decir que la principal funcionalidad desarrollada en este TFM es la posibilidad de que el alumno puede crear sus propias preguntas desde su aplicación y compartirlas con el resto de alumnos mejorando de este modo notablemente el aprendizaje puesto que el nivel de implicación del alumno en el proceso aumenta. Además, en la visualización de las notas en el sistema de evaluación continua, se ha desarrollado una lista, bajo la gráfica, en la que el alumno puede ver las preguntas que ha ido respondiendo a lo largo del curso (incluido su detalle en ventanas emergentes pulsando en los puntos destacados en la gráfica) mejorando también notablemente el proceso de aprendizaje puesto que de este modo es capaz de observar de una manera más consciente sus errores para aprender de ellos y sus aciertos para afianzar los conocimientos adquiridos.

En definitiva, se han desarrollado una serie de funcionalidades de interés para el alumno que mejoran y resuelven los problemas a los que se enfrentaba en las versiones previas del entorno virtual de aprendizaje AIM-Mobile Learning Platform cumpliendo así los objetivos de este TFM.

## 5.3. Posibles trabajos futuros

En este apartado se describen brevemente algunos posibles trabajos futuros a desarrollar en el entorno virtual de aprendizaje AIM-Mobile Learning Platform para cubrir diferentes casos de uso según los comentarios de alumnos y profesores sobre la nueva versión del entorno:

- **Modificar contraseña:** al igual que en la versión previa del entorno virtual de aprendizaje, se puede modificar la contraseña del usuario, tanto de la plataforma del alumno como de la aplicación del profesor, desde la propia aplicación del profesor. Además, en esta nueva versión del entorno virtual de aprendizaje se puede re-inicializar la contraseña del usuario de la plataforma del alumno desde esta plataforma, es decir, si un usuario de la plataforma del alumno olvida su contraseña puede solicitar el envío de un *email* con su nueva contraseña. Sin embargo, este envío sólo es posible desde el servidor de la ETSIT. Así pues, en un posible trabajo futuro se podría crear un *script* de instalación y configuración de un servidor de correo saliente para incluirlo en el instalador del entorno virtual o también se podría mostrar una opción de configuración en la aplicación del profesor para que el administrador de la plataforma indique el servidor de correo saliente. En otro posible trabajo futuro relacionado con la modificación de la contraseña del usuario alumno se podría añadir un menú de ajustes en la aplicación para que el alumno pudiera modificar su contraseña a su gusto como la primera vez que se registró en la plataforma (en esta versión se envía una contraseña aleatoria generada por el entorno virtual de aprendizaje).
- **Revisar preguntas creadas por los alumnos:** en esta nueva versión del entorno virtual de aprendizaje un alumno puede crear preguntas para comprender y afianzar conocimientos generando sus propios contenidos. Sin embargo, esta nueva funcionalidad hace perder el control del contenido generado al profesor debido a la creación en tiempo real de las nuevas preguntas, es decir, un profesor puede eliminar nuevas publicaciones, pero si se generan al instante no podría evitarlo. Así pues, en un posible trabajo futuro se podría revisar el proceso de creación de preguntas desde la plataforma del profesor, por ejemplo, evitando la creación instantánea de preguntas para que el profesor desde un área específica en su aplicación pueda revisar las futuras publicaciones modificándolas en caso de erratas o eliminándolas en el peor caso. Además, se podría crear una relación entre la entidad «Student» y «Question» para gestionar mejor el autor de la nueva pregunta.
- **Mensajería entre alumnos y profesores:** en otro posible trabajo futuro se podría desarrollar un sistema de mensajería entre profesores y alumnos rápido y sencillo mediante la creación de buzones de entrada y salida tanto en las aplicaciones del alumno y del profesor para abrir un canal de comunicación en tiempo real entre el alumno y el profesor al estilo de la popular aplicación móvil de mensajería Whatsapp. Por ejemplo, se podría crear una nueva tabla en la base de datos con el emisor, receptor, fecha y

contenido del mensaje a consultar por parte de la aplicación cliente y servidora cada vez que un usuario inicie sesión en su respectiva plataforma.

- **Notificaciones:** al hilo del posible trabajo futuro anterior se podría crear un sistema de notificaciones para recordar al usuario en la plataforma del alumno eventos de carácter importante como fechas de examen, entregas de prácticas, etc. mediante notificaciones de tipo *push*, es decir, avisos generados y enviados desde un servidor remoto con entrega inmediata en la aplicación cliente o también almacenados por un tiempo hasta su envío, para notificar al alumno dicho evento importante cuanto antes en su dispositivo móvil.
- **Calendario:** también se podría crear una nueva funcionalidad en la aplicación del alumno en la que visualizar gráficamente en un calendario los eventos importantes en conjunción con formatos ampliamente utilizado como iCalendar o Google Calendar.
- **Formato de los ficheros CSV con las listas de estudiantes:** en esta nueva versión del entorno virtual de aprendizaje, al añadir listas de estudiantes a una asignatura en la plataforma del profesor, estas se subían en ficheros XLS o XLSX y en ficheros CSV. Sin embargo, los ficheros CSV tienen que tener un formato adecuado y dependiendo de la zona geográfica del sistema operativo en el que se ejecuta la subida, los valores de este tipo de ficheros se separan mediante un carácter (por defecto, la coma) u otro. Así pues, en un posible trabajo futuro se podría desarrollar una nueva funcionalidad en la aplicación del profesor en la que se pueden subir ficheros CSV independientemente del carácter separador de valores, flexibilizando el proceso de subida de alumnos. Para ello, entre las posibles opciones, habría que modificar el código de la plataforma para hacerlo de forma transparente al usuario, o también se podría desarrollar un menú desplegable en la aplicación del profesor en la que el usuario pudiera elegir el carácter separador de valores que utiliza el fichero CSV que va a utilizar para añadir la lista de estudiantes a la asignatura.
- **Subida de ficheros de notas:** en la versión actual de la plataforma se permite realizar una evaluación continua de los alumnos a través de las respuestas a las preguntas publicadas y mediante la asignación de positivos o negativos. Sin embargo, en un posible trabajo futuro se podría desarrollar una nueva funcionalidad que permita a los usuarios de la plataforma del profesor subir ficheros con información adicionales sobre la evaluación de los alumnos como notas de trabajos, exámenes parciales o globales. De este modo se podría centralizar en el mismo sistema las calificaciones de los alumnos en el entorno virtual de aprendizaje, mejorando así su gestión.
- **Inicios de sesión como alumno invitado:** se podría plantear el desarrollo de una nueva funcionalidad en la aplicación del alumno que permitiera iniciar sesión en ciertas funcionalidades de esta, pero de forma temporal. Por ejemplo, para que solo ciertos alumnos participen en concursos de preguntas en tiempo real al estilo del sitio web Kahoot! [52].



- **Hora del servidor transparente al usuario:** a la hora de crear preguntas desde la aplicación del alumno, el alumno tiene que ser consciente de la hora al instante en el servidor donde se aloja la aplicación servidora puesto que la fecha y la hora de creación serán la hora regente en dicho servidor. En un posible trabajo futuro se podrían unificar todos los campos que contengan fechas y horas en el entorno virtual de aprendizaje estudiando cómo se trata este tipo de problemas en las aplicaciones web que trabajan con localizaciones de aplicación cliente y servidora situadas en diferentes zonas horarias.
- **Detección automática de la nueva versión:** se ha observado que, al actualizar el entorno virtual de aprendizaje, no se detecta de inmediato la nueva versión, sino que es necesario borrar la caché del navegador web o los datos de la aplicación nativa en un dispositivo móvil. Así pues, en otro posible trabajo futuro se podría estudiar cual es la mejor forma de desarrollar esta detección automática.



# Bibliografía

- [1] D. González Ortega, «Implementación de servicios sobre Redes de Nueva Generación». Universidad de Valladolid, 01-mar-2016.
- [2] J. Debrán Pérez, «Diseño e implementación del entorno de aprendizaje virtual AIM-Mobile Learning Platform». Universidad de Valladolid, 10-sep-2015.
- [3] Á. San Jose Cabezas, «Diseño e implementación de nuevas funcionalidades a una plataforma de aprendizaje web interactiva». Universidad de Valladolid, 28-abr-2015.
- [4] G. Goyanes San Frutos, «AIM-Mobile Learning Platform - Documentación para desarrolladores». Universidad de Valladolid, 01-jul-2016.
- [5] M. Rodríguez Cayetano, «Introducción al Lenguaje Java». Universidad de Valladolid, 01-mar-2016.
- [6] M. Antón Rodríguez y M. Á. Pérez Juárez, «HTML, CSS y JavaScript». Universidad de Valladolid, 01-sep-2006.
- [7] «W3Schools Online Web Tutorials». [En línea]. Disponible en: <https://www.w3schools.com/>. [Accedido: 05-may-2017].
- [8] «spring.io». [En línea]. Disponible en: <https://spring.io/>. [Accedido: 05-may-2017].
- [9] «Docs | Sencha Touch 2.4», *Sencha Documentation*. [En línea]. Disponible en: <http://docs.sencha.com/touchindex.html>. [Accedido: 05-may-2017].
- [10] L. Boonstra, *Hands-On Sencha Touch 2*. O'Reilly, 2014.
- [11] «Documentation - Apache Cordova». [En línea]. Disponible en: <https://cordova.apache.org/docs/en/latest/>. [Accedido: 05-may-2017].
- [12] «Git». [En línea]. Disponible en: <https://git-scm.com/>. [Accedido: 05-may-2017].
- [13] «java.com: Java y Tú». [En línea]. Disponible en: <https://www.java.com/es/>. [Accedido: 05-may-2017].
- [14] «About the Java Technology (The Java™ Tutorials > Getting Started > The Java Technology Phenomenon)». [En línea]. Disponible en: <http://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>. [Accedido: 05-may-2017].
- [15] «Spring Framework». [En línea]. Disponible en: <http://projects.spring.io/spring-framework/>. [Accedido: 05-may-2017].
- [16] «Spring Data JPA». [En línea]. Disponible en: <http://projects.spring.io/spring-data-jpa/>. [Accedido: 05-may-2017].
- [17] «Spring Data». [En línea]. Disponible en: <http://projects.spring.io/spring-data/>. [Accedido: 05-may-2017].

- [18] «Spring Security». [En línea]. Disponible en: <https://projects.spring.io/spring-security/>. [Accedido: 05-may-2017].
- [19] «W3C HTML». [En línea]. Disponible en: <https://www.w3.org/html/>. [Accedido: 05-may-2017].
- [20] «Cascading Style Sheets». [En línea]. Disponible en: <https://www.w3.org/Style/CSS/>. [Accedido: 05-may-2017].
- [21] «Thymeleaf». [En línea]. Disponible en: <http://www.thymeleaf.org/>. [Accedido: 05-may-2017].
- [22] «Ecosystem - Thymeleaf». [En línea]. Disponible en: <http://www.thymeleaf.org/ecosystem.html#thymeleaf-spring-security>. [Accedido: 05-may-2017].
- [23] «Bootstrap · The world's most popular mobile-first and responsive front-end framework.» [En línea]. Disponible en: <http://getbootstrap.com/>. [Accedido: 05-may-2017].
- [24] jQuery F.- [jquery.org](http://jquery.org), «jQuery». .
- [25] «Font Awesome, the iconic font and CSS toolkit». [En línea]. Disponible en: <http://fontawesome.io/>. [Accedido: 05-may-2017].
- [26] «Bootstrap 3 Tutorial». [En línea]. Disponible en: <https://www.w3schools.com/bootstrap/>. [Accedido: 05-may-2017].
- [27] «Hibernate ORM - Hibernate ORM». [En línea]. Disponible en: <http://hibernate.org/orm/>. [Accedido: 05-may-2017].
- [28] «c3p0-v0.9.5.2 - JDBC3 Connection and Statement Pooling - Documentation». [En línea]. Disponible en: <http://www.mchange.com/projects/c3p0/>. [Accedido: 05-may-2017].
- [29] T. your name here, «The AspectJ Project». [En línea]. Disponible en: <http://www.eclipse.org/aspectj/>. [Accedido: 05-may-2017].
- [30] «Log4j – Apache Log4j 2 - Apache Log4j 2». [En línea]. Disponible en: <https://logging.apache.org/log4j/2.x/>. [Accedido: 05-may-2017].
- [31] «SLF4J». [En línea]. Disponible en: <https://www.slf4j.org/>. [Accedido: 05-may-2017].
- [32] «Sencha Touch», *Sencha*, 27-mar-2015. [En línea]. Disponible en: <https://www.sencha.com/products/touch/>. [Accedido: 05-may-2017].
- [33] «All About Applications | Sencha Touch 2.4», *Sencha Documentation*. [En línea]. Disponible en: [http://docs.sencha.com/touchcore\\_concepts/about\\_applications.html](http://docs.sencha.com/touchcore_concepts/about_applications.html). [Accedido: 05-may-2017].
- [34] «Apache Cordova». [En línea]. Disponible en: <https://cordova.apache.org/>. [Accedido: 05-may-2017].

- [35] «Standard ECMA-404». [En línea]. Disponible en: <https://www.ecma-international.org/publications/standards/Ecma-404.htm>. [Accedido: 05-may-2017].
- [36] «Spring Tools». [En línea]. Disponible en: <https://spring.io/tools>. [Accedido: 05-may-2017].
- [37] «Sublime Text: The text editor you'll fall in love with». [En línea]. Disponible en: <https://www.sublimetext.com/>. [Accedido: 05-may-2017].
- [38] <https://github.com/summernote>, «Summernote - Super Simple WYSIWYG editor», *Summernote - Super Simple WYSIWYG editor*. [En línea]. Disponible en: <http://summernote.org/>. [Accedido: 05-may-2017].
- [39] «Google Code Archive - Long-term storage for Google Code Project Hosting.» [En línea]. Disponible en: <https://code.google.com/archive/p/flying-saucer/>. [Accedido: 05-may-2017].
- [40] «iText». [En línea]. Disponible en: <http://itextpdf.com/>. [Accedido: 05-may-2017].
- [41] «JTidy - JTidy». [En línea]. Disponible en: <http://jtidy.sourceforge.net/>. [Accedido: 05-may-2017].
- [42] «Apache Commons – Apache Commons». [En línea]. Disponible en: <https://commons.apache.org/>. [Accedido: 05-may-2017].
- [43] «FileUpload – Home». [En línea]. Disponible en: <https://commons.apache.org/proper/commons-fileupload/>. [Accedido: 05-may-2017].
- [44] «Commons IO – Commons IO Overview». [En línea]. Disponible en: <https://commons.apache.org/proper/commons-io/>. [Accedido: 05-may-2017].
- [45] «FFmpeg». [En línea]. Disponible en: <https://ffmpeg.org/>. [Accedido: 05-may-2017].
- [46] «morris.js». [En línea]. Disponible en: <http://morrisjs.github.io/morris.js/>. [Accedido: 05-may-2017].
- [47] «Apache POI - the Java API for Microsoft Documents». [En línea]. Disponible en: <https://poi.apache.org/>. [Accedido: 05-may-2017].
- [48] «Create circular SVG charts with Circles : Lugo Labs». [En línea]. Disponible en: <http://www.lugolabs.com/circles>. [Accedido: 05-may-2017].
- [49] M. Piattini y A. de Miguel Castaño, *Fundamentos y modelos de bases de datos*. RA-MA, 1999.
- [50] C. J. Date, *Introducción a los sistemas de bases de datos*. Pearson Education, 2001.
- [51] «MariaDB.org», *MariaDB.org*. [En línea]. Disponible en: <https://mariadb.org/>. [Accedido: 05-may-2017].
- [52] «Kahoot!» [En línea]. Disponible en: <https://kahoot.it/#/>. [Accedido: 05-may-2017].