



Universidad
Carlos III de Madrid

Departamento de informática
Grado Ingeniería Informática

TRABAJO FIN DE GRADO

**APLICACIÓN MÓVIL BASADA EN
REDES SOCIALES PARA DAR
SOPORTE A LA REALIZACIÓN DE
EJERCICIOS EN CLASE**

Autor: Álvaro Mateos Jiménez

Tutor: Alberto Heredia García, Javier García Guzmán

Leganés, octubre 2013

Título: Aplicación móvil basada en redes sociales para dar soporte a la realización de ejercicios en clase.

Autor: Álvaro Mateos Jiménez

Director: Alberto Heredia García

EL TRIBUNAL

Presidente: Juan Manuel Estévez Tapiador

Vocal: Harith Taha Abdulla Aljumaily

Secretario: Damaris Fuentes Lorenzo

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día **08 de octubre de 2013** en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de Madrid, acuerda otorgarle la CALIFICACIÓN de

SECRETARIO

VOCAL

PRESIDENTE

Resumen

Este trabajo de fin de grado ha consistido en el diseño e implementación de una aplicación para tablets, que permite compartir ejercicios de clase de una asignatura entre el profesor y el alumnado, con el fin de complementar los contenidos dados en las clases presenciales.

Debido al gran número de plataformas de aprendizaje online que están disponibles actualmente en el mercado, el objetivo principal de este proyecto es unificar el acceso a estos sistemas desde una misma aplicación independientemente de la plataforma seleccionada aportando transparencia y flexibilidad tanto al profesor como al alumno.

Entre las principales funcionalidades de la aplicación destaca la posibilidad de que el profesor comparta el enunciado de un nuevo ejercicio y que el alumno pueda visualizar el contenido de este y envíe al profesor la solución del mismo. También ofrece al profesor la posibilidad de ver un listado de todos sus alumnos o de los ejercicios que ha compartido. Además, la aplicación permite mostrar qué alumnos han enviado la solución a un ejercicio en concreto, y viceversa, qué ejercicios ha resuelto un alumno determinado.

Destacar por otro lado, que la aplicación hace uso de la identidad personal de la red social de Twitter para autenticar al usuario, permitiendo asociar los datos de acceso de una plataforma de aprendizaje a dicha identidad. Además se ha desarrollado un servicio web que hace de capa intermedia entre la aplicación y los diferentes servidores de contenidos. Mencionar que en este proyecto se ha simulado la parte del LMS mediante un servidor FTP.

Respecto a la metodología de trabajo se ha optado por las metodologías ágiles, haciendo uso de Scrum ya que permite una mayor flexibilidad y facilita la planificación adaptativa y progresiva, basada en desarrollar software estable y funcional en cada iteración. Se ha elegido Android como plataforma móvil y se ha hecho uso del lenguaje de programación Java mediante el entorno de desarrollo Eclipse, que ofrece herramientas de simulación y de depuración del código.

Abstract

This dissertation has consisted of the design and implementation of an application for tablets, that allows sharing classroom exercises from a subject between teacher and student, in order to complement the contents given by the teacher in face-to-face classrooms.

In order to the large number of online learning platforms are available today in the market, the main purpose of this project is unify the access to this kind of systems in an unique application regardless of the selected platform, supplying transparency and flexibility to both teacher and learner.

Among the main functions of the application it is highlighted the possibility of sharing a new exercise definition so that the student can see that content and send to the teacher the solution to that exercise. Also, it offers the possibility of showing a list of students or shared exercise. Furthermore, the application allows to display which students have sent a solution to a concrete exercise, or vice versa, what exercise has been solved by a student.

On the other hand, the application uses the personal identity from social network Twitter to authenticate user. This allows associating the access data from a learning platform to the Twitter identity. Also a web service has been development like a middle layer between the application and different content servers. It is important to mention that LMS part has been simulated using a FTP server.

In relation to work methodology, has been chosen agile methodologies, using Scrum as it allows greater flexibility and facilitate adaptive planning and based on developing progressive, stable and functional software at each iteration. Android has been chosen as a mobile platform and it has made use of the Java programming language using the Eclipse development environment that offers tools for simulation and debugging code.

Contenido

1.	Introducción	14
1.1	Área de conocimiento	14
1.2	Problemática a resolver	15
1.3	Objetivos	15
1.4	Motivación.....	16
1.5	Método de trabajo	17
1.6	Organización de la documentación.....	18
1.7	Glosario de términos	19
1.8	Abreviaturas	20
	Introduction.....	21
	Knowledge area.....	21
	Problem to solve.....	22
	Objectives	22
	Motivation.....	23
	Work methodology.....	24
	Organization of the documentation	25
	Glossary of terms	26
2.	Estado del arte.....	28
2.1	Análisis del mercado.....	28
2.1.1	Teacher Notebook.....	29
2.1.2	Cuaderno del Profesor.....	29
2.1.3	AndroClass-Teacher Tablet.....	30
2.1.4	Blackboard Mobile Learn	31
2.1.5	iDoceo	32
2.1.6	Edmodo	33
2.1.7	Slideshare	34

2.2	Resultados del análisis de mercado	35
2.3	Plataforma de desarrollo	36
2.4	Metodología de desarrollo ágil	40
2.5	Resumen del estado del arte	41
3.	Especificación de requisitos	44
3.1	Requisitos específicos	44
3.2	Casos de uso	48
3.3	Restricciones.....	58
4.	Diseño.....	61
4.1	Prototipo	61
4.2	Alternativas de diseño	70
4.2.1	Código LMS	70
4.2.2	Autenticación Twitter	71
4.2.3	Formato de los datos	72
4.2.4	Estructura de los contenidos.....	73
4.3	Diseño final de la aplicación	74
4.4	Diagramas de clase	81
4.4.1	Aplicación del profesor.....	81
4.4.2	Aplicación del alumno	87
4.5	Diagramas de secuencia	92
4.5.1	Recuperar listado de alumnos.....	92
4.5.2	Recuperar listado de ejercicios	93
4.5.3	Añadir un nuevo ejercicio	94
4.5.4	Añadir cuenta LMS al servicio web	96
4.5.5	Mostrar y cambiar entre cuentas Twitter	97
4.5.6	Mostrar el enunciado de un ejercicio	98
4.5.7	Mostrar la solución de un ejercicio.....	99

4.5.8	Enviar solución de un ejercicio	100
4.5.9	Añadir asignatura de un profesor.....	101
5.	Pruebas	104
5.1	Pruebas unitarias.....	104
5.2	Pruebas de sistema	108
6.	Planificación y presupuesto	119
6.1	Planificación.....	119
6.1.1	Primera iteración.....	119
6.1.2	Segunda iteración	120
6.1.3	Tercera iteración	120
6.1.4	Cuarta iteración	121
6.1.5	Quinta iteración	121
6.1.6	Sexta iteración	122
6.1.7	Séptima iteración	122
6.2	Presupuesto.....	123
7.	Conclusiones y líneas futuras	126
7.1	Conclusiones.....	126
7.2	Líneas futuras	128
	Conclusions and futures researches	129
	Conclusions	129
	Future researches	131
	Bibliografía.....	133

Lista de tablas

Tabla 1: Resultado análisis de mercado	36
Tabla 2: Tabla modelo requisitos	44
Tabla 3: CP001	44
Tabla 4: CP002	44
Tabla 5: CP003	45
Tabla 6: CP004	45
Tabla 7: CP005	45
Tabla 8: CP006	45
Tabla 9: CP007	45
Tabla 10: CP008	45
Tabla 11: CP009	46
Tabla 12: CP010	46
Tabla 13: CP011	46
Tabla 14: CP013	46
Tabla 15: CA001	46
Tabla 16: CA002	47
Tabla 17: CA003	47
Tabla 18: CA004	47
Tabla 19: CA005	47
Tabla 20: CA006	47
Tabla 21: CA007	48
Tabla 22: CA008	48
Tabla 23: CA009	48
Tabla 24: Plantilla caso de uso	50
Tabla 25: Caso de uso Ver lista alumnos	50
Tabla 26: Caso de uso Ver lista ejercicios	51
Tabla 27: Caso de uso Subir un ejercicio	52
Tabla 28: Caso de uso Añadir curso	53
Tabla 29: Caso de uso Cambiar cuenta Twitter	54
Tabla 30: Caso de uso Subir solución	55
Tabla 31: Caso de uso Ver enunciado ejercicio	56
Tabla 32: Caso de uso Ver solución ejercicio	57

Tabla 33: Caso de uso Cambiar asignatura	58
Tabla 34: Plantilla pruebas de sistema	109
Tabla 35: PS001	109
Tabla 36: PS002	110
Tabla 37: PS003	110
Tabla 38: PS004	111
Tabla 39: PS005	112
Tabla 40: PS006	112
Tabla 41: PS007	112
Tabla 42: PS008	113
Tabla 43: PS009	113
Tabla 44: PS010	114
Tabla 45: PS011	114
Tabla 46: PS012	115
Tabla 47: PS013	115
Tabla 48: PS014	116
Tabla 49: PS015	116
Tabla 50: PS016	117
Tabla 51: Tareas de la primera iteración	119
Tabla 52: Tareas de la segunda iteración	120
Tabla 53: Tareas de la tercera iteración	121
Tabla 54: Tareas de la cuarta iteración	121
Tabla 55: Tareas de la quinta iteración	122
Tabla 56: Tareas de la sexta iteración	122
Tabla 57: Tareas de la séptima iteración	123
Tabla 58: Tiempo dedicado al proyecto	123
Tabla 59: Costes de personal	124
Tabla 60: Costes amortizaciones	124
Tabla 61: Coste total del proyecto	124

Lista de ilustraciones

Ilustración 1: Teacher notebook icono	29
Ilustración 2: Teacher notebook interfaz.....	29
Ilustración 3: Cuaderno del profesor icono	30
Ilustración 4: Cuaderno del profesor menú	30
Ilustración 5: Cuaderno del profesor interfaz	30
Ilustración 6: AndroClass-Teacher Tablet icono	31
Ilustración 7: AndroClass-Teacher Tablet interfaz.....	31
Ilustración 8: Blackboard icono	32
Ilustración 9: Blackboard interfaz.....	32
Ilustración 10: Blackboard ejercicio	32
Ilustración 11: iDoceo interfaz	33
Ilustración 12: Edmodo	34
Ilustración 13: Slideshare	35
Ilustración 14: Volumen de mercado sistemas operativos móviles	37
Ilustración 15: Unidades vendidas según sistema operativo	37
Ilustración 16: Arquitectura Android	38
Ilustración 17: Ciclo de vida SCRUM	40
Ilustración 18: Diagrama casos de uso (Profesor)	49
Ilustración 19: Diagrama casos de uso (Alumno).....	49
Ilustración 20: Interfaz principal del profesor (v1).....	62
Ilustración 21: Interfaz iniciar sesión del profesor (v1)	62
Ilustración 22: Interfaz alumno seleccionado del profesor (v1)	63
Ilustración 23: Interfaz para subir un enunciado (v1)	63
Ilustración 24: Interfaz principal del alumno (v1)	64
Ilustración 25: Interfaz para subir una solución (v1)	64
Ilustración 26: Interfaz principal del profesor (v2).....	65
Ilustración 27: Interfaz para seleccionar alumno (v2)	65
Ilustración 28: Interfaz principal del profesor I (v3)	66
Ilustración 29: Interfaz principal del profesor II (v3)	67
Ilustración 30: Interfaz del profesor al seleccionar un enunciado (v3).....	67
Ilustración 31: Interfaz del profesor al seleccionar un alumno (v3)	68

Ilustración 32: Interfaz del profesor para cambiar de cuenta (v3).....	69
Ilustración 33: Interfaz principal del alumno (v3).....	69
Ilustración 34: Autenticación externa de Twitter	71
Ilustración 35: Ejemplo fichero JSON.....	72
Ilustración 36: Ejemplo fichero XML.....	73
Ilustración 37: Diseño final de la aplicación	74
Ilustración 38: Interfaz principal listado de alumnos.....	75
Ilustración 39: Interfaz principal listado de ejercicios	76
Ilustración 40: Interfaz principal alumnos que han resuelto un ejercicio	76
Ilustración 41: Mostrar solución de un alumno	77
Ilustración 42: Cambiar entre identidades de Twitter.....	78
Ilustración 43: Interfaz principal del alumno	78
Ilustración 44: Interfaz para cambiar entre asignaturas	79
Ilustración 45: Interfaz para añadir un enunciado	80
Ilustración 46: Interfaz para añadir una solución	80
Ilustración 47: Diagrama de clases Profesor	81
Ilustración 48: Clase AlumnosSolucionAdapter	82
Ilustración 49: Clase CuentaAdapter	82
Ilustración 50: Clase CuentaUsuario	82
Ilustración 51: Clase AlumnosSolucion.....	83
Ilustración 52: Clase MainActivity	83
Ilustración 53: Clase ListFragment.....	83
Ilustración 54: Clase ContentFragment.....	84
Ilustración 55: Clase Nuevo_enunciado.....	85
Ilustración 56: Clase MyListFragment	86
Ilustración 57: Clase AnadirCursoActivity	87
Ilustración 58: Diagrama de clases Alumno.....	87
Ilustración 59: Clase MainActivity	88
Ilustración 60: Clase EjerciciosSolucionAdapter	88
Ilustración 61: Clase EjercicioSolucion	88
Ilustración 62: Clase Enviar_solucion	89
Ilustración 63: Clase ListaFragment	90
Ilustración 64: Clase ContentFragment.....	90
Ilustración 65: Clase HelperDB.....	91

Ilustración 66: Clase CuentaAdapter	91
Ilustración 67: Clase CuentaUsuario	91
Ilustración 68: Clase AnadirAsignaturaActivity	92
Ilustración 69: Diagrama secuencia Listar alumnos	92
Ilustración 70: Diagrama de secuencia Listar ejercicios.....	93
Ilustración 71: Diagrama de secuencia Añadir enunciado	95
Ilustración 72: Diagrama de secuencia Añadir cuenta LMS	96
Ilustración 73: Diagrama de secuencia Cambiar cuenta Twitter	97
Ilustración 74: Diagrama de secuencia Mostrar enunciado	98
Ilustración 75: Diagrama de secuencia Mostrar solución	99
Ilustración 76: Diagrama de clase Enviar Solución.....	100
Ilustración 77: Diagrama de clase Añadir Asignatura	102
Ilustración 78: Pruebas unitarias NetworkTest (Profesor)	104
Ilustración 79: Pruebas unitarias ListFragmentTest (Profesor).....	105
Ilustración 80: Pruebas unitarias NetworkTest (Alumno)	106
Ilustración 81: Pruebas unitarias ListFragmentTest (Alumno)	107
Ilustración 82: Pruebas unitarias FTPTest	108

CAPITULO 1

1. Introducción

En este primer apartado se van a describir algunos aspectos introductorios sobre el proyecto, como son el área de conocimiento, la problemática a resolver y los objetivos, así como la motivación y la metodología de trabajo que se ha seguido durante el desarrollo del proyecto. Por último, se detalla un listado de abreviaturas y términos y la estructura que va a seguir a lo largo de este documento.

1.1 Área de conocimiento

Este proyecto consiste en realizar una aplicación basada en los sistemas denominados Mobile Personal Learning Environment. Son entornos online que ofrecen un complemento a las clases impartidas por un profesor, en el aprendizaje de cualquier estudiante.

Para entender esta idea hay que definir en primer lugar que son las plataformas LMS (Learning Management System) o entornos virtuales de aprendizaje. Uno de los mejores ejemplos que se puede nombrar al pensar en estos sistemas, sería la plataforma Moodle. Consiste en un sitio web en el que los profesores pueden gestionar sus cursos de forma sencilla e intuitiva de manera que los alumnos puedan acceder a los recursos disponibles por esta tecnología en cualquier momento y a través de cualquier ordenador.

Aunque esta tecnología ha tenido mucha importancia y actualmente está presente en muchos sistemas educativos, en algunos contextos, las posibilidades que ofrece no cumplen determinadas condiciones requeridas. Es por ello, que han evolucionado a una tecnología donde la movilidad es transparente para el usuario, y los recursos a los que da soporte se estructuran de una forma personalizada siguiendo los principios propios de la denominada web 2.0.

Por ello, el Mobile Personal Learning Environment representa un cambio del uso de las tecnologías en el campo de la educación, permitiendo a los alumnos elegir la forma de aprendizaje y dando la posibilidad de acceder a recursos disponibles en

diferentes fuentes. Esto ha permitido una evolución de estos entornos, en especial por la influencia que los teléfonos inteligentes y las tablets han tenido en este tipo de sistemas. Estos dispositivos aportan flexibilidad y dinamismo al proceso de aprendizaje online, un complemento de gran utilidad en la enseñanza y comunicación entre profesores y alumnos.

1.2 Problemática a resolver

Como se ha mencionado en el apartado anterior, hoy en día existen multitud de plataformas que gestionan diferentes cursos en línea, lo que para el usuario puede llegar a ser una tarea complicada: para poder acceder a los recursos disponibles en cada plataforma, debe registrarse creándose un perfil nuevo con un nombre de usuario y contraseñas diferentes. Esto puede resultar costoso ya que no es fácil recordar múltiples perfiles para cada plataforma virtual.

Para resolver este problema se ha llegado a la conclusión de que es necesario desarrollar una aplicación que permita, a través de su identidad en una red social, que el usuario pueda acceder a cualquier plataforma de manera transparente y sin necesidad de tener que recordar los diferentes usuarios y contraseñas utilizados.

Por lo que se pretende realizar una aplicación que permita la conexión a diferentes plataformas de aprendizaje online y ofrezca al profesor, la posibilidad de compartir ejercicios de clase, de manera que los alumnos puedan acceder a estos contenidos y sirva de complemento a los temas impartidos en la clase presencial.

1.3 Objetivos

El objetivo principal de este proyecto es la realización de una aplicación que utilice la identidad social de sus usuarios, para permitir la conexión a un repositorio de contenidos, en el que el profesor pueda compartir recursos, en este caso, ejercicios de clase, y que el alumno pueda acceder a ellos y resolverlos. En concreto se va a desarrollar una aplicación para el profesor y otra para el alumno.

Los dos objetivos principales de este proyecto son:

- Desarrollar una plataforma que permita al profesor compartir ejercicios de clase y visualizar el contenido de los enunciados y de las soluciones enviadas por los alumnos.
- Desarrollar una plataforma para el alumno que permita acceder a un repositorio de contenidos donde se almacenen los ejercicios de clase enviados por el profesor, pueda enviar soluciones y visualizar el contenido de los enunciados y las soluciones.

Los objetivos específicos para cada una de estas plataformas son los siguientes:

- El profesor:
 - Visualizar un listado de todos los ejercicios que ha compartido.
 - Subir el enunciado de un ejercicio.
 - Visualizar el contenido de este fichero.
 - Ver las soluciones que envían los alumnos.
- El alumno:
 - Visualizar un listado de los ejercicios disponibles que haya añadido el profesor.
 - Ver el enunciado de un ejercicio en concreto.
 - Subir una solución.
 - Visualizar el contenido de dicha solución.

1.4 Motivación

Actualmente, el uso de dispositivos móviles es muy común en la vida de muchas personas. Permiten realizar un gran número de tareas facilitando el día a día de los usuarios. Y no sólo hablamos de teléfonos móviles, sino también de tabletas. Cada vez es más frecuente el uso de estas tabletas como herramienta de ocio o de trabajo.

Hoy en día el uso de estas tecnologías ofrece una amplia variedad de posibilidades. Una de ellas es la de incorporar estos dispositivos como método de trabajo para los

profesores, de forma que sea sencillo compartir información, y que los alumnos puedan acceder a los recursos de una manera práctica y rápida en cualquier momento y en cualquier lugar, como complemento a las clases presenciales.

Existen muchas plataformas que gestionan los recursos que un profesor pone a disposición del estudiante como puede ser Moodle, BlackBoard, etc, por lo que la principal motivación es la de realizar un desarrollo que permita unificar todas esas plataformas de forma que el alumno, desde una misma aplicación, pueda acceder a los recursos independientemente de donde se guarden. Esto se va a llevar a cabo haciendo uso de la identidad social de Twitter del usuario. Se ha elegido esta red social debido a que es una de las más extendidas entre los usuarios de smartphones o tablets y porque es una herramienta muy sencilla, que aporta confianza a los usuarios. Además, permite diferenciar de forma unívoca a un usuario de otro.

Por otro lado, otra motivación más es el hecho de que se ha desarrollado en Android. Actualmente, está en auge y un gran porcentaje de los dispositivos móviles que hay en el mercado, utilizan este sistema operativo. Es una buena oportunidad para mejorar y profundizar en el desarrollo de este tipo de aplicaciones abriendo una puerta a un mercado amplio y en crecimiento.

1.5 Método de trabajo

Durante el desarrollo del proyecto se van a aplicar las denominadas metodologías ágiles. Éstas se basan principalmente en realizar entregas frecuentes y continuas de software que funcione correctamente. Esto se consigue mediante la comunicación entre el equipo de desarrollo y el cliente, aceptando y modificando los cambios que surjan cada vez, y manteniendo una atención continua a la calidad y al diseño.

Por lo que la metodología que se ha seguido a lo largo del proyecto ha sido la de dividir el proyecto en sucesivas iteraciones. Esto ha permitido que el desarrollo del mismo sea iterativo e incremental. En cada una de estas iteraciones se programaba una reunión en la que el tutor acordaba con el grupo de trabajo los objetivos a cumplir y las diferentes tareas para llevar a cabo dichos objetivos. La frecuencia de

estas reuniones ha sido típicamente cada dos o tres semanas. A lo largo de esas semanas el equipo trabajaba en las tareas marcadas con el fin de evaluar en la siguiente reunión la calidad y funcionamiento del software.

Las etapas en las que se ha dividido el proyecto se pueden resumir en las siguientes: En las primeras iteraciones las tareas se centraban en recapitular información sobre los requisitos básicos de la aplicación, en la realización de los prototipos y en los primeros diseños de la misma. En la siguiente etapa, el desarrollo se basó en la creación de las actividades e interfaces y la navegación a través de ellas, modificando y adaptando los diseños a los nuevos requisitos. En la tercera etapa, las iteraciones consistían en implementar las funcionalidades de la aplicación del profesor y del alumno, realizando en todo momento los cambios necesarios tanto en el código como en el diseño para cumplir con los objetivos del cliente. En la fase siguiente, el desarrollo se centró en la creación del LMS a través de un servidor FTP, accediendo mediante la identidad social de Twitter. Por último, la etapa de las pruebas, donde se verifica el correcto funcionamiento de la aplicación en su conjunto y que realmente se cumplen todos los requisitos que se han ido definiendo en cada una de las iteraciones del desarrollo.

Mencionar por otro lado, que durante el proyecto se van a utilizar diferentes herramientas que servirán como apoyo al desarrollo del mismo. Se ha hecho uso de la aplicación Redmine (<http://www.redmine.org/>) para llevar un control de las tareas y del esfuerzo que se ha empleado para cada una de ellas. Para gestionar el código fuente y tener un control de las diferentes versiones por las que pasa la aplicación se utilizará la herramienta Subversion. También servirá para tener una copia de seguridad en un servidor en caso de ocurra algo al directorio local. Por último, se trabajará con el entorno de programación Eclipse actualizado con el plugin correspondiente para desarrollar aplicaciones sobre el sistema operativo Android.

1.6 Organización de la documentación

En este apartado se va a detallar la estructura del documento y la información que se va a describir en cada uno de los capítulos:

- Capítulo 1: En el presente capítulo se describe el contexto del proyecto, la problemática a resolver, así como los objetivos que se pretenden cumplir.
- Capítulo 2: Se detalla el estado del arte, es decir, cuál es la situación actual en el mercado sobre este tipo de aplicaciones realizando un análisis de estas. Por otro lado se hablará de la plataforma de desarrollo elegida y sobre las metodologías ágiles que han sido la metodología utilizada en el proyecto.
- Capítulo 3: En este se hace una especificación detallada de requisitos de la aplicación. Se muestra una descripción de cada uno de ellos y se incluyen los casos de uso.
- Capítulo 4: Se detalla el diseño de la aplicación, desde el primer prototipo, hasta el diseño definitivo, pasando por las diferentes alternativas que se han manejado durante el desarrollo. Se describen también los diagramas de clases y de secuencia.
- Capítulo 5: En este capítulo se enumeran las diferentes pruebas y los resultados que se han obtenido para verificar el correcto funcionamiento de la aplicación.
- Capítulo 6: Se especifica las diferentes iteraciones que ha habido durante la planificación junto a las tareas que se han llevado a cabo para cumplir los objetivos marcados. Se incluye además una descripción detallada de los costes implicados en el desarrollo de éste proyecto.
- Capítulo 7: Se detallan de manera ordenada los resultados obtenidos a lo largo del desarrollo, las conclusiones finales y las futuras líneas tanto de mejora como de ampliación que se puedan aplicar a la aplicación.

1.7 Glosario de términos

- **App:** Se utiliza como abreviatura para denominar a las aplicaciones que se ejecutan en smartphones o tablets.
- **App Store:** Tienda virtual de Apple desde la que se puede acceder a través de Internet y descargar todo tipo de aplicaciones que corren bajo el sistema operativo iOS.

- **Framework:** Estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.
- **Google Play:** Tienda virtual de Google desde la que se puede acceder a través de Internet y descargar todo tipo de aplicaciones que corren bajo el sistema operativo Android.
- **iOs:** Sistema operativo desarrollado y distribuido por Apple.
- **JUnit:** Framework para crear pruebas unitarias en Java.
- **Moodle:** Sistema virtual de gestión de cursos para el aprendizaje en línea.
- **Scrum:** Metodología ágil para el desarrollo de software basado en iteraciones incrementales.
- **Smartphone:** Teléfono móvil inteligente. Tiene mayor capacidad de almacenamiento y realiza tareas similares a las que haría un pequeño ordenador.
- **Tablet:** Es un ordenador personal portátil en forma de tableta con la que el usuario puede interactuar a través de una pantalla multitáctil.
- **Wi-Fi:** Permite la conexión entre dispositivos electrónicos de forma inalámbrica para intercambiar datos o conectarse a Internet.

1.8 Abreviaturas

- **API:** Application programming interface.
- **APK:** Application package file.
- **AVD:** Android Virtual Device.
- **FTP:** File Transfer Protocol.
- **JSON:** JavaScript Object Notation.
- **LMS:** Learning Management System.
- **MPLE:** Mobile Personal Learning Environment.
- **SDK:** Software Development Kit.
- **XML:** Extensible Markup Language.

Introduction

In this first section it's going to describe some aspects to introduce this project, such as knowledge area, the problem to solve and objectives, the motivation and the work's methodology that has applied during the project development. Lastly, it's detailed a list of abbreviations and terms and structure that it's going to be followed throughout this document.

Knowledge area

This project consists of developing an application based in Mobile Personal Learning Environment systems. These online environments are a complement to the normal classes in any student learning.

To understand that idea, it's important to define, firstly, what LMS systems are. These virtual learning environments, like Moodle, consist in a web site where teachers can manage courses easily and students can access to these resources at any moment and through any computer.

Although this technology has had a lot of importance and, nowadays, is present in many educational systems, in some contexts, the possibilities don't cover certain mandatory conditions. This is why they have evolved towards a technology where mobility is transparent to the user, and the resources are structured according to the principles of Web 2.0.

Mobile Personal Learning Environment systems represent a change in the educational technologies, allowing students choose the way of learn, and access to available resources from different sources. This has allowed an evolution of these environments, especially because of the influence that smartphones and tablets have been in this type of systems. These devices provide flexibility and dynamism to the online learning process, a useful complement in teaching and communication between teachers and students.

Problem to solve

As mentioned in the previous section, nowadays, there are many different platforms that manage online courses, so for the user can be a complicated task: to access to the available resources on each platform, he must register creating a new profile with a user name and different passwords. This can be costly because it isn't easy to remember multiple profiles for each virtual platform.

To solve this problem it has been concluded that it's necessary to develop an application that can, through a social network identity, the user may access to any platform transparently without having to remember different usernames and passwords.

It's intended to develop an application which allows a connection to several online learning platforms and offers to the teachers, the possibility of sharing classroom exercises, so that students can access to this content and complement the topics given in a normal class.

Objectives

The main objective of this project is development an application that uses the user's social identity to connect to a content repository in which the teacher can share resources, in this case, class exercises, and a student can access them and solve them. Specifically, it's going to develop an application for the teacher and other for the student.

The two main objectives of this project are:

- Develop a platform that allows the teacher to share class exercises and display the content of the statements and solutions sent by students.
- Develop a platform for the student that allows to access to a content repository where store class exercises sent by the teacher, send solutions and display the content of the statements and solutions.

The specific objectives for each of these platforms include:

- The teacher:
 - View a list of all the exercises teacher has shared.
 - Upload the statement of an exercise.
 - View the content of this file.
 - See the solutions sent by students.
- The student:
 - Display a list of available exercises teacher has added.
 - See the statement of a particular exercise.
 - Send a solution.
 - View the content of this solution.

Motivation

Nowadays, the use of mobile devices is very common in daily life of many people. They allow doing a large number of tasks facilitating the user's day-to-day. The use of tablets is more frequent as a tool for leisure or work.

Technology offers a great variety of possibilities. One among them is to incorporate these devices as a work method for teachers, so it would make it easier to share information. Equally students could have access to the resources in a fast and practical way at anytime and anywhere, as a complement to face to face lessons.

There are many platforms that manage the resources that a teacher disposes to the students, such as Moodle, BlackBoard, etc. For that reason, the principal motivation is to impulse a development that unifies all these platforms in some way. Like that, students, from the same application, could have access to the resources no matter where they these are saved. All of these will be done by using the user's social identity from Twitter. This social network has been chosen because its popularity among smartphones and tablets users and also because it an easy and reliable tool. Moreover, it allows setting the difference from one user to another.

On the other hand, another motivation is the fact that the project has been developed in Android. Currently, it is on its peak, and a huge percentage of the mobile devices that are in the market, use this operating system. It is a great chance to improve and look deeply into this kind of applications development, not closing doors to a huge and increasing market.

Work methodology

In the developing period of this project, agile methodologies have been used. Principally, these are based on frequent and continuous delivers of the software that works properly. This is achieved through communication between the developing team and the client, accepting and modifying the changes that could eventually emerge, caring constantly about quality and design.

So the methodology that has been followed during the project has been to divide it in followings stages. That has made the project's development in stages and incremental. In any of these stages, a meeting was programmed where the tutor established with the working group the objectives and the different tasks to accomplish. The frequency of the meetings has usually been every two or three weeks. Throughout these weeks the team worked in the given tasks in order to evaluate in the next meeting, the quality and running of the software.

The stages the project has been divided in, can be summarized in the followings:

The first stages, the task were focused on getting information about the basic requirements of the application, prototype realizations and first design. The next stage was about creating the activities, the interfaces and browsing through them, modifying and adapting the designs to the new requirements. In the third stage, the steps were about implementing the teachers and students application's functionalities, making all the necessary changes in the code and in the design to accomplish the client's objective. In the next stage, the development was focused in the LMS creation through a FTP server, having access with the Twitter social

identity. Finally, the proof stage, where the correct running of the application was verified, testing that all the requirements have been well-done.

Needs to be mentioned, that during the project, different tools are going to be used that will support the development itself. The Redmine application (<http://www.redmine.org/>) has been used to control the tasks and the effort in every one of them. To manage the source code and control the different versions of the application, the Subversion tool has been used. It will also serve to have a back-up copy in a server in case something happened to the local path. Finally, I will work with the programming environment Eclipse with the corresponding plug-in to develop application with the operating system Android.

Organization of the documentation

This section it's going to detail the structure of the document and the content of each chapter:

- Chapter 1: This chapter describes the context of the project, the problems to be solved and the objectives want to accomplish.
- Chapter 2: It details the state of the art, i.e., what is the current situation in the market about this type of application with an analysis of these. On the other hand, it's going to discuss the chosen development platform and about agile methodologies that have been the methodology used in the project.
- Chapter 3: Describes the specification of requirement's application. It shows a description of each of them and included the use cases.
- Chapter 4: Details the design of the application, from the first prototype to the final design, through the different alternatives that have been handled during development. It also describes the class diagrams and sequence.
- Chapter 5: This chapter lists the different tests and the results that have been obtained to verify the correct operation of the application.
- Chapter 6: Specifies the different iterations that have occurred during the planning with the tasks to be undertaken to meet the objectives. It also includes a detailed description of the costs involved in the development of this project.

- Chapter 7: Reports in an orderly way, the results obtained during the development, the final conclusions and future lines of both expansion and improvement that can be applied to the application.

Glossary of terms

- **App:** Used as an abbreviation to call the applications running on smartphones or tablets.
- **App Store:** Apple online where you can access through Internet and download all sorts of applications that run under the operating system iOS.
- **Framework:** Conceptual and technological structure defined support, usually with artifacts or specific software modules, which can be the basis for the organization and software development.
- **Google Play:** Google Web Store where you can access through Internet and download all sorts of applications running under the Android operating system.
- **iOs:** Operative system developed and distributed by Apple.
- **JUnit:** Framework to create unit tests in Java.
- **Moodle:** Virtual system to manage online learning courses.
- **Scrum:** Agile methodology for software development based on incremental iterations.
- **Smartphone:** Smart phone. It has more storage and performs tasks similar to those that would make a small computer.
- **Tablet:** It is a portable personal computer in tablet shape which the user can interact through a multitouch screen.
- **Wi-Fi:** Allows connection between electronic devices to exchange data wirelessly or connect to the Internet.

Capítulo 2

2. Estado del arte

En esta segunda sección se va a realizar un estudio de mercado comparando las aplicaciones que hay actualmente disponibles para los usuarios con la que se quiere desarrollar en este proyecto. Este estudio se centrará tanto en plataformas móviles como en aplicaciones web. Por otro lado, se hará una descripción del entorno de desarrollo elegido y los motivos por los cuales se ha decidido por Android en vez de iOS. Se detallará en último lugar, en qué se basan las metodologías ágiles que se han aplicado durante el desarrollo del proyecto.

2.1 Análisis del mercado

Para realizar el análisis de mercado de las aplicaciones existentes, se ha realizado una búsqueda en las tiendas virtuales de los sistemas operativos móviles con mayor tasa de mercado en la actualidad como son Google Play de Android, y iTunes para iOS. Para complementar esta información se ha llevado a cabo otra búsqueda a través del buscador web de Google.

Por cada resultado encontrado se detallan cuáles son las utilidades que ofrece y sus características principales, destacando los aspectos más favorables y las funcionalidades de las que carece en comparación con la aplicación que se desea desarrollar. Todas las búsquedas han sido realizadas en el mes de julio del 2013.

Las palabras clave que se han utilizado para encontrar las diferentes aplicaciones han sido las siguientes:

Teacher notebook	Cuaderno	Learn
Student	Estudiante	Profesor
Notebook	Learning	Aprendizaje

Cabe mencionar que a partir de estas búsquedas se han encontrado una gran variedad de aplicaciones disponibles, sin embargo, se describen a continuación, aquellas más relevantes y que poseen funcionalidades similares a las de la que se está desarrollando:

2.1.1 Teacher Notebook

Esta aplicación se centra en la parte de la evaluación de los alumnos. Almacena un listado de los estudiantes de un curso y permite visualizar las calificaciones que han obtenido en un test. Por otro lado, ofrece la posibilidad de llevar un control de la asistencia de las clases y tiene la funcionalidad de poder enviar documentos a través de emails.



Ilustración 1: Teacher notebook icono

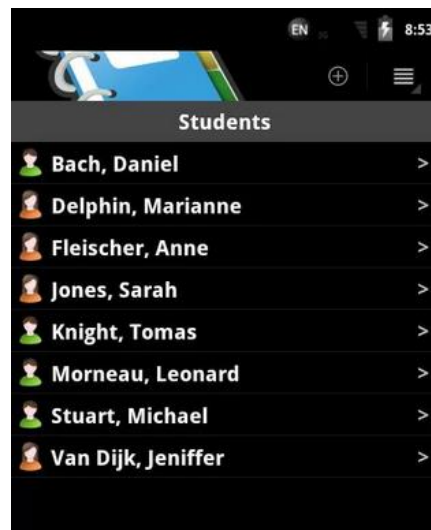


Ilustración 2: Teacher notebook interfaz

2.1.2 Cuaderno del Profesor

Esta segunda aplicación es más completa que la primera. El objetivo principal es el de sustituir el cuaderno físico del profesor para gestionar las clases a través de la aplicación.

Permite calificar a los alumnos y guardar información relativa a las fichas de los estudiantes de cada grupo. Ofrece además, la posibilidad de configurar un horario de clase y la planificación de los temarios e incluso acceder a la agenda de Google para definir el calendario.



Ilustración 3: Cuaderno del profesor icono



Ilustración 4: Cuaderno del profesor menú

1ª Evaluación 21/02/2013	
EVALUACIÓN	FECHA
✓	Busquets Burgos, Sergio
✗	Casillas Fernández, Iker
✓	Fábregas Soler, Cesc
🕒	Iniesta Luján, Andrés
✓	Jiménez Silva, David
✗	Navas González, Jesús
✓	Piqué Bernabéu, Gerard
✓	Ramos García, Sergio

Ilustración 5: Cuaderno del profesor interfaz

2.1.3 AndroClass-Teacher Tablet

A diferencia de las anteriores esta aplicación es bastante más compleja. Por ello no es gratis, y está diseñada para utilizarla a través de una Tablet. Ofrece una interfaz mucho más compleja y visual.

Además de las funcionalidades similares a las que se han descrito en las anteriores aplicaciones esta permite añadir fotografías a la ficha de los alumnos. Ofrece la opción de visualizar las calificaciones de los alumnos por materia y calcular la nota media.

Permite guardar notas y tareas y almacenar los expedientes de los alumnos. Una de las características más destacables en comparación con las anteriores, es que esta cifra los datos almacenados en la base de datos y se requiere de una contraseña para poder acceder e iniciar la aplicación.



Ilustración 6: AndroClass-Teacher Tablet icono

16	16	16	12	2	16	16	16	16	16	16
5°A	5°B	8°A	8°B	8°C	9°A	9°C	10°A	10°D	12°A	
PI Bj -123	2.PI -456	4.PI -890	4.PI -234	PI Bj -345	3.PI -345	PI Bj -456	2.PI -678	2.PI -123	3.PI -345	
9°C										Sortier Index
Daniel, Demo (16*)										Recordatorio
	Aleman									05/25/13 05/30/13
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final		recordatorio clase
	8.51	9.33	9.20	7.32	8.40	9.50	7.83	7.80		05/07/13 06/04/13
	Matemáticas									DVD llevar
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final	Falt	04/23/13 04/30/13 Matemáticas
	10	3	4	11	5	2	6			Daniela, Demo
Tina, Test (15)										Falta
	Aleman									04/18/13 05/02/13 Aleman
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final	Falt	Mika, Model
	5.79	8.80	7.83	4.48	9.00	7.00	5.60	5.01	2	Falta
	Matemáticas									05/25/13 05/26/13 Aleman
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final		Daniel, Demo
	8.60	8.00	9.00	8.50	7.50	8.00	8.80	8.54	1	test - Retraso
Max, Muster (15) [9°A]										05/25/13 05/26/13 Aleman
	Test									Bets, Beispiel
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final	Falt	test - Falta
	9.00	9.00	9.00	9.17	9.00	9.33	7.86	9.09	1	05/25/13 05/31/13 Aleman
	Aleman									Max, Muster
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final		recordatorio alumno
	11	7	4	7	4	3	7			05/25/13 06/01/13 Aleman
Bets, Beispiel (15)										Falta
	Matemáticas									
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final		
	8.16	8.80	6.00	7.60	8.00	10.00	7.33	7.82		
	Test									
	Conce	Oral	Escri	Proce	Tarea	TrCl	Actit	Final		
	10	5	2	5	1	1	3			

Ilustración 7: AndroClass-Teacher Tablet interfaz

2.1.4 Blackboard Mobile Learn

Esta aplicación, a diferencia de las anteriormente descritas, se centra principalmente en los recursos disponibles a través de un curso para acceder a ellos en cualquier momento y para cualquier curso, un objetivo muy similar al que se persigue en este proyecto.

Sus principales características son las de recibir notificaciones sobre cualquier actividad que se lleve a cabo en el curso y visualizar las calificaciones. Además permite adjuntar archivos, leer novedades y realizar publicaciones.



Ilustración 8: Blackboard icono



Ilustración 9: Blackboard interfaz



Ilustración 10: Blackboard ejercicio

2.1.5 iDoceo

Esta aplicación simula un cuaderno de notas para el profesor. La plataforma utilizada es iOS, no es gratuita y está diseñada para utilizarse a través de un iPad.

Permite almacenar un listado de los alumnos y la información asociada a los estudiantes como a las clases. Ofrece la funcionalidad de llevar un diario de las clases y configurar y visualizar un horario de sesiones. Como características más destacables es que se requiere de una contraseña para acceder a ella y permite la restauración de datos perdidos a través de Dropbox.

Biology Sample 101 1st semester		Test 2 Calc	Test 3 Calc	Grade	Project 1	Project 2	Homework 1	Homework 2	Homework 3	Homework 4	Homework 5	Average
Anthony S. Maloney (A1)	1	68	53	C-	C+	A+	84	57	78	57	57	78
Cathy C. Mahon	2	56	65	E	C+	B+	90	69	63	69	69	67
Edward M. Tobey	3	78	70	C	C+	A	80	88	65	88	88	79
Felipe G. Baker	4	55	45	E	A+	B-	80	78	66	78	78	70
Ileana D. Adams	5	63	75	C	C-	B-	84	45	66	45	45	72
Inez E. Moody	6	53	76	E	C	B-	70	55	67	55	55	62
Jerry A. Frizzell	7	45	73	E	C	B	59	53	64	53	53	61
John I. Fair	8	78	74	D	C	C-	70	72	63	72	72	69
Jonathan M. Weiss	9	45	70	C-	C-	B+	58	59	64	59	59	70
Michael S. Peterson	10	19	79	F	B+	B+	70	73	55	73	73	55
Rebecca D. Kemp	11	45	69	E	E	B-	76	74	58	74	74	56
Rikki J. Scott	12	66	78	D	E	C+	70	69	59	69	69	62

Ilustración 11: iDoceo interfaz

2.1.6 Edmodo

Esta aplicación fue encontrada a través del buscador de Google. Se basa en una plataforma social centrada en el alumnado de menor edad, y es capaz de permitir la comunicación entre los alumnos y el profesor en un entorno cerrado. Se encuentra disponible tanto para la plataforma de iOS como en Android, e incluso tiene disponible una versión como página web en el sitio <http://www.edmodo.com/>. Diferencia entre diferentes roles como son Profesor, Alumno, o incluso da la opción para que puedan registrarse los padres de los alumnos.

Permite la posibilidad de crear clases y grupos donde se pueden añadir preguntas o ejercicios que pueden resolver los alumnos. Dentro de la página web se pueden encontrar juegos y además el profesor puede crear eventos en un calendario. Al ser una red social se muestran las novedades y se requiere de un usuario y contraseña para acceder al sitio.



Ilustración 12: Edmodo

2.1.7 Slideshare

Es una plataforma online de aprendizaje que permite compartir recursos en forma de presentaciones de diapositivas. Ofrece además la opción de conectar con las redes sociales para buscar usuarios y compartir información.

Como Edmodo, para acceder a esta aplicación web se requiere de una identificación del usuario, pudiendo ser a través de alguna red social. Permite al usuario suscribirse a cursos para poder acceder a los recursos disponibles como una forma complementaria a la docencia de las clases presenciales. Mencionar por otro lado, que está disponible la versión para dispositivos Android.

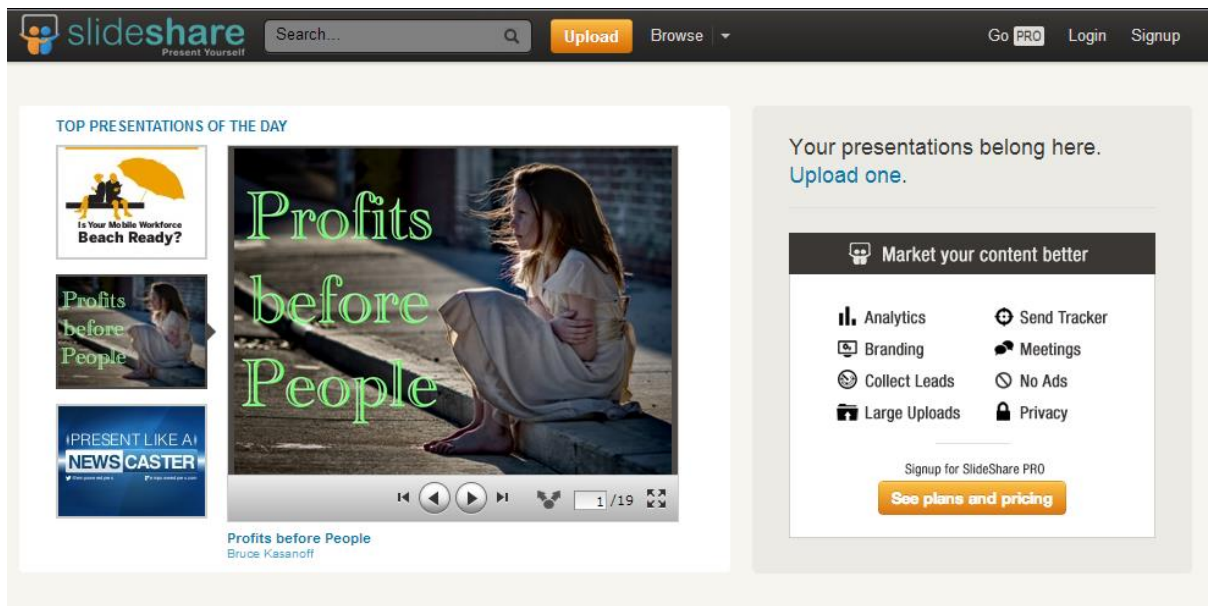


Ilustración 13: Slideshare

2.2 Resultados del análisis de mercado

En la siguiente tabla se hace una comparativa de las aplicaciones descritas indicando si cumple o no los siguientes criterios:

- Identificación del usuario: Si la aplicación permite la autenticación del usuario frente a algún tipo de sistema.
- Conexión a redes sociales: Si permite la conexión a redes sociales.
- Suscribirse a cursos: Si ofrece al alumno la opción de suscribirse a un curso para acceder a los recursos del mismo.
- Disponible para Android: Si la aplicación está disponible para la plataforma Android.
- Complemento a clases: Si el alumno puede acceder a recursos que sirvan como complemento a las clases presenciales.
- Subir ficheros: Si la aplicación permite compartir ficheros.
- Visualizar el contenido: Si ofrece la funcionalidad de mostrar el contenido de los recursos compartidos.

	Identificación del usuario	Conexión a redes sociales	Suscribirse a cursos	Disponible para Android	Complemento a clases	Subir ficheros	Visualización del contenido
Teacher Notebook	No	No	No	Si	No	No	No
Cuaderno del Profesor	No	No	No	Si	N	N	No
AndroClass-Teacher Tablet	Si	No	No	Si	N	N	No
Blackboard Mobile Learn	No	No	Si	Si	Si	Si	No
iDoceo	Si	No	Si	No	No	No	No
Edmodo	Si	No	Si	Si	Si	Si	No
Slideshare	Si	Si	Si	No	Si	Si	Si

Tabla 1: Resultado análisis de mercado

Como se puede apreciar la mayoría de las aplicaciones ofrecen un amplio repertorio de funcionalidades relacionadas con las tareas asociadas a una clase, como son las de evaluar a los alumnos, llevar un control de la asistencia a clase o configurar horarios o sesiones. Sin embargo, sólo dos aplicaciones permiten estar suscritos en cursos y acceder a los recursos disponibles a través de estos, y sólo una de ellas, Slideshare, permite la integración con redes sociales.

Se puede concluir que es necesario implementar una aplicación que permita acceder a diferentes cursos, independiente de la plataforma, a través de la identidad de la red social de Twitter. Además, muy pocas de las aplicaciones que se han analizado ofrecen la posibilidad de gestionar los recursos y visualizar el contenido de los ejercicios así como de poder enviar ficheros a la plataforma, requisitos que debe cumplir la aplicación a implementar.

2.3 Plataforma de desarrollo

Actualmente, los entornos móviles más utilizados en el mercado, según un estudio realizado en el mes de mayo de este año, son Android, BlackBerry, iOS, Symbian, Windows Phone y otros:

Top Five Smartphone Operating Systems, Shipments, and Market Share, 1Q 2013 (Units in Millions)

Operating System	1Q13 Shipment Volume	1Q13 Market Share	1Q12 Shipment Volume	1Q12 Market Share	Year over Year Change
Android	162.1	75.0%	90.3	59.1%	79.5%
iOS	37.4	17.3%	35.1	23.0%	6.6%
Windows Phone	7.0	3.2%	3.0	2.0%	133.3%
BlackBerry OS	6.3	2.9%	9.7	6.4%	-35.1%
Linux	2.1	1.0%	3.6	2.4%	-41.7%
Symbian	1.2	0.6%	10.4	6.8%	-88.5%
Others	0.1	0.0%	0.6	0.4%	-83.3%
Total	216.2	100.0%	152.7	100.0%	41.6%

Ilustración 14: Volumen de mercado sistemas operativos móviles

Como se puede apreciar el volumen de mercado del sistema operativo Android abarca el 75% del total, es decir, tres de cada cuatro móvil que se venden, tienen el sistema operativo de Google. Empresas como Samsung, HTC o Sony han doblado sus ventas alcanzando, en conjunto, aproximadamente 150 millones de dispositivos.

Worldwide Smartphone Sales to End Users by Operating System in 1Q13 (Thousands of Units)

Operating System	1Q13 Units	1Q13 Market Share (%)	1Q12 Units	1Q12 Market Share (%)
Android	156,186.0	74.4	83,684.4	56.9
iOS	38,331.8	18.2	33,120.5	22.5
Research In Motion	6,218.6	3.0	9,939.3	6.8
Microsoft	5,989.2	2.9	2,722.5	1.9
Bada	1,370.8	0.7	3,843.7	2.6
Symbian	1,349.4	0.6	12,466.9	8.5
Others	600.3	0.3	1,242.9	0.8
Total	210,046.1	100.0	147,020.2	100.0

Source: Gartner (May 2013)

Ilustración 15: Unidades vendidas según sistema operativo

Atendiendo a los datos de ambas figuras, se ha decidido desarrollar esta aplicación utilizando la plataforma Android y, aunque el volumen de mercado ha sido uno de los principales motivos a la hora de elegir la plataforma de desarrollo, hay otros que se han tenido en cuenta para inclinar la balanza en el lado de Android en vez de iOS.

El primero de estos motivos es que la tasa de descarga de aplicaciones en Android corresponde a un 51% del total, mientras que iOS alcanza el 40%. Por otro lado, las aplicaciones de iOS están optimizadas para un Smartphone o tablet en concreto, iPhone y iPad, respectivamente, mientras que Android se ejecuta en una gran variedad de dispositivos con características diferentes.

Android es un sistema operativo propiedad de Google, diseñado principalmente para dispositivos móviles y con pantalla táctil, como pueden ser smartphones o tablets. A diferencia de iOS o Symbian, este sistema está basado en Linux. Debido a esta característica, Android permite que cualquier usuario pueda bajarse el código fuente, verlo, e incluso modificarlo y volver a compilarlo. Los componentes que forma la arquitectura en la que se basa Android se resumen en la siguiente ilustración:

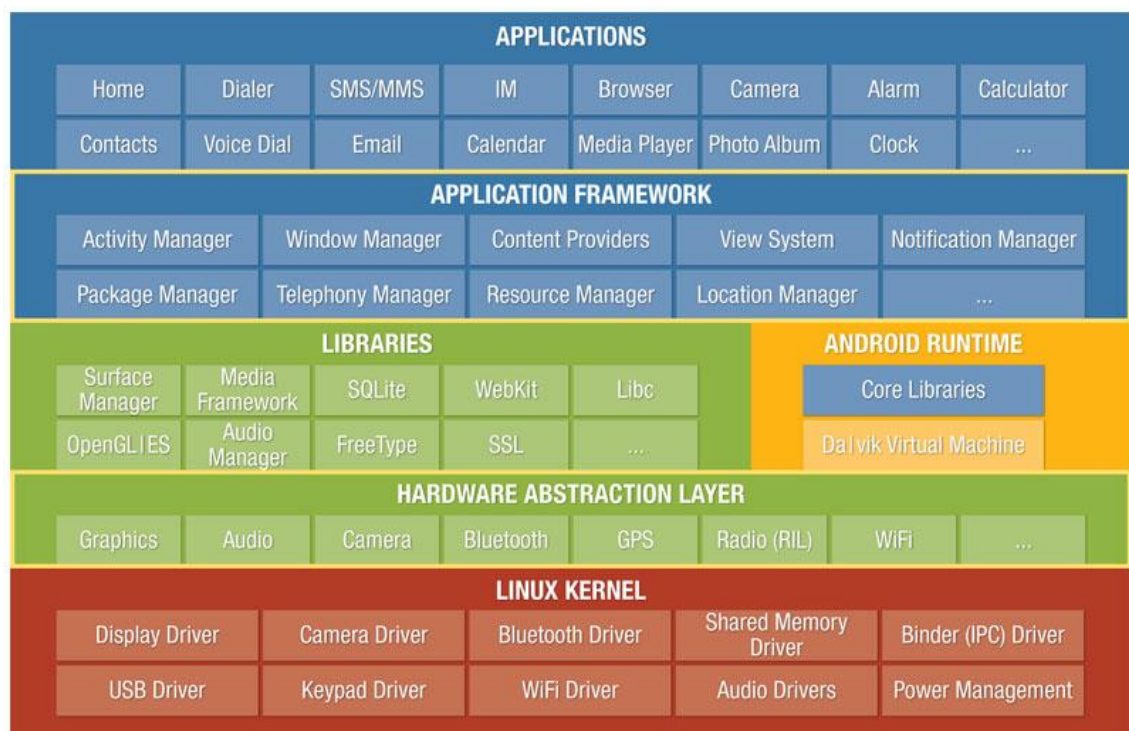


Ilustración 16: Arquitectura Android

Las aplicaciones que se desarrollan bajo este sistema se basan en una variación de Java denominada Dalvik. Esto hace mucho más fácil el desarrollo de aplicaciones ya que además, el sistema operativo proporciona interfaces que permiten al

programador acceder a cualquier función del dispositivo móvil. Sin embargo, hay que especificar que, aunque las aplicaciones están en el lenguaje de Java, no existe una máquina virtual de Java en la plataforma, sino que lo que se hace es compilar el bytecode en un ejecutable Dalvik y ejecutarlo en la máquina virtual Dalvik.

Algunas de las características más destacadas de Android son que ofrece un sistema de almacenamiento de bases de datos en SQLite propio de cada dispositivo, multitarea real de aplicaciones, esto es, que aquellas que no estén en primer plano, siguen recibiendo ciclos de reloj y además, soporta Tethering. Esto permite al dispositivo móvil ser usado como pasarela para permitir el acceso inalámbrico a la red a otros dispositivos.

Por otro lado ofrece un entorno de desarrollo integrado. Durante la implementación se utiliza Eclipse con la instalación del plugin correspondiente al SDK de desarrollo de Android. Ofrece un emulador de dispositivos hardware AVD, herramientas de depuración como el denominado LogCat y el DDMS, un explorador del sistema de ficheros del dispositivo, entre otros. En este caso se ha utilizado la versión de Eclipse 3.7 Indigo.

Las versiones que ha tenido este sistema operativo desde que se vendiera en Octubre del 2008 el primer móvil con Android han sido las siguientes:

Versión	Nombre
1.0	Apple Pie
1.1	Banana Bread
1.5	Cupcake
1.6	Donut
2.0/2.01/2.1	Eclair
2.2-2.2.3	Froyo
2.3-2.3.2 / 2.3.3-2.3.7	Gingerbread
3.0/3.1/3.2	Honeycomb
4.0-4.0.2/4.0.3-4.0.4	Ice Cream Sandwich

4.1/4.2/4.3	Jelly Bean
4.4	KitKat

2.4 Metodología de desarrollo ágil

A lo largo de este proyecto se va a aplicar SCRUM como metodología de desarrollo ágil. Es un método de trabajo para la gestión y desarrollo de software basada en un proceso incremental e iterativo por lo que el proyecto está dividido en una serie de entregas parciales llamadas Sprints. Este tipo de metodología se aplica en entornos en los que se necesita obtener resultados tempranos, donde los requisitos cambian a menudo y donde los aspectos más importantes son la flexibilidad y la productividad.

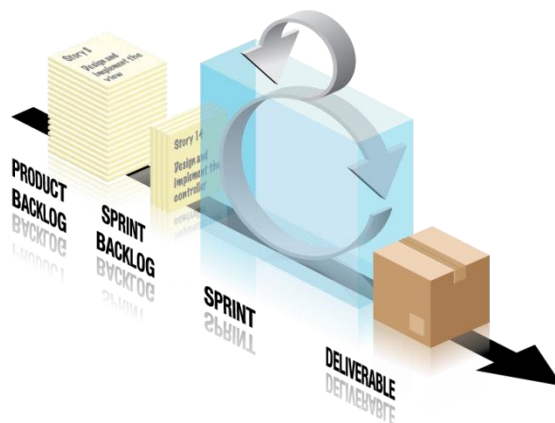


Ilustración 17: Ciclo de vida SCRUM

Como se ha comentado anteriormente un proyecto se divide en bloques de tiempos cortos y fijos, en el caso que nos ocupa, son iteraciones de un mes. En cada entrega se debe proporcionar la parte incremental correspondiente y acordada tanto de la parte del código de la implementación como de la documentación.

Cada proceso de iteración está constituido por 3 etapas: En la primera de ellas se lleva a cabo la planificación de ese bloque temporal. En este caso el tutor presenta los objetivos más prioritarios que deben alcanzarse, y posteriormente, se elabora la

lista de tareas que son necesarias para conseguir esos objetivos estimando el tiempo que se va a emplear en cada una de ellas.

En la segunda fase se realiza el desarrollo de cada una de las tareas propuestas. A lo largo de esta etapa se organizan reuniones o tutorías para mostrar los resultados obtenidos o aquellos obstáculos que impiden que los objetivos no puedan cumplirse o que no se puedan resolver.

En la última fase, se presenta al cliente los requisitos y objetivos alcanzados y se muestran cuáles han sido los cambios respecto a la parte incremental del proyecto. Por otro lado, se intentan resolver aquellas dificultades que se han ido encontrando a lo largo del mes y se acuerda una solución viable que cumpla los requisitos del cliente, y que a su vez, sea factible en el contexto de la implementación del proyecto.

Normalmente, en este tipo de metodologías ágiles, se hace uso de herramientas que facilitan al equipo de desarrollo la gestión de tareas. Para este punto se suele hacer uso de un tablero o pizarra dividido en columnas donde se muestra el estado de las tareas situadas en la columna que corresponda, ya sea “Tareas nuevas”, “En progreso” o “Realizadas”. Cada miembro del equipo se asigna una nueva tarea y según vaya progresando en ella, va colocándola en la columna correcta. En este proyecto, y como se ha dicho en apartados anteriores, esta función se lleva a cabo mediante la herramienta Redmine. Esto permite ver de una manera rápida qué tareas están sin empezar y cuales en desarrollo, y además, ver de forma global el estado final del proyecto.

2.5 Resumen del estado del arte

A lo largo de este apartado se ha realizado un estudio de las aplicaciones similares a la que se pretende desarrollar, y tras ver las funcionalidades y características de cada una de ellas en la sección 2.2, se ha llegado a la conclusión de que ninguna abarca los objetivos que se pretenden cumplir en este proyecto. Por lo que es necesario implementar una aplicación que permita acceder a diferentes cursos,

independientemente de la plataforma, a través de la identidad de la red social de Twitter, y en la que el profesor y el alumno puedan compartir los enunciados y las soluciones de ejercicios de clase.

Respecto a la plataforma de desarrollo se ha optado por Android debido a que actualmente el 75% del volumen de mercado corresponde a móviles que operan bajo el sistema operativo de Android. Además el porcentaje de descargas de aplicaciones es mucho mayor que el existente en el sistema operativo iOS, por lo que se deduce que el número de usuarios que pueden tener acceso a esta aplicación es mucho mayor si se realiza en Android que si se hiciera en otra plataforma de desarrollo de aplicaciones móviles.

En lo que a metodologías ágiles se refiere, se ha optado por Scrum debido a que en este proyecto todos los requisitos no están definidos al inicio y además pueden variar a lo largo del proyecto. Debido a las características que posee Scrum, permite llevar a cabo una metodología progresiva y adaptativa, de tal forma que en cada una de las iteraciones se definen nuevos requisitos, se adapta y modifica el código a las nuevas condiciones y se entrega software completamente funcional y estable, para evaluarlo en la siguiente iteración.

Capítulo 3

3. Especificación de requisitos

En este apartado se va a realizar una especificación de requisitos. Se van detallar los requisitos que debe cumplir el proyecto para garantizar el correcto funcionamiento de la aplicación. Por otro lado, se describen los casos de uso y, por último, se enumeran las restricciones que se deben cumplir.

3.1 Requisitos específicos

Para que quede más claro y sencillo se va a hacer una división entre la aplicación del profesor y la aplicación del alumno. Cada requisito se va a definir siguiendo la siguiente estructura:

Identificador		Versión	
Nombre			
Descripción			
Importancia	Esencial / Opcional	Prioridad	Alta / Media / Baja

Tabla 2: Tabla modelo requisitos

- Cuaderno del profesor:

Identificador	CP001	Versión	1
Nombre	Identificación del usuario		
Descripción	El usuario podrá iniciar sesión en la aplicación mediante su identidad social de Twitter		
Importancia	Esencial	Prioridad	Alta

Tabla 3: CP001

Identificador	CP002	Versión	1
Nombre	Listar identidades de Twitter		
Descripción	La aplicación mostrará una lista de las identidades de Twitter que el usuario tenga registradas en su Tablet		
Importancia	Esencial	Prioridad	Baja

Tabla 4: CP002

Identificador	CP003	Versión	1
Nombre	Conexión a LMS		

Descripción	La aplicación será capaz de conectarse a diferentes plataformas de aprendizaje (LMS)		
Importancia	Esencial	Prioridad	Alta

Tabla 5: CP003

Identificador	CP004	Versión	1
Nombre	Listado de los alumnos		
Descripción	La aplicación mostrará en forma de lista el nombre y apellidos de los alumnos que siguen un curso		
Importancia	Esencial	Prioridad	Alta

Tabla 6: CP004

Identificador	CP005	Versión	1
Nombre	Listado de los ejercicios		
Descripción	La aplicación mostrará en forma de lista todos los ejercicios que el profesor ha subido al servidor		
Importancia	Esencial	Prioridad	Alta

Tabla 7: CP005

Identificador	CP006	Versión	1
Nombre	Listado de los ejercicios resueltos por un alumno		
Descripción	La aplicación será capaz de listar solamente aquellos ejercicios que un determinado alumno ha enviado una solución.		
Importancia	Esencial	Prioridad	Alta

Tabla 8: CP006

Identificador	CP007	Versión	1
Nombre	Listado de los alumnos que han resuelto un ejercicio		
Descripción	La aplicación podrá listar solamente aquellos alumnos que hayan enviado una solución a un ejercicio seleccionado		
Importancia	Esencial	Prioridad	Alta

Tabla 9: CP007

Identificador	CP008	Versión	1
Nombre	Subida de un fichero		
Descripción	La aplicación será capaz de subir un archivo que el profesor haya seleccionado del dispositivo como nuevo ejercicio		
Importancia	Esencial	Prioridad	Alta

Tabla 10: CP008

Identificador	CP009	Versión	1
Nombre	Mostrar enunciado de un ejercicio		
Descripción	El profesor podrá visualizar el contenido de un enunciado que haya subido con anterioridad		
Importancia	Esencial	Prioridad	Alta

Tabla 11: CP009

Identificador	CP010	Versión	1
Nombre	Mostrar solución de un ejercicio		
Descripción	El profesor podrá visualizar el contenido de una solución de un ejercicio enviada por un alumno		
Importancia	Esencial	Prioridad	Alta

Tabla 12: CP010

Identificador	CP011	Versión	1
Nombre	Guardar fichero en la plataforma		
Descripción	La aplicación será capaz de almacenar el fichero seleccionado en la ruta correspondiente del servidor		
Importancia	Esencial	Prioridad	Alta

Tabla 13: CP011

Identificador	CP013	Versión	1
Nombre	Configurar cuentas LMS		
Descripción	El usuario podrá configurar los datos asociados a las cuentas utilizadas para acceder a las plataformas de aprendizaje		
Importancia	Esencial	Prioridad	Media

Tabla 14: CP013

- Cuaderno del alumno:

Identificador	CA001	Versión	1
Nombre	Identificación del usuario		
Descripción	El usuario podrá iniciar sesión en la aplicación mediante su identidad social de Twitter		
Importancia	Esencial	Prioridad	Alta

Tabla 15: CA001

Identificador	CA002	Versión	1
Nombre	Listar identidades de Twitter		
Descripción	La aplicación mostrará una lista de las identidades de Twitter que el usuario tenga registradas en su Tablet		
Importancia	Esencial	Prioridad	Baja

Tabla 16: CA002

Identificador	CA003	Versión	1
Nombre	Conexión a LMS		
Descripción	La aplicación será capaz de conectarse a diferentes plataformas de aprendizaje (LMS)		
Importancia	Esencial	Prioridad	Alta

Tabla 17: CA003

Identificador	CA004	Versión	1
Nombre	Listado de ejercicios		
Descripción	La aplicación será capaz de listar todos los ejercicios que el profesor de un curso ha subido		
Importancia	Esencial	Prioridad	Alta

Tabla 18: CA004

Identificador	CA005	Versión	1
Nombre	Subir solución de un ejercicio		
Descripción	La aplicación será capaz de subir un archivo que el alumno haya seleccionado del dispositivo como solución a un ejercicio		
Importancia	Esencial	Prioridad	Alta

Tabla 19: CA005

Identificador	CA006	Versión	1
Nombre	Guardar fichero en la plataforma		
Descripción	La aplicación será capaz de almacenar el fichero seleccionado en la ruta correspondiente del servidor		
Importancia	Esencial	Prioridad	Alta

Tabla 20: CA006

Identificador	CA007	Versión	1
Nombre	Mostrar enunciado de un ejercicio		
Descripción	El alumno podrá visualizar el contenido de un enunciado que el profesor haya subido con anterioridad		
Importancia	Esencial	Prioridad	Alta

Tabla 21: CA007

Identificador	CA008	Versión	1
Nombre	Mostrar solución de un ejercicio		
Descripción	El alumno podrá visualizar el contenido de una solución de un ejercicio que haya enviado con anterioridad		
Importancia	Esencial	Prioridad	Alta

Tabla 22: CA008

Identificador	CA009	Versión	1
Nombre	Añadir asignatura		
Descripción	El usuario podrá añadir de forma manual la identidad social correspondiente a una asignatura para acceder a los ejercicios de clase compartidos por el profesor		
Importancia	Esencial	Prioridad	Media

Tabla 23: CA009

3.2 Casos de uso

En este apartado se van a definir los casos de uso aplicación con el objetivo de describir el comportamiento de la aplicación desde el punto de vista del usuario. Es decir, un caso de uso define como un actor interactúa con el sistema para realizar una tarea en concreto, y los pasos que éste realiza para llevarla a cabo.

A continuación se muestra de forma global un diagrama con los casos de uso que se describirán con más detalle más adelante:

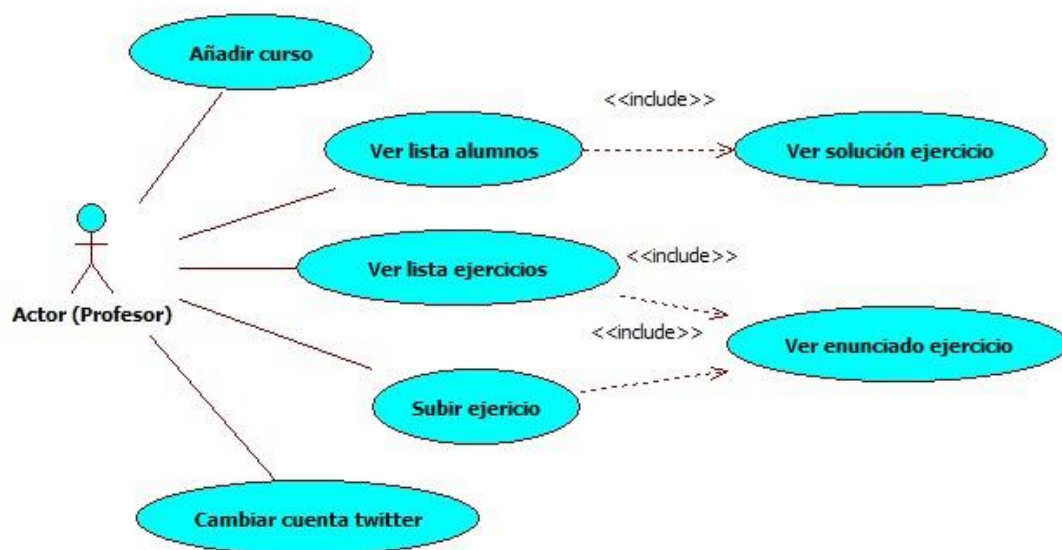


Ilustración 18: Diagrama casos de uso (Profesor)

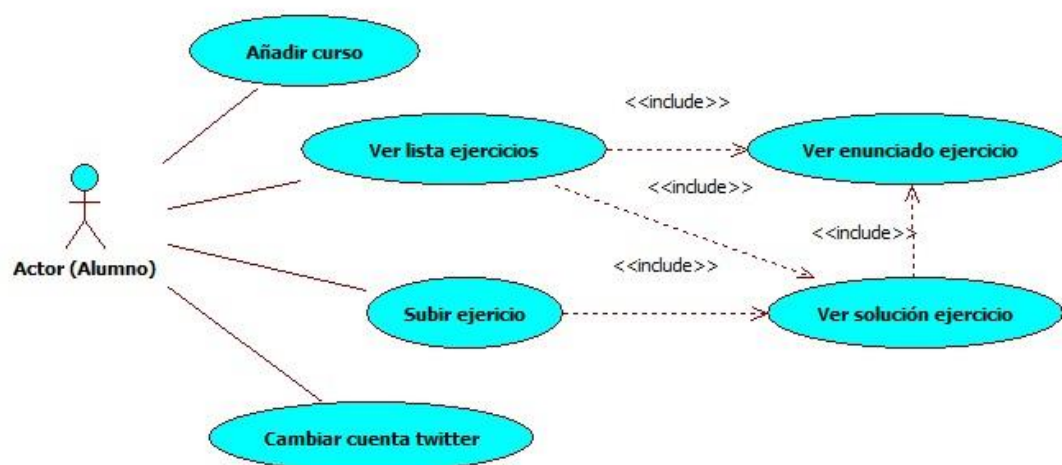


Ilustración 19: Diagrama casos de uso (Alumno)

Cada uno de los casos de uso se va a describir siguiendo la siguiente plantilla:

Nombre	
Actor	
Requisito	
Descripción	
Precondiciones	
Flujo normal	

Flujo alternativo
Postcondiciones

Tabla 24: Plantilla caso de uso

Nombre	Ver lista alumnos
Actor	Profesor
Requisito	CP004
Descripción	
El profesor pulsa la pestaña para visualizar la lista de alumnos de un curso	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Añadir un curso 	
Flujo normal	
<ul style="list-style-type: none"> • La aplicación realiza la petición para mostrar la lista de alumnos al LMS • El LMS accede al directorio del servidor donde se guarda el fichero que contiene la información correspondiente a los alumnos • EL LMS lee el fichero y devuelve el nombre de todos los alumnos en formato JSON • La aplicación recibe el contenido y lo muestra en forma de lista 	
Flujo alternativo	
<ul style="list-style-type: none"> • La aplicación realiza la petición para mostrar la lista de alumnos al LMS • El LMS accede al directorio del servidor pero no encuentra el fichero de los datos de los alumnos • Se controla el error y se devuelve un contenido vacío • La aplicación no muestra ninguna lista 	
Postcondiciones	
<ul style="list-style-type: none"> • Se muestra una lista con los nombres de los alumnos que siguen el curso 	

Tabla 25: Caso de uso Ver lista alumnos

Nombre	Ver lista ejercicios
Actor	Profesor
Requisito	CP005
Descripción	
El profesor pulsa la pestaña para visualizar la lista de los ejercicios disponibles	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Añadir un curso • Subir algún ejercicio al servidor 	
Flujo normal	
<ul style="list-style-type: none"> • La aplicación realiza la petición para mostrar la lista de ejercicios al LMS • El LMS accede al directorio del servidor donde se guarda la estructura de los ejercicios • EL LMS recupera el nombre de las carpetas que almacenan el enunciado de un ejercicio en formato JSON • La aplicación recibe el contenido y lo muestra en forma de lista 	
Flujo alternativo	
<ul style="list-style-type: none"> • La aplicación realiza la petición para mostrar la lista de alumnos al LMS • El LMS accede al directorio del servidor pero no se encuentra ningún ejercicio • Se controla el error y se devuelve un contenido vacío • La aplicación no muestra ninguna lista 	
Postcondiciones	
<ul style="list-style-type: none"> • Se muestra una lista con los nombres de los alumnos que siguen el curso 	

Tabla 26: Caso de uso Ver lista ejercicios

Nombre	Subir un ejercicio
Actor	Profesor
Requisito	CP008
Descripción	
El profesor pulsa el botón para subir un ejercicio y selecciona un fichero de su dispositivo	
Precondiciones	

<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Añadir un curso • Tener el fichero correspondiente al enunciado en la tablet
Flujo normal
<ul style="list-style-type: none"> • La aplicación muestra la interfaz que va a llevar a cabo la subida del ejercicio • El profesor introduce el nombre de la carpeta donde se albergará el enunciado y selecciona la ruta correspondiente al fichero que desee compartir • La aplicación realiza la petición de subida de fichero y envía el fichero seleccionado al LMS • El LMS recibe el fichero elegido y accede al servidor • Crea un nuevo directorio con el nombre que ha introducido el profesor y guarda el fichero • EL LMS envía a la aplicación un mensaje de que el fichero ha sido subido con éxito • La aplicación recibe el mensaje, notifica al usuario de que se ha subido correctamente y vuelve a la interfaz principal
Flujo alternativo
<ul style="list-style-type: none"> • La aplicación realiza la petición al LMS para subir el fichero • El LMS no recibe de forma correcta el fichero • Se controla el error y el LMS envía un mensaje de error a la aplicación • La aplicación notifica al usuario que ha habido un error durante la subida del fichero
Postcondiciones
<ul style="list-style-type: none"> • El enunciado de un nuevo ejercicio está disponible y accesible para los alumnos

Tabla 27: Caso de uso Subir un ejercicio

Nombre	Añadir curso
Actor	Profesor
Requisito	CP013
Descripción	

El usuario pulsa el botón para añadir y configurar un nuevo curso
Precondiciones
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación
Flujo normal
<ul style="list-style-type: none"> • La aplicación muestra la interfaz donde se va a configurar los datos del curso • El usuario introduce la URL del servidor, el path necesario para acceder a los ejercicios y el nombre de usuario y la contraseña asociados a ese curso • La aplicación realiza la petición al LMS para añadir el nuevo curso y envía los datos introducidos • El LMS captura los datos enviados y los almacena en el fichero de configuración creado para dicho fin con formato JSON • EL LMS envía a la aplicación un mensaje de que se ha creado el curso correctamente • La aplicación recibe el mensaje, notifica al usuario de que ha ido correctamente y vuelve a la interfaz principal
Flujo alternativo
<ul style="list-style-type: none"> • El usuario introduce datos con formato incorrecto • La aplicación notifica al usuario que inserte los datos con el formato adecuado
Postcondiciones
<ul style="list-style-type: none"> • Un nuevo curso añadido

Tabla 28: Caso de uso Añadir curso

Nombre	Cambiar cuenta Twitter
Actor	Profesor / Alumno
Requisito	CP001/CA001, CP002/CA002
Descripción	
El usuario pulsa el botón para cambiar entre sus cuentas de Twitter	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Tener cuentas registradas en el dispositivo • Tener configurados cursos 	

Flujo normal
<ul style="list-style-type: none"> • La aplicación accede a la información registrada en el dispositivo del usuario y recupera el id de cada cuenta de Twitter • Se muestra dicha información en forma de lista en una ventana emergente • El usuario selecciona el id que desee • El LMS recupera los datos relativos al curso correspondiente al id que el usuario ha seleccionado • La aplicación se reinicia para actualizar la lista de ejercicios
Flujo alternativo
<ul style="list-style-type: none"> •
Postcondiciones
<ul style="list-style-type: none"> • La aplicación carga la lista de ejercicios y de alumnos del curso asociado al id que el usuario ha seleccionado

Tabla 29: Caso de uso Cambiar cuenta Twitter

Nombre	Subir solución
Actor	Alumno
Requisito	CA005
Descripción	
El alumno pulsa el botón para añadir una solución de un ejercicio	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Añadir un curso • Tener el fichero correspondiente al enunciado en la tablet 	
Flujo normal	
<ul style="list-style-type: none"> • La aplicación muestra la interfaz que va a llevar a cabo la subida de la solución • El alumno selecciona un ejercicio de los disponibles y selecciona la ruta correspondiente al fichero que desea enviar al profesor • La aplicación realiza la petición de subida de fichero y envía el fichero seleccionado al LMS • El LMS recibe el fichero elegido y accede al servidor • Accede al directorio correspondiente al ejercicio seleccionado y guarda la 	

<p>solución</p> <ul style="list-style-type: none"> • EL LMS envía a la aplicación un mensaje de que el fichero ha sido subido con éxito • La aplicación recibe el mensaje, notifica al usuario de que se ha subido correctamente y vuelve a la interfaz principal
Flujo alternativo
<ul style="list-style-type: none"> • La aplicación realiza la petición al LMS para subir el fichero • El LMS no recibe de forma correcta el fichero • Se controla el error y el LMS envía un mensaje de error a la aplicación • La aplicación notifica al usuario que ha habido un error durante la subida del fichero
Postcondiciones
<ul style="list-style-type: none"> • La solución de un ejercicio determinado está disponible y accesible para que el profesor pueda consultarla

Tabla 30: Caso de uso Subir solución

Nombre	Ver enunciado ejercicio
Actor	Profesor / Alumno
Requisito	CP009/CA007
Descripción	
El usuario selecciona un ejercicio para visualizar el enunciado del mismo	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Ejercicios disponibles 	
Flujo normal	
<ul style="list-style-type: none"> • El usuario selecciona de la lista un ejercicio • La aplicación realiza la petición al LMS para visualizar el enunciado del ejercicio seleccionado • El LMS accede al servidor y busca la carpeta con el nombre del ejercicio seleccionado • Una vez en el directorio, localiza el fichero que corresponde al enunciado y envía a la aplicación la ruta de este • La aplicación recibe la url del fichero y muestra el contenido al usuario a 	

través de la herramienta de google
Flujo alternativo
<ul style="list-style-type: none"> • La aplicación realiza la petición al LMS para visualizar el enunciado del ejercicio seleccionado • El LMS accede al servidor y busca la carpeta con el nombre del ejercicio seleccionado • No se encuentra el fichero correcto • La aplicación muestra un mensaje de que no se puede mostrar el contenido del fichero
Postcondiciones
<ul style="list-style-type: none"> • El usuario visualiza en su dispositivo el enunciado que el profesor ha añadido

Tabla 31: Caso de uso Ver enunciado ejercicio

Nombre	Ver solución ejercicio
Actor	Profesor / Alumno
Requisito	CP010/CA008
Descripción	
El usuario selecciona un ejercicio para visualizar la solución del mismo	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación • Ejercicios disponibles • Estar disponible la solución a ese ejercicio 	
Flujo normal	
<ul style="list-style-type: none"> • El usuario selecciona de la lista un ejercicio • La aplicación realiza la petición al LMS para visualizar la solución del ejercicio seleccionado • El LMS accede al servidor y busca la carpeta con el nombre del ejercicio seleccionado • Una vez en el directorio, localiza el fichero que corresponde a la solución cuyo nombre coincide con el id de twitter del alumno y envía la ruta a la aplicación • La aplicación recibe la url del fichero y muestra el contenido al usuario a 	

través de la herramienta de google
Flujo alternativo
<ul style="list-style-type: none"> • La aplicación realiza la petición al LMS para visualizar la solución del ejercicio seleccionado • El LMS accede al servidor y busca la carpeta con el nombre del ejercicio seleccionado • No se encuentra el fichero de la solución que se requiere • La aplicación muestra un mensaje de que no se puede mostrar el contenido del fichero
Postcondiciones
<ul style="list-style-type: none"> • El usuario visualiza en su dispositivo la solución que el alumno ha enviado al profesor

Tabla 32: Caso de uso Ver solución ejercicio

Nombre	Seleccionar una asignatura
Actor	Alumno
Requisito	CA003/CA009
Descripción	
El alumno pulsa la pestaña para mostrar las asignaturas añadidas anteriormente	
Precondiciones	
<ul style="list-style-type: none"> • Iniciar sesión en la aplicación 	
Flujo normal	
<ul style="list-style-type: none"> • El alumno pulsa en la opción para cambiar de asignatura • El alumno pulsa en el botón de añadir asignatura • LA aplicación carga la interfaz correspondiente • El alumno introduce el nombre de la asignatura • El alumno pulsa en el botón de Guardar • La aplicación lo almacena y muestra en un popup todas las asignaturas • El alumno selecciona la asignatura que desee • La aplicación se recarga 	
Flujo alternativo	
<ul style="list-style-type: none"> • El alumno selecciona una asignatura 	

<ul style="list-style-type: none"> • La asignatura no existe o hay fallo al cargar los datos de configuración • Se controla el error y se notifica al usuario
Postcondiciones
<ul style="list-style-type: none"> • Se muestra una lista con los ejercicios disponibles de esa asignatura

Tabla 33: Caso de uso Cambiar asignatura

3.3 Restricciones

Además de los requisitos definidos anteriormente, es importante destacar que esta aplicación necesita interactuar con dos sistemas ajenos a la implementación de este proyecto por lo que garantizar el correcto funcionamiento de estos queda fuera de los requisitos y obligaciones del desarrollo de esta aplicación. Los sistemas que entran en juego y las funcionalidades que van a llevar a cabo son las siguientes:

- Red social Twitter: Como se ha dicho anteriormente se va a hacer uso de la identidad de Twitter del usuario para autenticarse y acceder a los recursos de la aplicación.
- Servidor LMS: Será el lugar donde se almacene la estructura del sistema de carpetas en la que se guardarán los ejercicios del profesor y las soluciones de los alumnos.

Como cualquier proyecto va a tener diferentes restricciones para asegurarse el buen funcionamiento de la aplicación. En este caso la comunicación con el LMS se realiza a través de FTP, por lo que el sistema se ha implementado en base a una API determinada. En el momento que se desee añadir otra plataforma como LMS se deberá desarrollar siguiendo el API correspondiente.

Mencionar por otro lado, que la velocidad con que la aplicación recupere y muestre los ejercicios compartidos depende completamente del terminal sobre el que se ejecute y de las condiciones de la red en ese momento, además de las dependencias que surjan por el uso de Twitter o del servidor LMS en cuestión.

La aplicación requiere en todo momento la conexión a Internet para acceder y compartir ejercicios. Como aspecto positivo, no almacenará ningún dato de acceso del usuario, por lo que no se requiere de ningún tipo de sistema de cifrado de datos y la seguridad recae en las medidas de seguridad utilizadas por la red social Twitter.

Capítulo 4

4. Diseño

A lo largo de este apartado se va a mostrar los diferentes prototipos que se han diseñado durante el desarrollo del proyecto, desde el prototipo inicial hasta el diseño que finalmente se ha implementado.

Por otro lado, se describen las diferentes alternativas que se han barajado hasta el final del proyecto detallando los puntos más favorables y los menos, y se adjuntan diagramas de secuencia y un diagrama de clases para entender con más facilidad el funcionamiento de la aplicación.

4.1 Prototipo

En este apartado se muestra el primer prototipo que se diseñó tras realizar el primer análisis de las funcionalidades principales. Este diseño se realizó al empezar el desarrollo durante el mes de enero del 2013. Como se ha dicho en apartados anteriores, se ha dividido el proyecto en dos aplicaciones. Se describirá, por cada versión del prototipo, primero el diseño de la aplicación del profesor y posteriormente, la del alumno.

En la siguiente figura se muestra la pantalla principal de la aplicación del profesor. Esta primera interfaz el profesor puede visualizar en una lista los id de los alumnos que siguen un curso. En la parte de la derecha aparece mediante una notificación si el alumno ha enviado una nueva solución. En la barra superior se muestran dos botones; uno para enviar un nuevo enunciado y el otro para introducir los datos del profesor de Twitter para iniciar sesión.

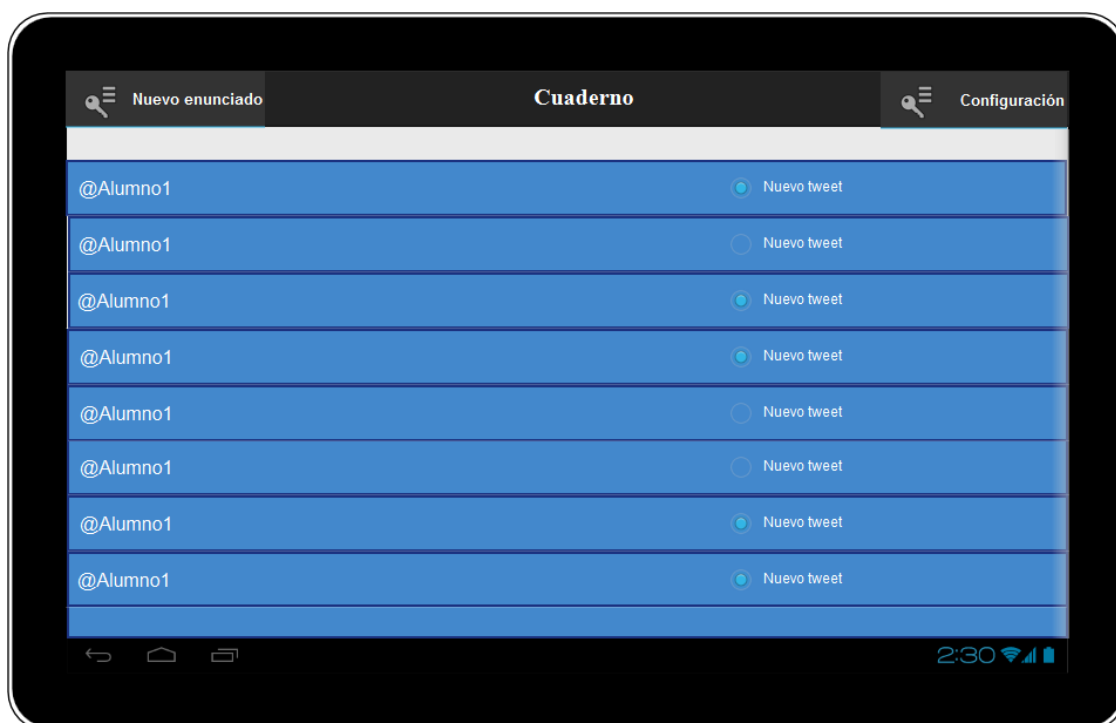


Ilustración 20: Interfaz principal del profesor (v1)

A continuación se muestra la interfaz para iniciar sesión o cambiar de cuenta de twitter.

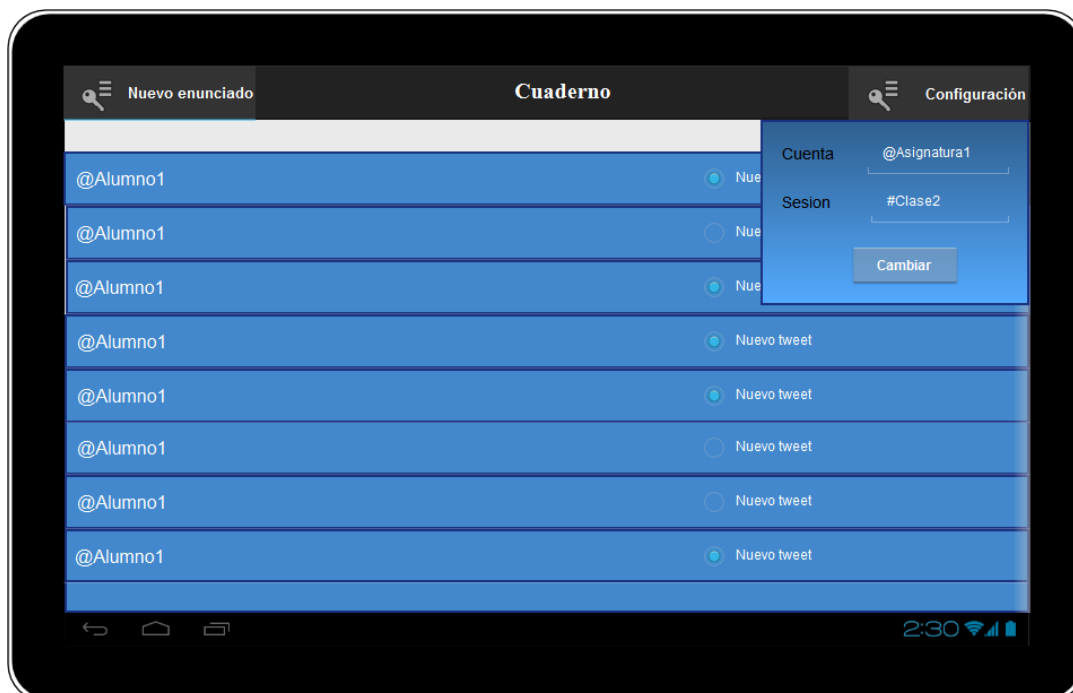


Ilustración 21: Interfaz iniciar sesión del profesor (v1)

En la siguiente interfaz el profesor visualizaría todos los mensajes intercambiados con un alumno seleccionado anteriormente. Como se puede ver el primer mensaje sería el enunciado y el siguiente, la solución del alumno. Además, se mostraría la evaluación del profesor y comentarios relativos a la solución o dudas del alumno.

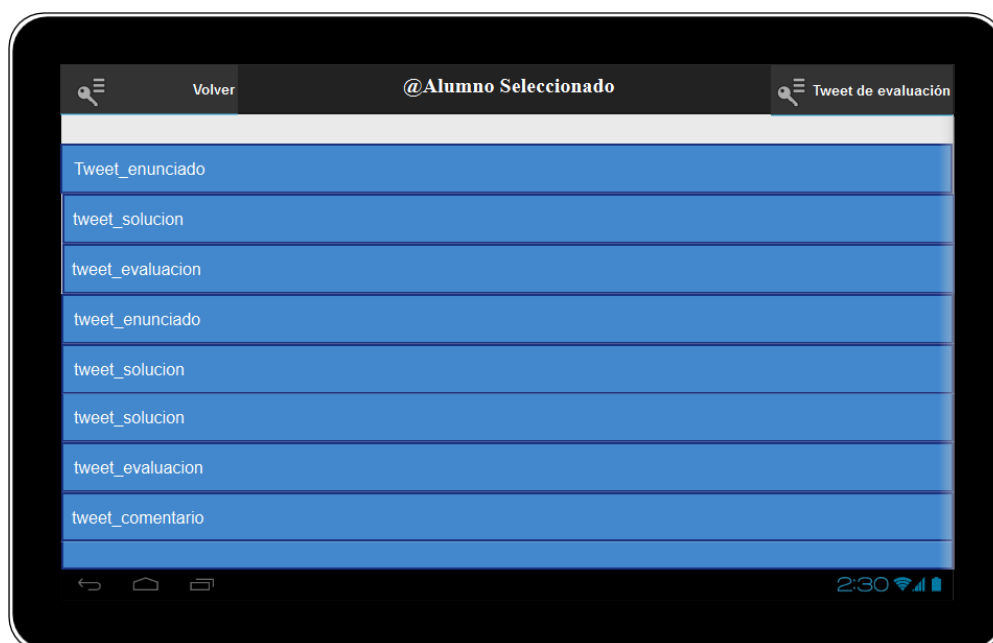


Ilustración 22: Interfaz alumno seleccionado del profesor (v1)

Para subir un enunciado el profesor accedería a la siguiente interfaz:

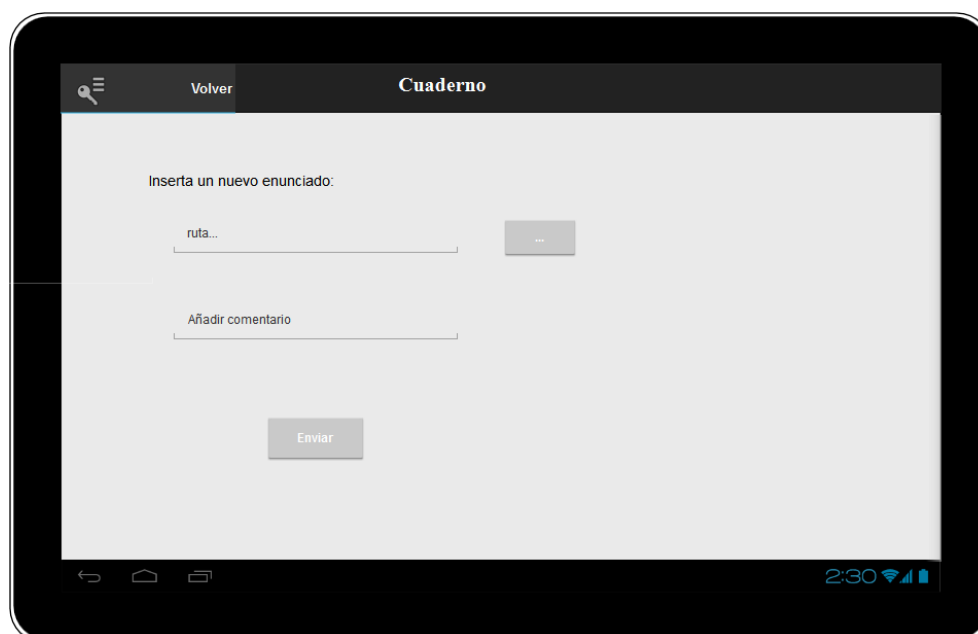


Ilustración 23: Interfaz para subir un enunciado (v1)

Para la primera versión del diseño de la aplicación del alumno, las interfaces son similares a las que tiene el profesor, a excepción de que su pantalla principal mostraría directamente los mensajes intercambiados, en vez de la lista de alumnos que tenía el profesor:

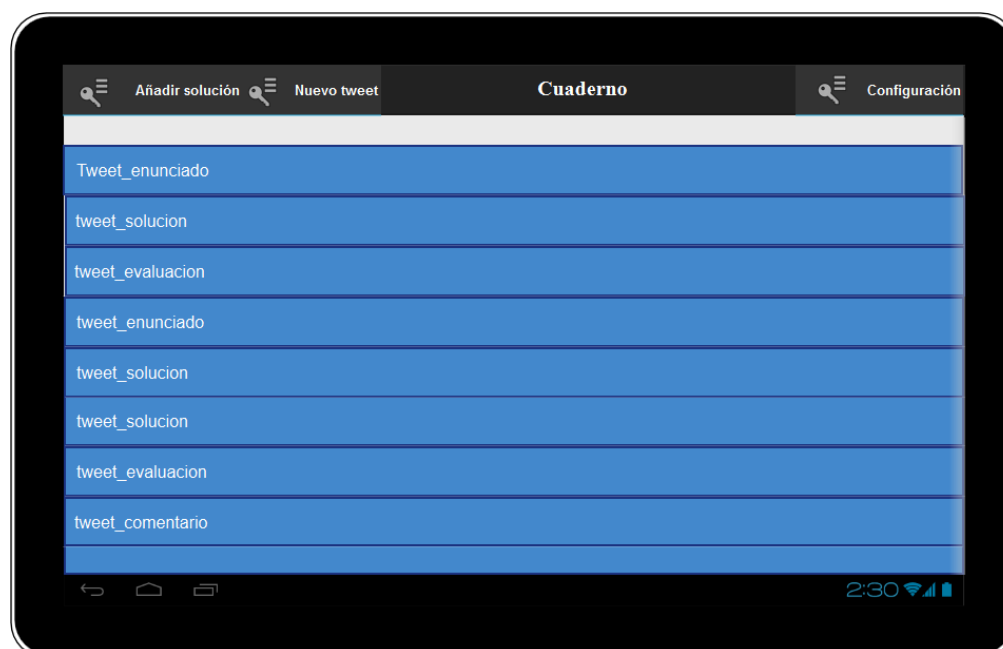


Ilustración 24: Interfaz principal del alumno (v1)

De forma análoga al profesor, el alumno accedería a la siguiente interfaz para enviar una solución al profesor:

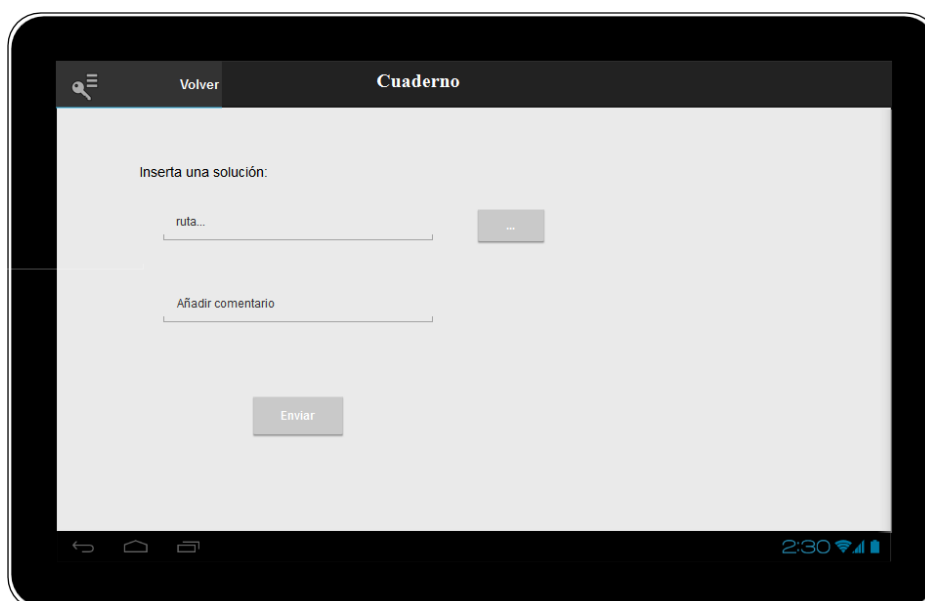


Ilustración 25: Interfaz para subir una solución (v1)

En la siguiente iteración del desarrollo, se diseñó la segunda versión del prototipo. En esta ocasión, la pantalla principal del profesor sería diferente a la anterior: No mostraría la lista de alumnos al iniciar la aplicación, sino un listado con todos los ejercicios que ha subido:



Ilustración 26: Interfaz principal del profesor (v2)

Al seleccionar un ejercicio en concreto, el profesor accedería a la siguiente interfaz para elegir un alumno. El resto de interfaces serían similares a las de la primera versión.



Ilustración 27: Interfaz para seleccionar alumno (v2)

Respecto a la aplicación del alumno, no habría ningún cambio entre la primera versión y la segunda. Sin embargo, se realizó un tercer diseño para el prototipo en ambas aplicaciones que sería el definitivo y la base para iniciar la implementación del proyecto.

El principal cambio es que en este diseño la distribución de la interfaz principal es completamente diferente, ya que ahora se divide en dos partes: En la de la izquierda se muestra tanto los alumnos como los enunciados, mientras que en la derecha, se visualiza el contenido del fichero en cuestión, de esta forma se aprovecha mucho mejor el espacio del dispositivo.

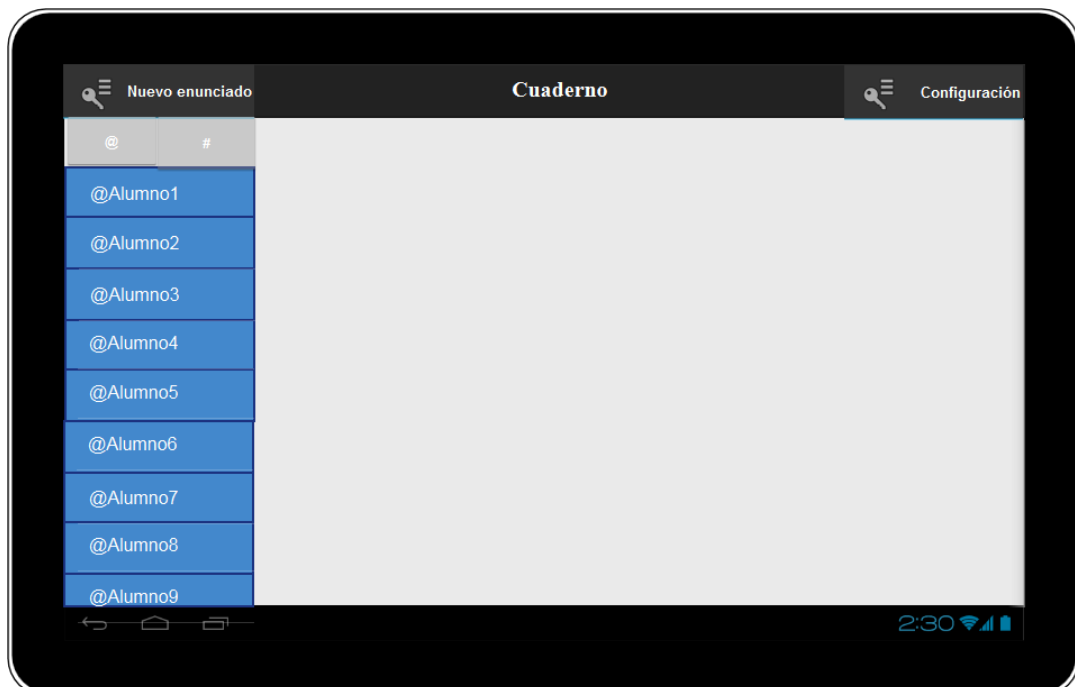


Ilustración 28: Interfaz principal del profesor I (v3)

Dependiendo de qué pestaña seleccione el profesor, se mostrará la lista de alumnos o la lista de ejercicios como se muestra en la siguiente ilustración:

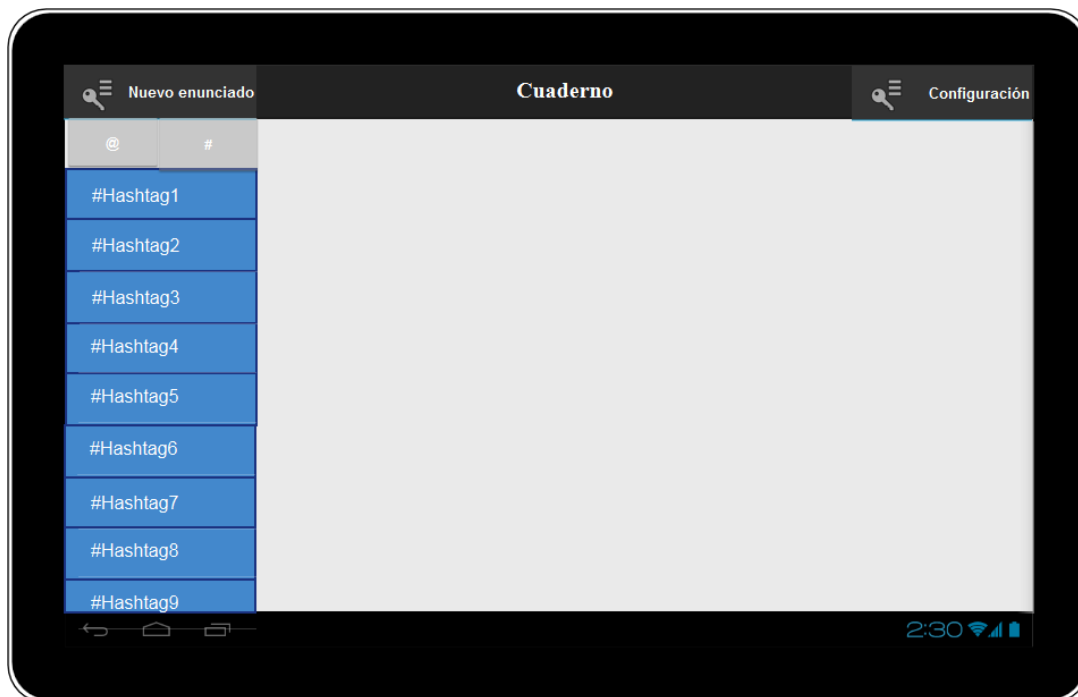


Ilustración 29: Interfaz principal del profesor II (v3)

Si selecciona un ejercicio, en la parte de la izquierda se cargan los alumnos que están en el curso, mientras que en el resto de la pantalla se muestra el enunciado del ejercicio seleccionado:

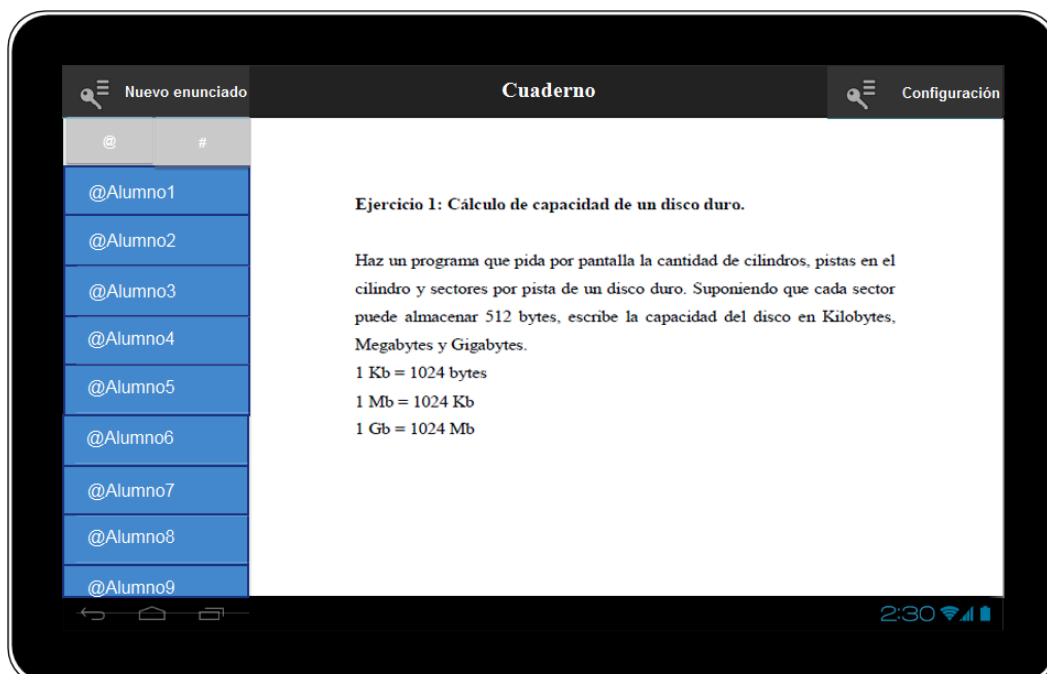


Ilustración 30: Interfaz del profesor al seleccionar un enunciado (v3)

Si por el contrario, selecciona un alumno de la lista, la aplicación mostrará el contenido de la solución que haya enviado dicho alumno. Como se puede ver se ha añadido un botón para volver a ver el enunciado de un ejercicio si se desea.

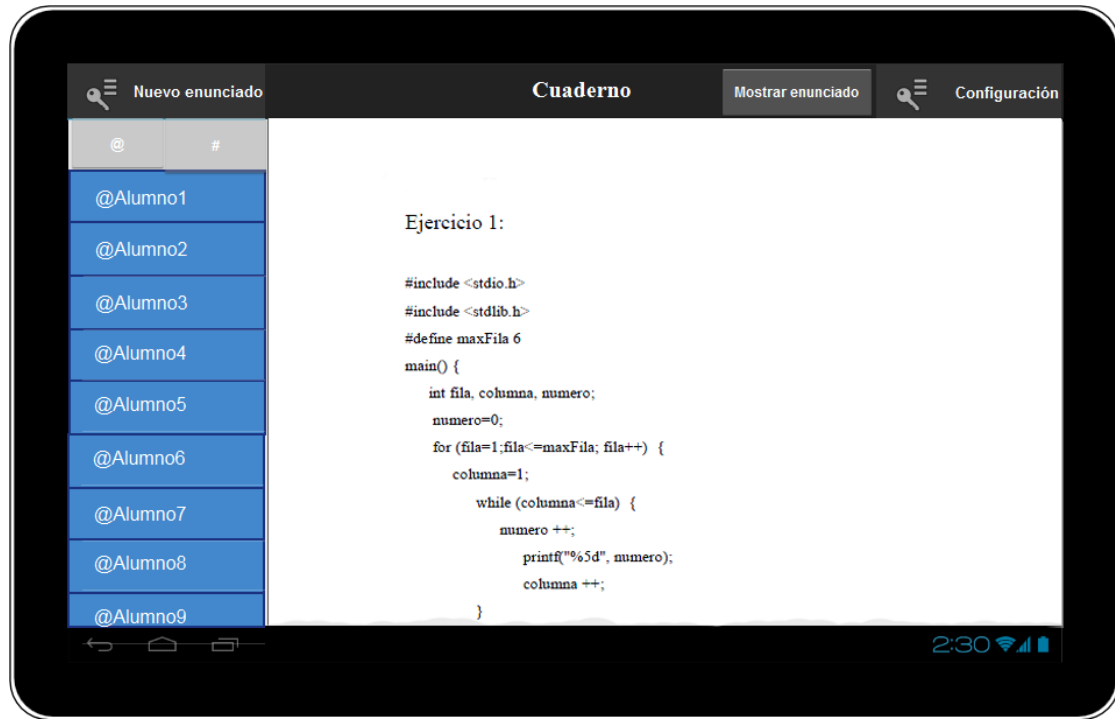


Ilustración 31: Interfaz del profesor al seleccionar un alumno (v3)

Por último, en este diseño, la aplicación recuperaría las cuentas de Twitter que el usuario tenga registradas en la Tablet para mostrarlas en una ventana emergente para que el profesor pueda seleccionar una de ellas:



Ilustración 32: Interfaz del profesor para cambiar de cuenta (v3)

El diseño relativo a la aplicación del alumno es análogo al del profesor con la diferencia de que en la parte de la izquierda de la interfaz principal siempre se muestran los ejercicios disponibles por el profesor, y al pulsar sobre uno, la parte de la derecha mostraría el enunciado del ejercicio:

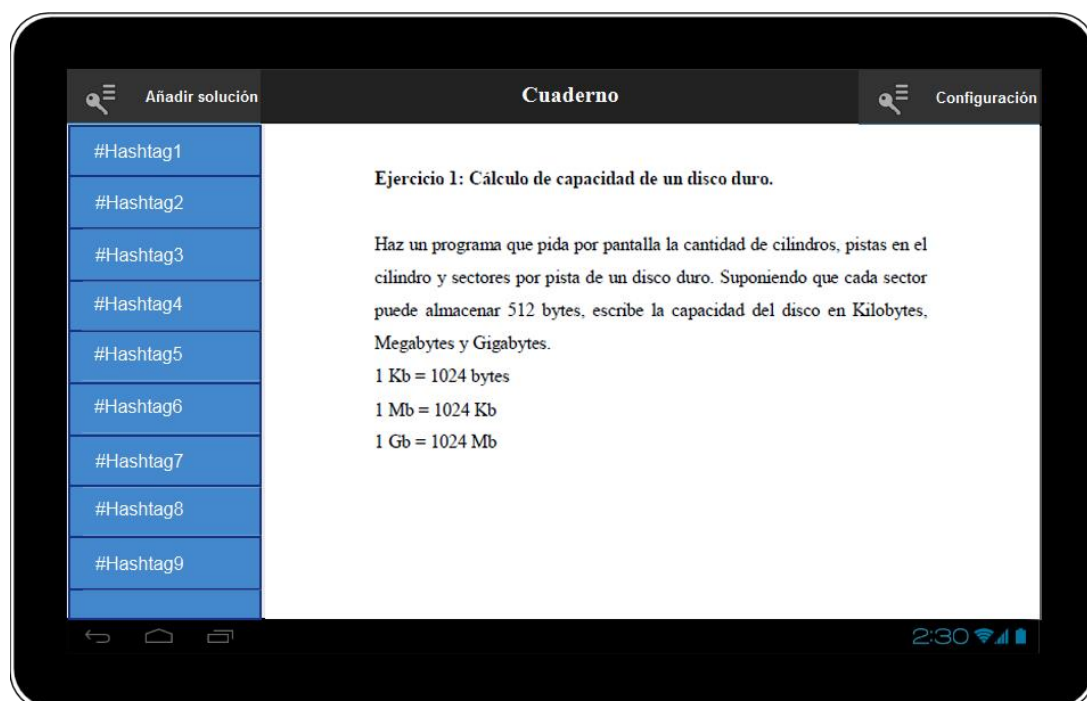


Ilustración 33: Interfaz principal del alumno (v3)

4.2 Alternativas de diseño

En esta sección se van a detallar las diferentes alternativas de diseño que se han tenido en cuenta durante el desarrollo del proyecto argumentando las ventajas e inconvenientes de cada opción y los motivos por los cuales se ha elegido una alternativa en concreto.

4.2.1 Código LMS

Uno de los primeros dilemas que han surgido durante la realización del diseño es el de la ubicación del código que permite la conexión y la comunicación entre la aplicación y el servidor de almacenamiento de los enunciados y ejercicios.

En una primera opción se desarrolló todo el código en una clase propia de la aplicación. Como ventaja se puede decir que al estar alojada dentro del dispositivo, la aplicación no tiene que depender de otro sistema externo, por lo que el tiempo de respuesta de cada petición es mucho menor. A pesar de este aspecto positivo, esta alternativa implica que la aplicación requiera más espacio en el dispositivo, y además, si se desea realizar algún cambio en el LMS, esto implicaría modificar toda la aplicación y publicar una nueva versión de la misma.

La segunda opción, que es la que finalmente se ha implementado en el diseño final, es la de alojar el código relativo al LMS en un sistema externo a la aplicación. En este caso, se ha optado por el desarrollo de un servicio web a través de la plataforma Play Framework que permite alojar e implementar aplicaciones web con Java.

Es decir, este framework permite tener el código que realiza las peticiones al servidor de almacenamiento en la nube y separado del resto de código correspondiente a la aplicación. La principal ventaja que posee esta alternativa es que al crear una capa independiente se pueden realizar cambios en el servicio web sin que por ello se tenga que modificar la aplicación, y por otro lado, el tamaño final de la aplicación va a ser mucho menor. Como desventajas se puede mencionar que al estar en la nube, el tiempo de respuesta va a verse afectado por las condiciones de

la red, por lo que el tiempo de ejecución puede verse incrementado, y que se requiere de un servidor para alojar dicho servicio web.

4.2.2 Autenticación Twitter

Otro aspecto del diseño que ofrecía varias opciones es el de la manera en que un usuario inicia sesión y se autentica utilizando el id de Twitter. La primera alternativa del diseño fue la de hacer uso de la librería Twitter4J y redireccionar al usuario a través del navegador del dispositivo al sitio web de Twitter para introducir los datos requeridos.

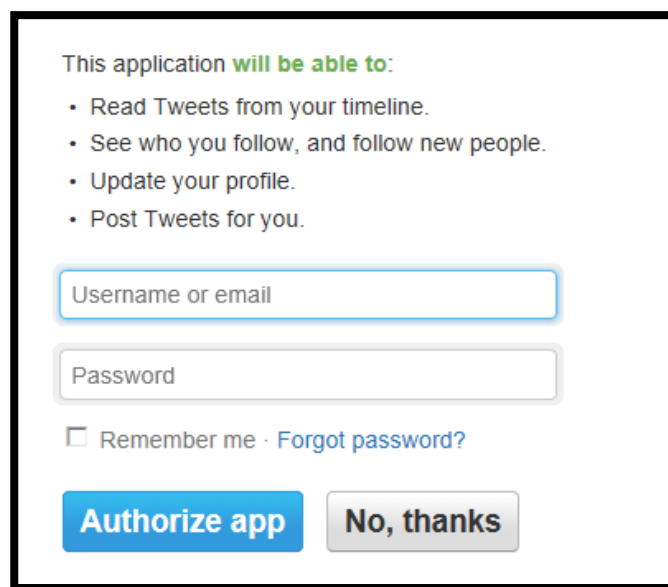
The image shows a Twitter authorization dialog box. At the top, it says "This application will be able to:" followed by a list of permissions: "Read Tweets from your timeline.", "See who you follow, and follow new people.", "Update your profile.", and "Post Tweets for you.". Below the list are two input fields: "Username or email" and "Password". Under the password field is a checkbox labeled "Remember me" and a link "Forgot password?". At the bottom are two buttons: "Authorize app" (blue) and "No, thanks" (gray).

Ilustración 34: Autenticación externa de Twitter

Esto implica que el usuario tiene que acceder a la aplicación, después ir al navegador, y volver de nuevo a la aplicación. Esto puede ser un poco tedioso, además, el hecho de acceder a un recurso externo y que no sea transparente al usuario no es funcional.

La segunda opción consistía en que la aplicación acceda a los datos de las cuentas que el usuario ha registrado previamente en el dispositivo, y mostrarlo en la interfaz para que el usuario seleccione la deseada.

4.2.3 Formato de los datos

A la hora de almacenar o transferir datos de la aplicación al servicio web y viceversa había que elegir que lenguaje utilizar para organizar y manipular la información solicitada por las diferentes peticiones. Las dos alternativas que se barajaron fueron XML y JSON.

Las ventajas del primero es que es un lenguaje flexible, y es mucho más sencillo de leer para una persona. Además, al utilizar etiquetas permite añadir diferentes atributos a los campos complementando la información. Sin embargo, se ha decidido por JSON porque es mucho más sencillo y más manejable cuando la cantidad de información a representar es pequeña como es este caso. No añade cabeceras como hace XML que incrementan el volumen de los datos a transferir y por otro lado, al no utilizar apertura y cierre de etiquetas, mediante JSON se consigue transferir el mismo volumen de información en ficheros mucho más pequeños y manejables, como se puede apreciar en los dos ejemplos siguientes:

```
{
  "Alumnos": [
    { "Nombre": "John" , "Apellidos": "Doe", "Twitter": "JDoe" },
    { "Nombre": "Anna" , "Apellidos": "Smith", "Twitter": "ASmith" },
    { "Nombre": "Peter" , "Apellidos": "Jones", "Twitter": "PJones" },
    { "Nombre": "Bart" , "Apellidos": "Simpson", "Twitter": "BSimpson" },
    { "Nombre": "Peter" , "Apellidos": "Parker", "Twitter": "Spiderman" },
    { "Nombre": "Tony" , "Apellidos": "Stark", "Twitter": "Ironman" },
  ]
}
```

Ilustración 35: Ejemplo fichero JSON


```

<alumnos>
  <alumno>
    <Nombre>John</Nombre> <Apellidos>Doe</Apellidos> <Twitter>JDoe</Twitter>
  </alumno>
  <alumno>
    <Nombre>Anna</Nombre> <Apellidos>Smith</Apellidos> <Twitter>ASmith</Twitter>
  </alumno>
  <alumno>
    <Nombre>Peter</Nombre> <Apellidos>Jones</Apellidos> <Twitter>PJones</Twitter>
  </alumno>
  <alumno>
    <Nombre>Bart</Nombre> <Apellidos>Simpson</Apellidos> <Twitter>BSimpson</Twitter>
  </alumno>
  <alumno>
    <Nombre>Peter</Nombre> <Apellidos>Parker</Apellidos> <Twitter>Spiderman</Twitter>
  </alumno>
  <alumno>
    <Nombre>Tony</Nombre> <Apellidos>Stark</Apellidos> <Twitter>Ironman</Twitter>
  </alumno>
</alumnos>

```

Ilustración 36: Ejemplo fichero XML

4.2.4 Estructura de los contenidos

Diseñar la estructura de las carpetas que almacenan los diferentes recursos es uno de los pilares más importantes del proyecto, ya que dependiendo de lo que se elija el tiempo de ejecución puede variar.

Una de las alternativas consistía en la creación de una carpeta por cada alumno que tuviera acceso al curso, almacenando en ellas el enunciado del ejercicio y la solución de dicho alumno. Tras analizar esta opción se demostró que, aunque la organización es válida, se creaban muchas carpetas de las cuales algunas sólo contenían el enunciado del ejercicio y dicho fichero se almacenaba tantas veces como carpetas se hubieran creado. Este sistema requería de mayor tiempo de cómputo ya que para conocer qué alumnos habían respondido a un ejercicio había que acceder y leer el contenido de cada carpeta.

Por lo que después de hacer algunas pruebas se decidió implementar finalmente la segunda opción. En esta, en vez de crear una carpeta por cada alumno, se crea una carpeta por ejercicio. De esta manera, solo se almacena el enunciado de dicho ejercicio una sola vez, y es mucho más sencillo recuperar la información del

servidor. Por ejemplo, para listar los ejercicios disponibles solo hay que recuperar el nombre de las carpetas sin acceder a ellas, y para saber qué alumnos han respondido a un ejercicio en concreto, simplemente hay que acceder a la carpeta que aloje ese ejercicio y recuperar el nombre de todos los ficheros que contengan la solución del alumno.

4.3 Diseño final de la aplicación

Tras estudiar los puntos más favorables de cada una de las alternativas de diseño, se ha decidido finalmente por una implementación formada por capas bien diferenciadas como se indica en la ilustración siguiente:



Ilustración 37: Diseño final de la aplicación

Para la autenticación del usuario se hará uso de la red social Twitter, accediendo a los datos registrados en el dispositivo por dicho sistema. La implementación del LMS se realizará en un servicio web ubicado en un servidor externo, al cual la aplicación podrá enviar peticiones y recibir la información requerida según la funcionalidad.

Por último, se requiere de otro servidor diferente que hará las funciones de repositorio de los contenidos compartidos por el profesor y accesible para los alumnos. En este caso, utiliza el protocolo FTP para transferir los datos desde el dispositivo al servidor y viceversa.

A continuación se muestran las interfaces de la aplicación finalizada. En la siguiente ilustración se visualiza la pantalla principal que verá el profesor al iniciar sesión. Dos pestañas donde puede elegir entre ver el listado de alumnos o ver un listado de ejercicios dependiendo que pestaña seleccione. En este primer caso se muestra la pestaña de Alumnos activada, por lo que la aplicación recupera un listado de los alumnos que siguen este curso.

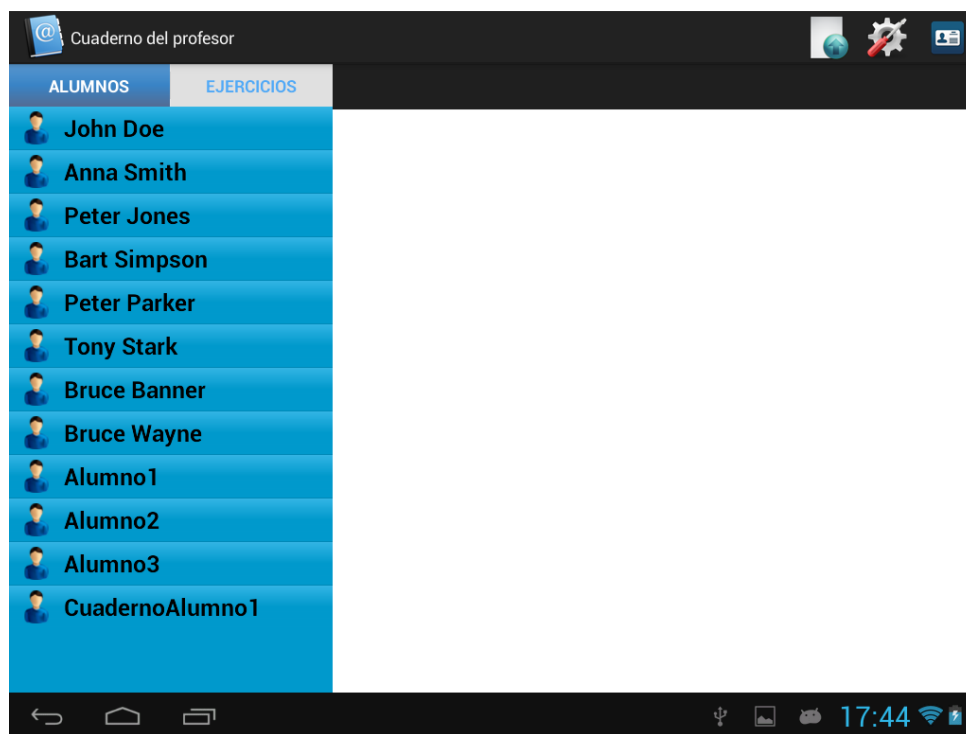


Ilustración 38: Interfaz principal listado de alumnos

Si por el contrario, selecciona la pestaña de Ejercicios, en la parte de la izquierda de la pantalla se mostrarán todos los ejercicios que el profesor haya compartido con anterioridad:

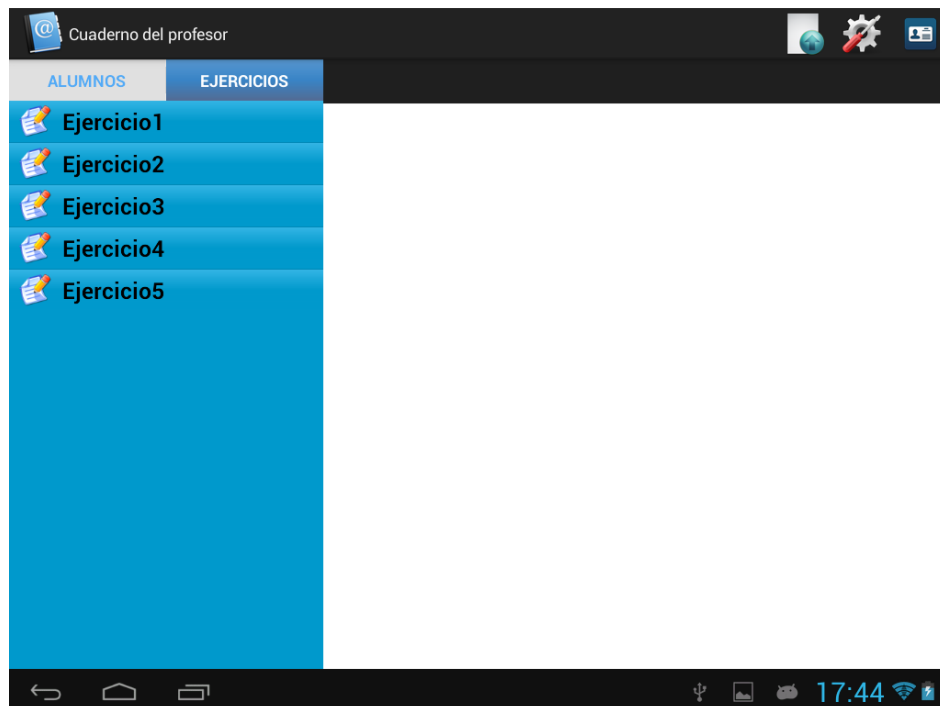


Ilustración 39: Interfaz principal listado de ejercicios

En la siguiente ilustración se puede ver la interfaz al seleccionar un ejercicio, en este caso “Ejercicio1”, de la pantalla anterior. En la parte de la derecha se visualiza el contenido del enunciado de ese ejercicio, mientras que en la derecha se notifican qué alumnos han contestado y cuáles no.



Ilustración 40: Interfaz principal alumnos que han resuelto un ejercicio

En la siguiente ilustración se muestra el contenido de la solución al ejercicio seleccionado al pulsar sobre uno de los alumnos que sí han enviado una solución al profesor.

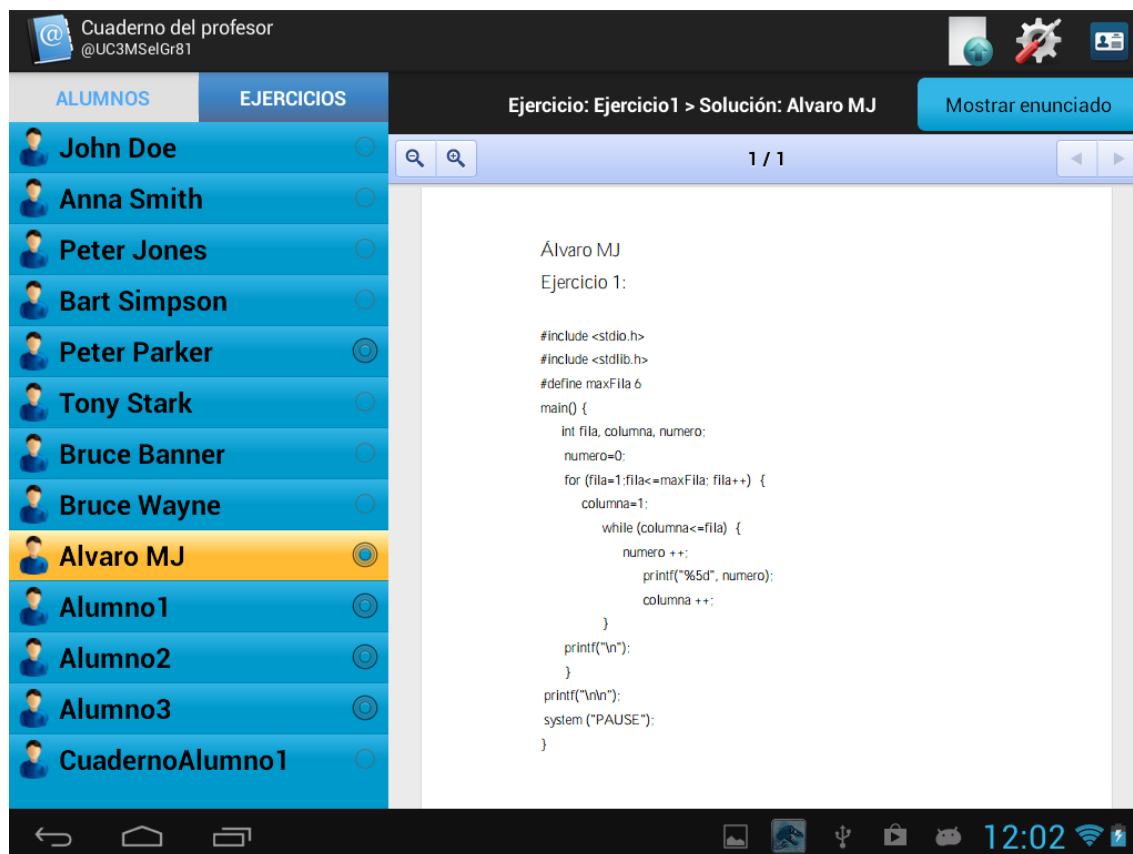


Ilustración 41: Mostrar solución de un alumno

En la siguiente ilustración se muestra la interfaz, donde el profesor o el alumno pueden alternar entre las diferentes identidades sociales que están registradas en su dispositivo, a través de un pop-up central.

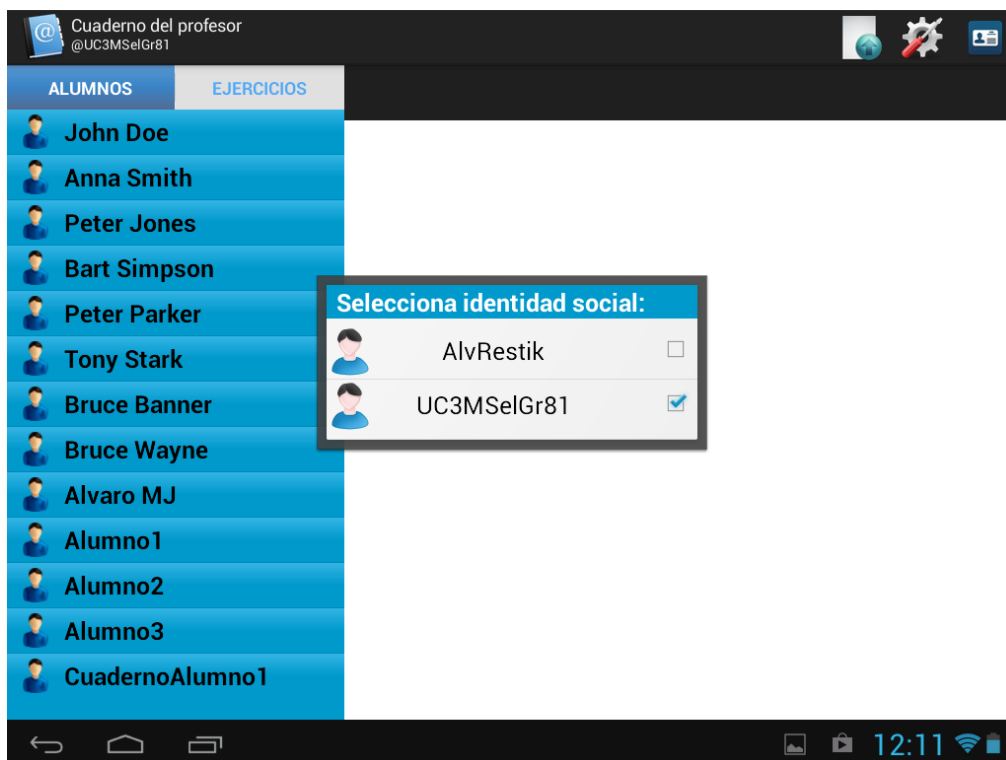


Ilustración 42: Cambiar entre identidades de Twitter

La ilustración 40 muestra la interfaz inicial que visualiza un alumno al iniciar la aplicación. Aparece en un listado todos los ejercicios disponibles y se notifica cuáles de ellos, el propio alumno, ya ha enviado una solución.

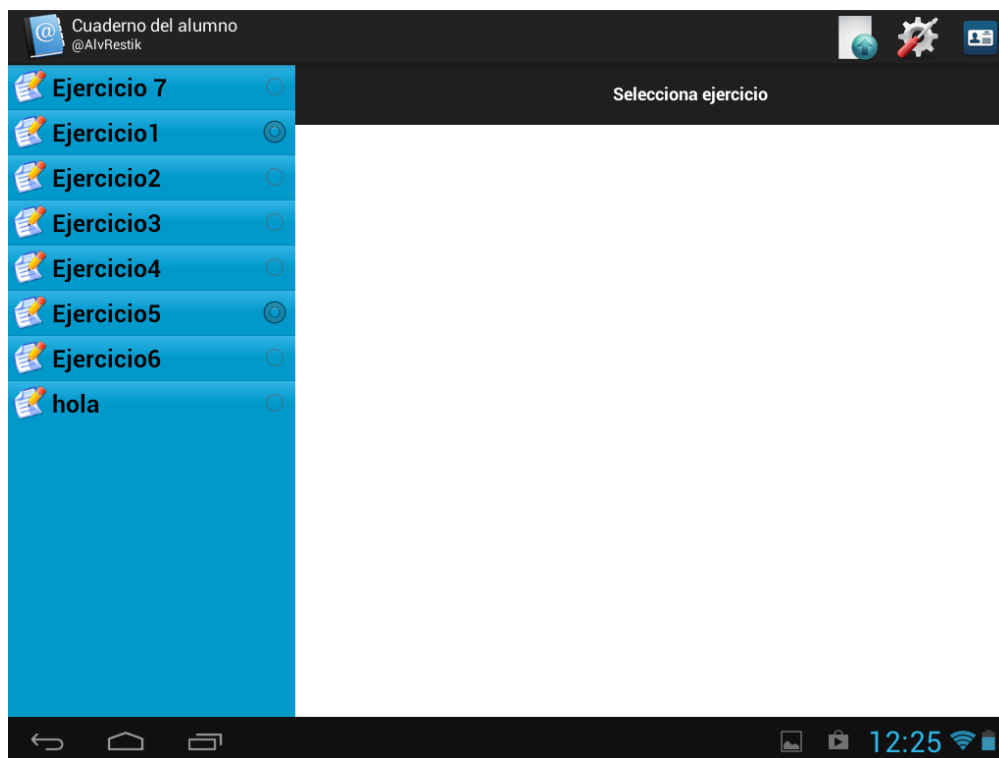


Ilustración 43: Interfaz principal del alumno

En la siguiente figura el alumno puede seleccionar entre las asignaturas que haya añadido con anterioridad a la aplicación.

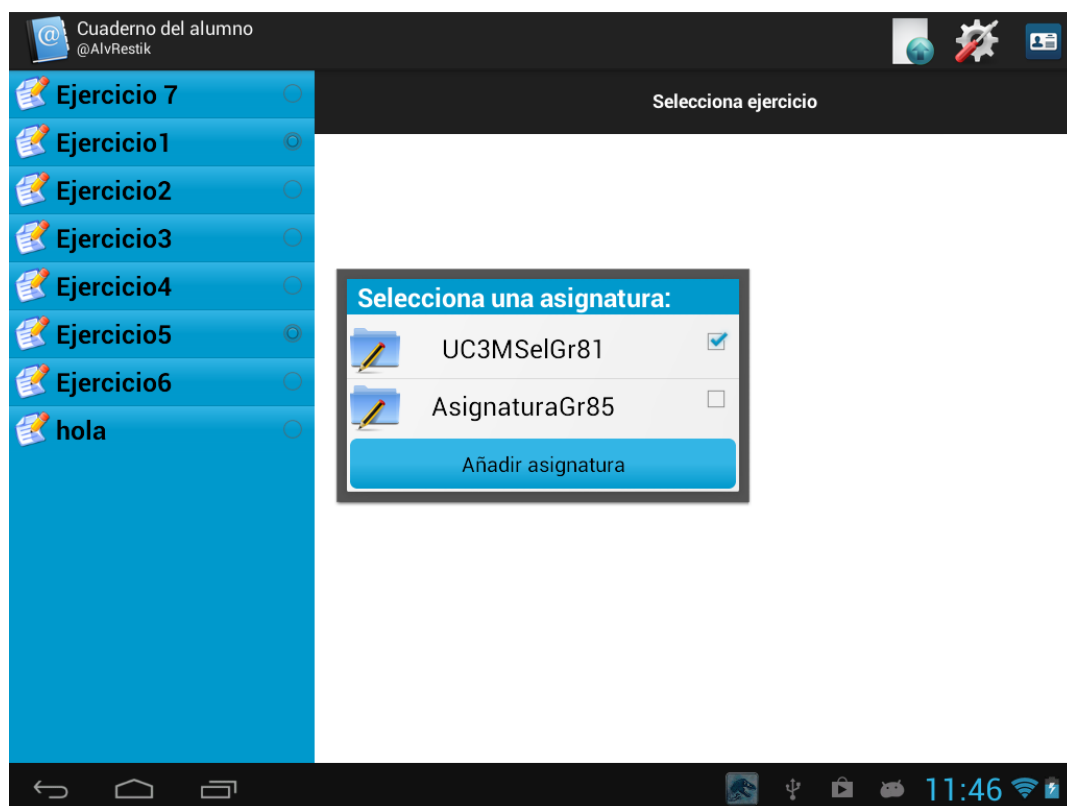


Ilustración 44: Interfaz para cambiar entre asignaturas

En la ilustración 42 se muestra la interfaz que se carga en la pantalla para añadir un nuevo enunciado al servidor de contenidos, para que posteriormente esté disponible para que tanto el profesor como los alumnos puedan acceder a él.

< @ Añadir enunciado

Título del ejercicio:

Seleccionar enunciado:

Ruta del fichero _____ Buscar

Enviar

← ↗ 📷 12:18 📶 🔋

Ilustración 45: Interfaz para añadir un enunciado

En la última figura se muestra la interfaz de la aplicación del alumno que permite seleccionar un fichero y enviarlo como solución al repositorio.

< @ Enviar solución

Selecciona un ejercicio:

Ejercicio1 ▾

Seleccionar solución:

Ruta del fichero _____ Buscar

Enviar

← ↗ 📷 12:18 📶 🔋

Ilustración 46: Interfaz para añadir una solución

4.4 Diagramas de clase

En este apartado se describe el diagrama de clases correspondiente al diseño establecido para la implementación de la aplicación. En un primer lugar se especifica mediante un diagrama las relaciones de las clases para dar una visión general. Posteriormente se describe cada una de las clases existentes con sus atributos y métodos. Mencionar que al haberse desarrollado dos aplicaciones, se muestra, en primer lugar, el diagrama de clases de la aplicación del profesor y, a continuación, la del alumno, ya que aunque sigue una estructura similar existen algunas diferencias.

4.4.1 Aplicación del profesor

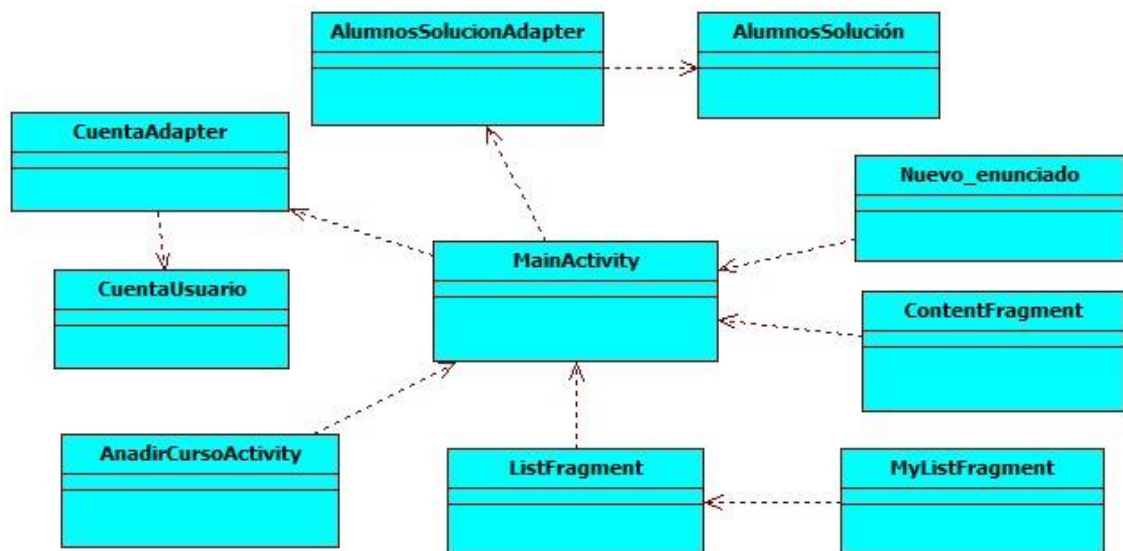


Ilustración 47: Diagrama de clases Profesor

A continuación se describen cada una de las clases de manera detallada:

Esta clase permite personalizar la forma en que se muestra un ítem en un ListView. En este caso, se aplica para mostrar y notificar qué alumnos han enviado una solución a un ejercicio seleccionado.

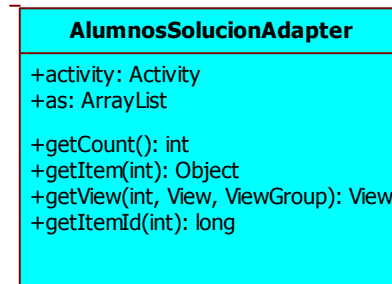


Ilustración 48: Clase AlumnosSolucionAdapter

Como la clase anterior, mediante `CuentaAdapter`, se personaliza la forma de mostrar en un `ListView` las cuentas de un usuario.

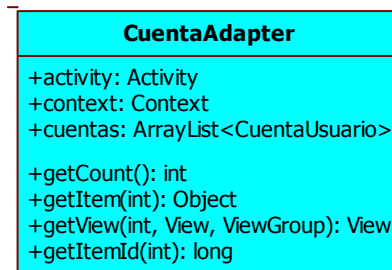


Ilustración 49: Clase CuentaAdapter

Mediante esta clase se crean objetos de tipo `CuentaUsuario` con los que recuperar los datos de una cuenta Twitter y gestionar cual es la que está haciendo uso el alumno.

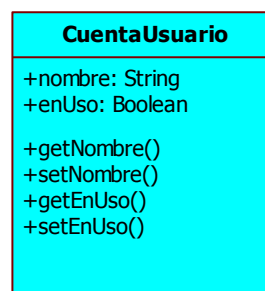


Ilustración 50: Clase CuentaUsuario

La siguiente clase define un objeto de tipo `AlumnosSolucion` y permite gestionar el nombre del ejercicio y de conocer si el alumno ha enviado una solución al mismo.

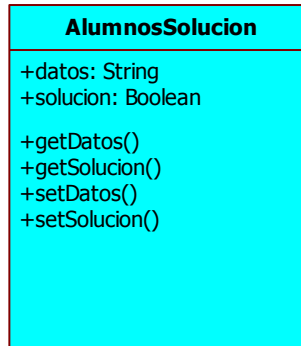


Ilustración 51: Clase AlumnosSolucion

Esta es la clase inicial de la aplicación. Se encarga de mostrar los botones de las diferentes opciones que tiene el usuario en la parte superior de la pantalla, y de cargar los dos fragments que van a formar la interfaz principal, como ya se ha indicado anteriormente.

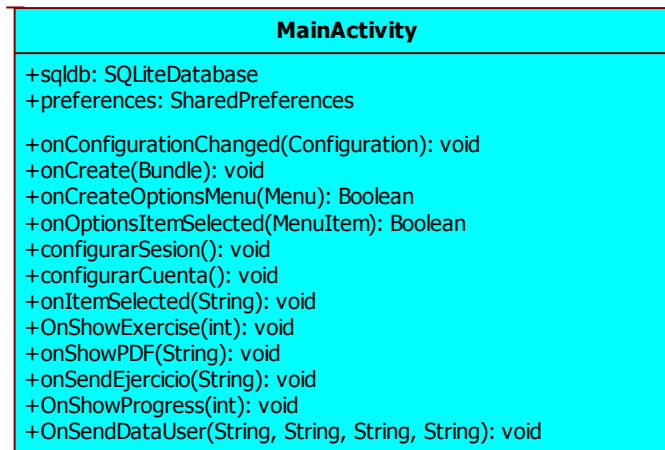


Ilustración 52: Clase MainActivity

La clase ListFragment se encarga de crear y gestionar la parte de la izquierda de la interfaz, es decir, el layout compuesto por las dos pestañas de Alumnos y Ejercicios y el ListView donde se muestra los listados correspondientes.

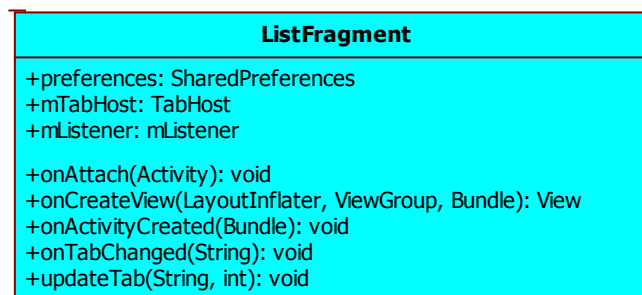


Ilustración 53: Clase ListFragment

La siguiente clase es análoga a la anterior, pero crea y gestiona el resto de la interfaz principal, es decir, donde se muestra el contenido de los enunciados o las soluciones de los ejercicios. Dentro de esta se ha implementado una clase asíncrona para solicitar el fichero en cuestión y volcar su contenido en la interfaz.

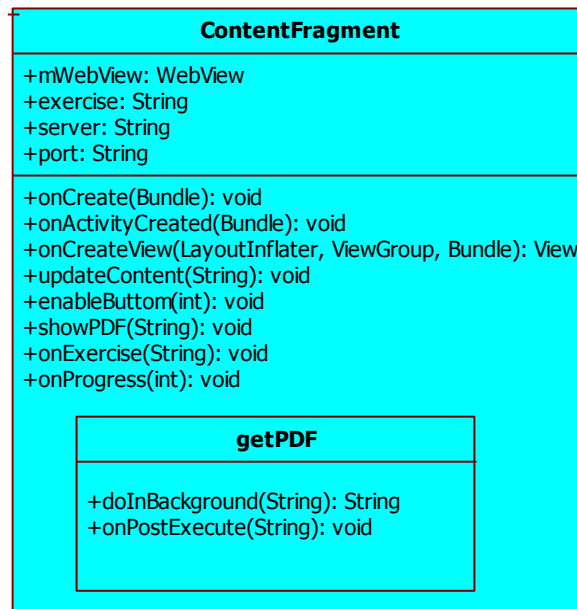


Ilustración 54: Clase ContentFragment

Esta clase se encarga de enviar la petición para enviar un nuevo fichero al servidor. Una de sus funcionalidades es la de buscar y seleccionar un archivo almacenado en el sistema de ficheros del dispositivo. Además contiene una clase asíncrona que permite enviar dicho archivo y realizar la petición correspondiente al servicio web para almacenarlo en el repositorio de recursos.

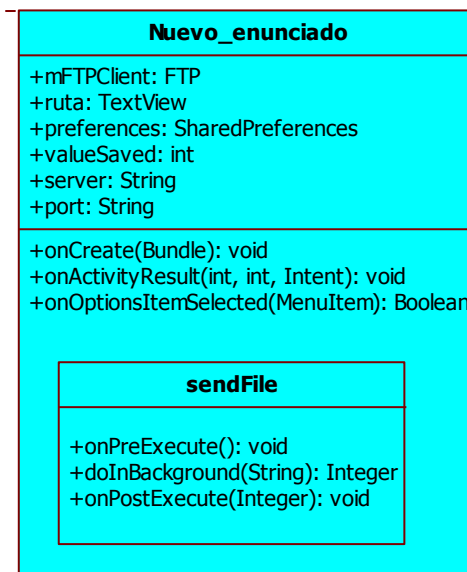


Ilustración 55: Clase Nuevo_enunciado

La clase MyListFragment es la más importante de todas las clases implementadas, ya que se encarga de la mayoría de las funciones esenciales para el correcto funcionamiento de la aplicación. Como se puede observar, además de los métodos comunes en las actividades de una aplicación Android, se han creado en ella nuevas clases que realizan funciones asíncronas, para mostrar los resultados una vez que ha concluido la tarea en cuestión.

- **getPDF:** Envía la petición para recuperar y mostrar el contenido de un enunciado o solución de un ejercicio.
- **getAllfiles:** Permite solicitar la lista de los ejercicios que ha compartido el profesor y mostrarla en el fragmento de interfaz correspondiente.
- **getFiles:** Se encarga de solicitar y almacenar en la aplicación de manera temporal una lista con el nombre de todos los ficheros que ha subido el profesor, para posteriores consultas.
- **getListaAlumnos:** Permite solicitar la lista de los alumnos que siguen un curso y mostrarla para que el profesor pueda seleccionar el que desee posteriormente.
- **getAlumnosSol:** Su funcionalidad consiste en mostrar al profesor que alumnos han contestado a un ejercicio y cuáles no, mediante una notificación visual.

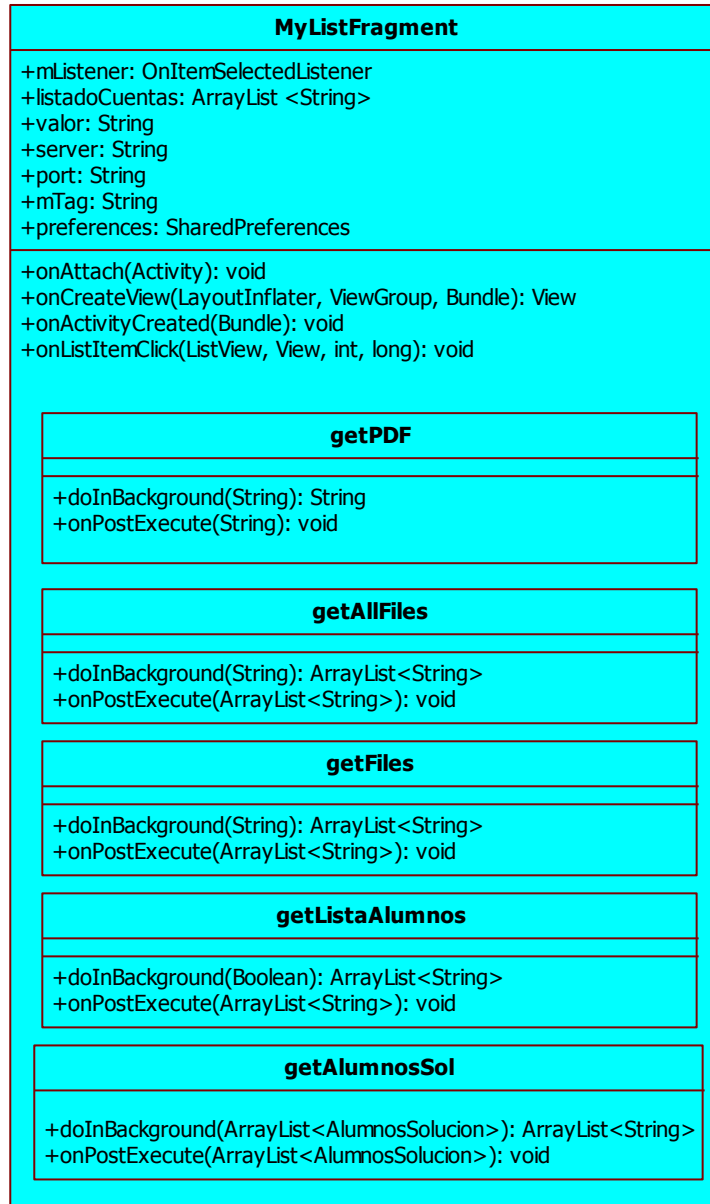


Ilustración 56: Clase MyListFragment

Por último la clase AnadirCursoActivity permite al profesor añadir a la configuración del servicio web los datos correspondientes a un repositorio de contenidos. En ella se ha implementado una clase asíncrona:

- saveData: Envía los datos y la petición al servicio web para que los almacene al fichero de configuración correspondiente.

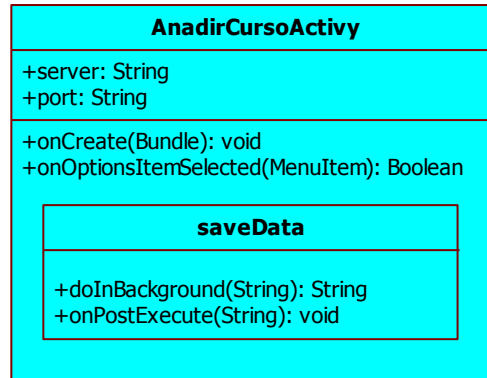


Ilustración 57: Clase AnadirCursoActivy

4.4.2 Aplicación del alumno

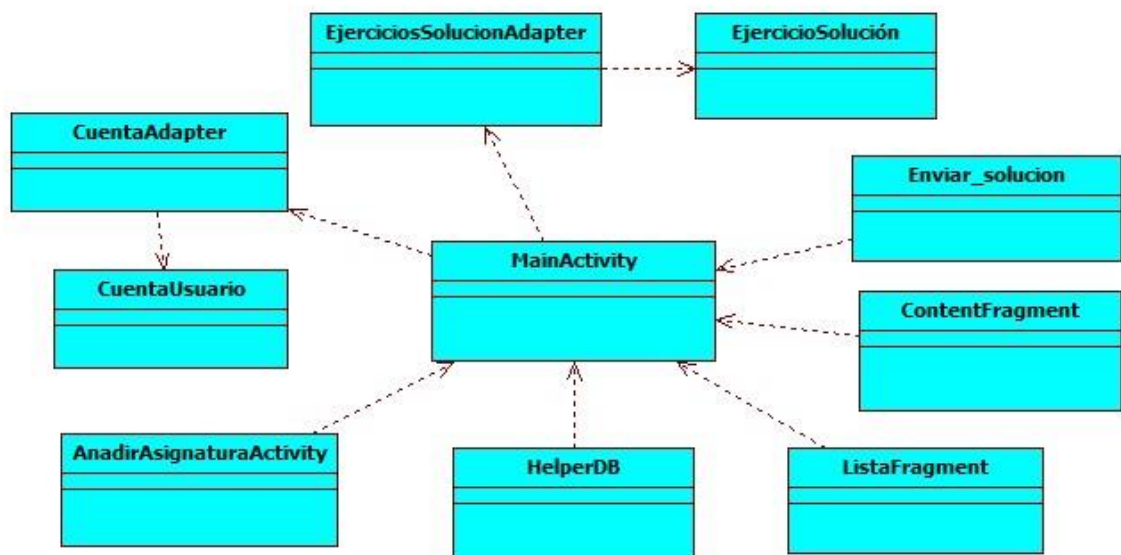


Ilustración 58: Diagrama de clases Alumno

Análogamente, se describen de forma detallada cada una de las clases que forman el diagrama anterior. Esta es la clase inicial de la aplicación. Se encarga de mostrar los botones de las diferentes opciones que tiene el usuario en la parte superior de la pantalla, y de cargar los dos fragments que van a formar la interfaz principal.

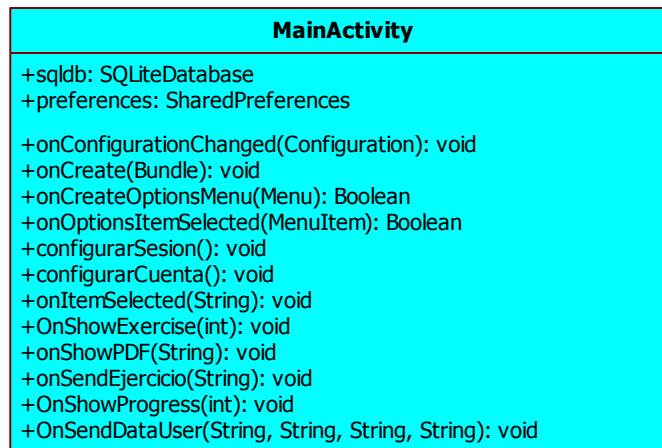


Ilustración 59: Clase MainActivity

La clase EjerciciosSolucionAdapter permite personalizar la forma en que se muestra un ítem en un ListView. En este caso, se aplica para mostrar y notificar qué ejercicios ha resuelto el propio alumno y cuales no ha enviado una solución.

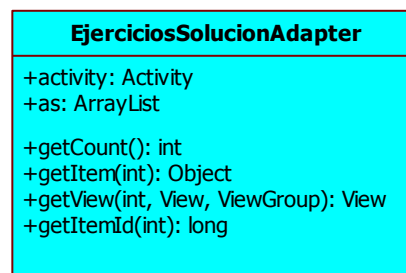


Ilustración 60: Clase EjerciciosSolucionAdapter

La siguiente clase define un objeto de tipo EjercicioSolucion y permite gestionar el nombre del ejercicio y de conocer si el alumno ha enviado una solución al mismo.

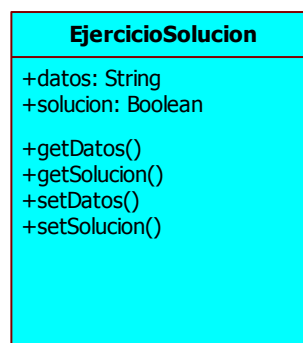


Ilustración 61: Clase EjercicioSolucion

Esta clase permite al alumno enviar una solución a un ejercicio. Además de los métodos comunes a la actividad se han añadido dos clases asíncronas:

- `getFiles`: Recupera un listado con todos los ejercicios disponibles para que seleccione el que desee de entre todas las opciones.
- `sendFile`: Clase que envía la solución al servicio web para que lo almacene en el directorio que corresponda.

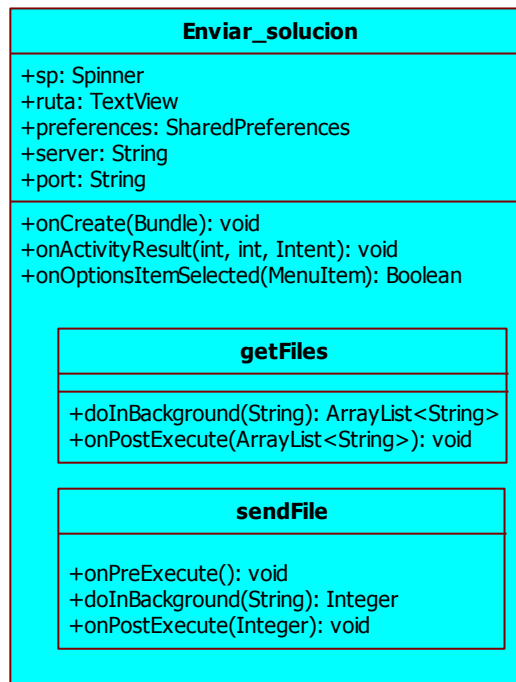


Ilustración 62: Clase `Enviar_solucion`

La clase `ListaFragment` se encarga de crear y gestionar la parte de la izquierda de la interfaz, es decir, el layout compuesto por el `ListView` donde se muestra un listado con los ejercicios que están disponibles. Incluye dos clases asíncronas:

- `getFiles`: Recupera un listado de los ejercicios que ha compartido el profesor anteriormente.
- `getPDF`: Envía la petición para recuperar y mostrar el contenido de un enunciado o solución de un ejercicio.

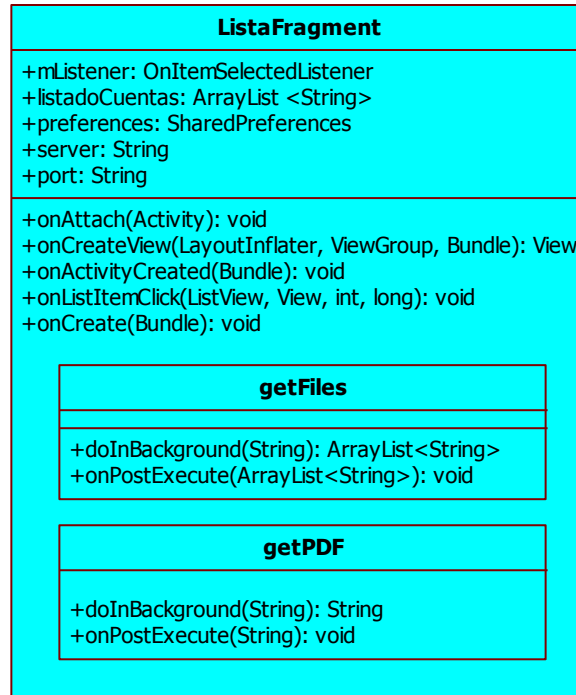


Ilustración 63: Clase ListaFragment

La siguiente clase crea y gestiona la interfaz principal donde se muestra el contenido de los enunciados o las soluciones de los ejercicios. Dentro de esta se ha implementado una clase asíncrona para solicitar el fichero en cuestión y volcar su contenido en la interfaz.

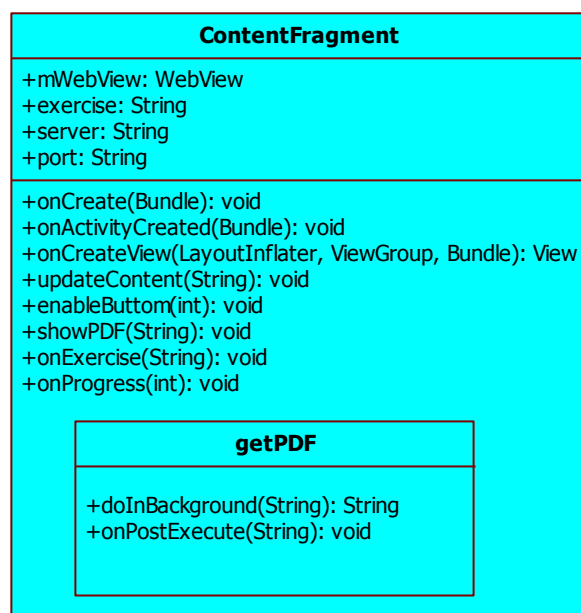


Ilustración 64: Clase ContentFragment

La siguiente clase permite crear y gestionar la base de datos de la aplicación del alumno.

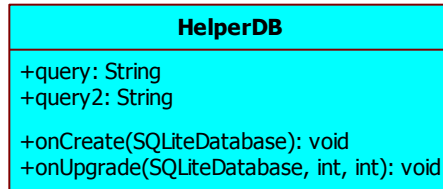


Ilustración 65: Clase HelperDB

La siguiente clase se aplica para personalizar la forma de mostrar en un ListView las cuentas de Twitter de un usuario.

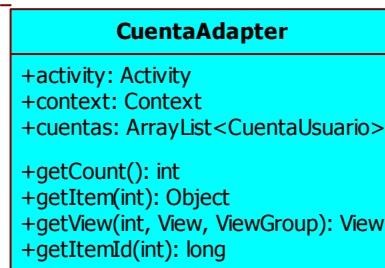


Ilustración 66: Clase CuentaAdapter

Mediante esta clase se crean objetos de tipo `CuentaUsuario` con los que recuperar los datos de una cuenta Twitter y gestionar cual es la que está haciendo uso el alumno.

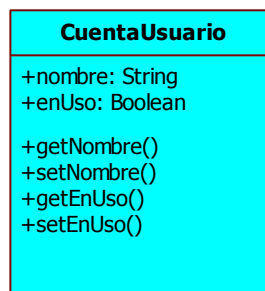


Ilustración 67: Clase CuentaUsuario

Por último, la clase `AnadirAsignaturaActivity` permite al alumno añadir manualmente el nombre de la asignatura para acceder a los ejercicios de clase que comparta el profesor.

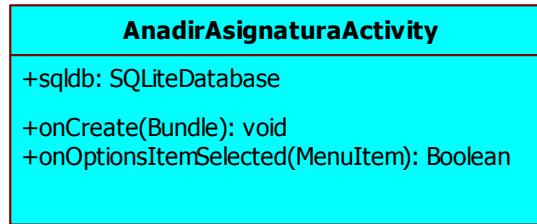


Ilustración 68: Clase AnadirAsignaturaActivity

4.5 Diagramas de secuencia

En este apartado se detallan los diagramas de secuencia de las funcionalidades principales a las que tiene acceso el usuario cuando inicia la aplicación en su dispositivo:

4.5.1 Recuperar listado de alumnos

En el siguiente diagrama de secuencia se muestra las acciones realizadas por el profesor para mostrar el listado de los alumnos que siguen un curso.

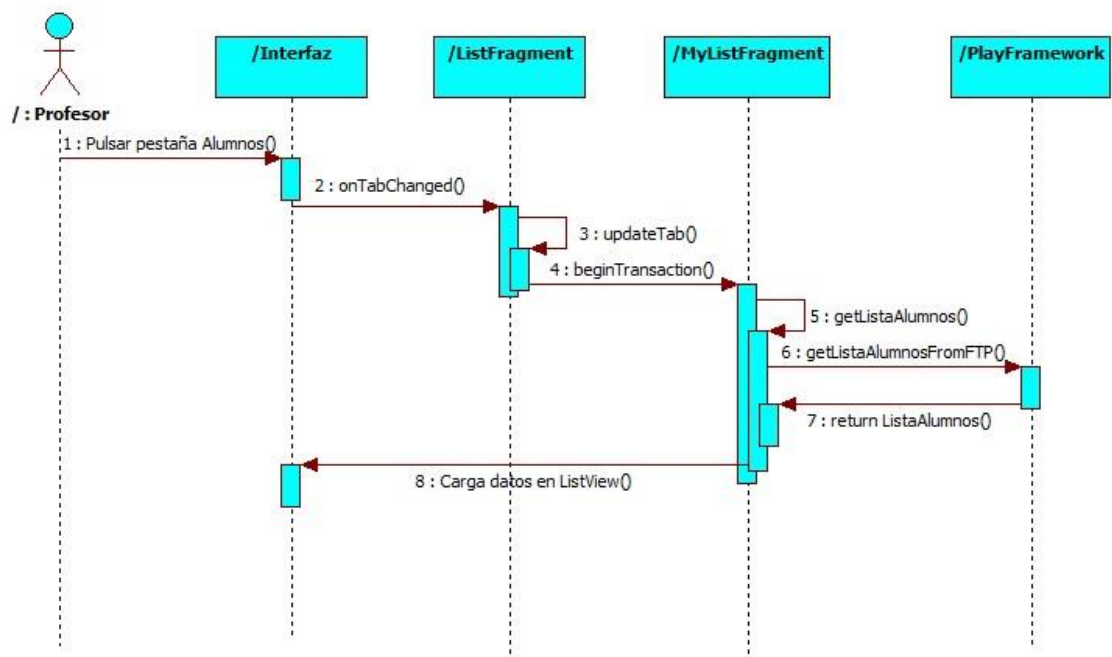


Ilustración 69: Diagrama secuencia Listar alumnos

1. El profesor selecciona la pestaña de Alumnos.
2. En la clase ListFragment se ejecuta el método onTabChanged, que se inicia cada vez que se activa una pestaña.
3. Se ejecuta el método updateTab que se encarga de actualizar el contenido del fragment.
4. Mediante beginTransaction se inicia una instancia de la clase MyListFragment.
5. Como se ha seleccionado la pestaña de Alumnos, al iniciar dicha instancia se ejecuta la clase asíncrona getListaAlumnos.
6. Para enviar la solicitud de recuperar la lista de alumnos se inicia el método getListaAlumnosFromFTP del servicio web.
7. Una vez que el servicio web ha finalizado con la tarea envía los datos a la aplicación.
8. Tras recibir los datos la clase asíncrona carga la información y la muestra en un listado.

4.5.2 Recuperar listado de ejercicios

En el siguiente diagrama de secuencia se muestra las acciones realizadas para mostrar el listado de los ejercicios que ha enviado el profesor.

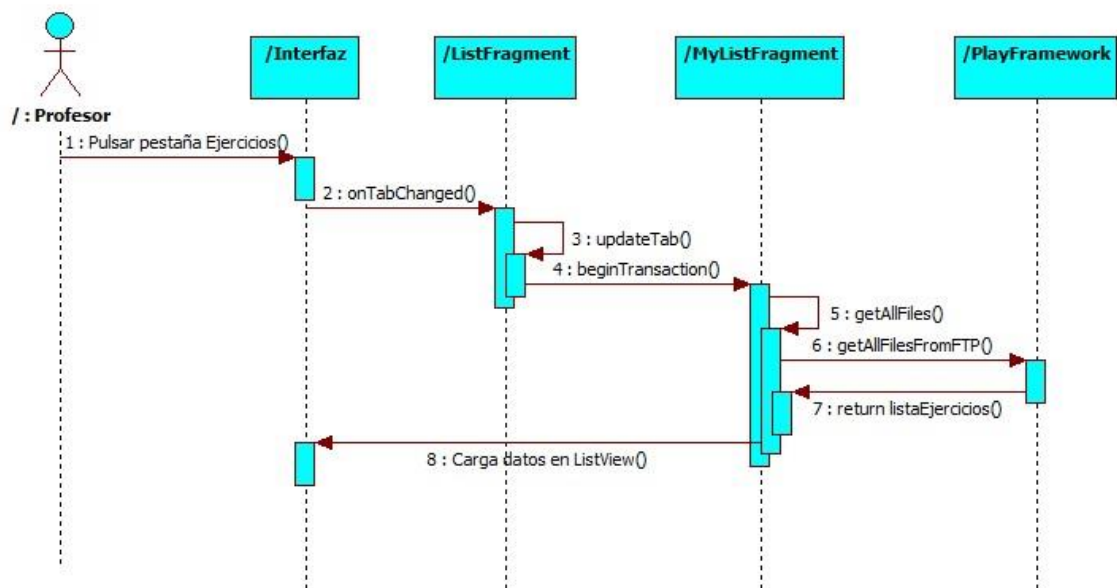


Ilustración 70: Diagrama de secuencia Listar ejercicios

1. El profesor selecciona la pestaña de Ejercicios.
2. En la clase ListFragment se ejecuta el método onTabChanged, que se inicia cada vez que se activa una pestaña.
3. Se ejecuta el método updateTab que se encarga de actualizar el contenido del fragment.
4. Mediante beginTransaction se inicia una instancia de la clase MyListFragment.
5. Como se ha seleccionado la pestaña de Ejercicios, al iniciar dicha instancia se ejecuta la clase asíncrona getAllFiles.
6. Para enviar la solicitud de recuperar la lista de ejercicios se inicia el método getAllFilesFromFTP del servicio web.
7. Una vez que el servicio web ha recuperado el nombre de todos los ejercicios envía los datos a la aplicación.
8. Tras recibir los datos la clase asíncrona carga la información y la muestra en un listado.

4.5.3 Añadir un nuevo ejercicio

En el siguiente diagrama de secuencia se muestra las acciones realizadas para enviar un nuevo enunciado accesible para los alumnos.

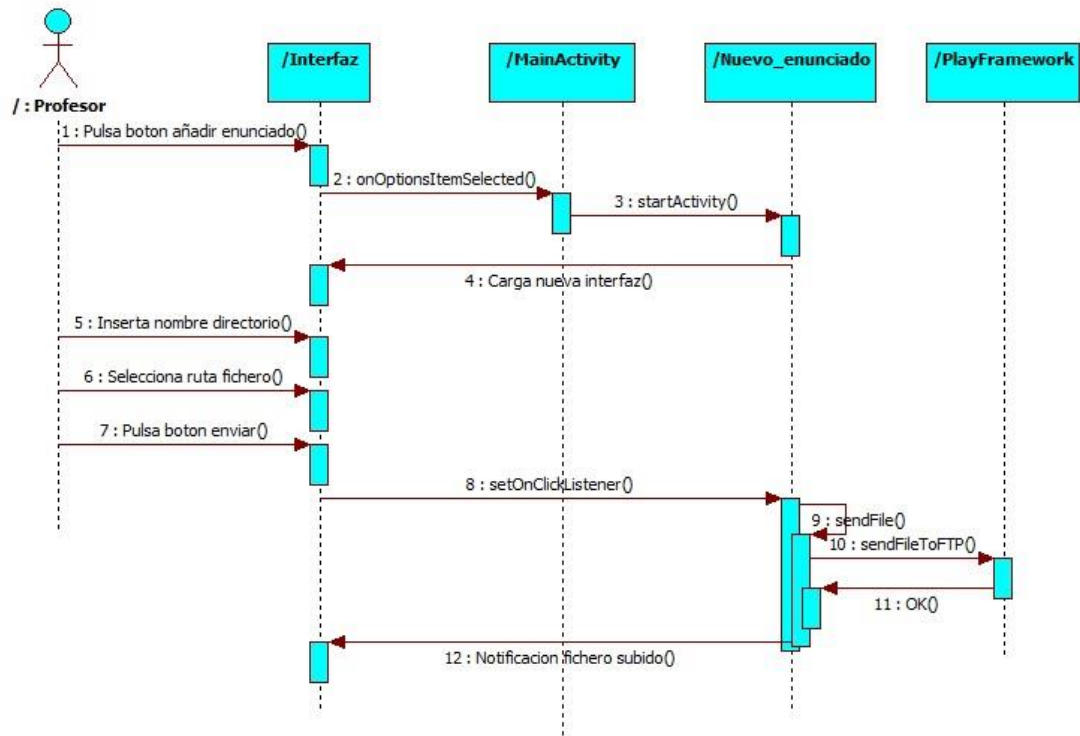


Ilustración 71: Diagrama de secuencia Añadir enunciado

1. El profesor pulsa la opción Subir enunciado.
2. En la clase MainActivity se ejecuta el método onOptionsItemSelected, que se inicia cada vez que se pulsa en una de las opciones principales.
3. Se ejecuta el método startActivity que inicia la actividad Nuevo_enunciado.
4. La aplicación carga la interfaz correspondiente.
5. El profesor inserta el nombre para el directorio raíz.
6. El profesor busca y selecciona un fichero del dispositivo.
7. El profesor pulsa el botón para enviar el archivo seleccionado.
8. Se ejecuta en la actividad el método setOnClickListener que realiza el código asociado a dicho botón.
9. Se inicia la clase asíncrona sendFile que realiza la petición al servicio web.
10. Se ejecuta el método sendFileToFTP en el servicio web para enviar el fichero al servidor FTP.
11. Al finalizar la subida envía un Ok a la aplicación para que continúe el proceso principal.
12. Tras recibir el Ok notifica al usuario de que se ha enviado el fichero de forma correcta.

4.5.4 Añadir cuenta LMS al servicio web

Este diagrama de secuencia muestra las acciones realizadas para añadir una nueva cuenta para acceder a los recursos de un LMS.

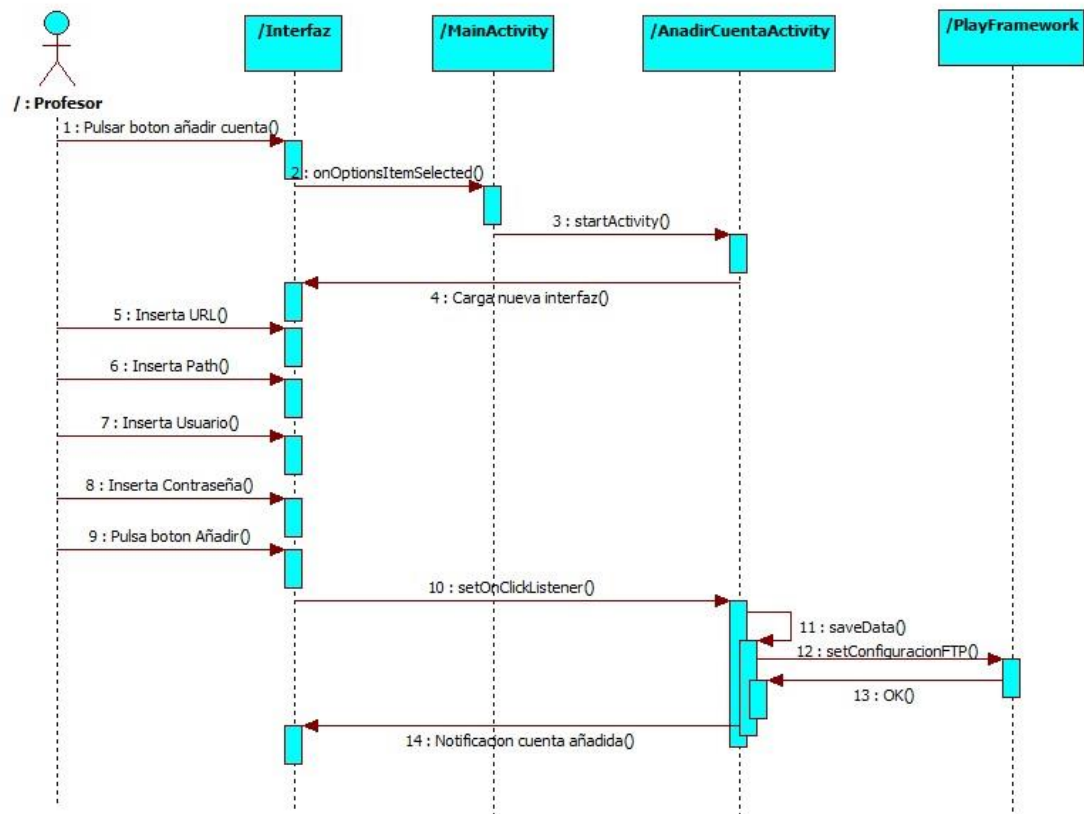


Ilustración 72: Diagrama de secuencia Añadir cuenta LMS

1. El profesor pulsa la opción Añadir nueva cuenta.
2. En la clase MainActivity se ejecuta el método onOptionsItemSelected, que se inicia cada vez que se pulsa en una de las opciones principales.
3. Se ejecuta el método startActivity que inicia la actividad AnadirCuentaActivity.
4. La aplicación carga la interfaz correspondiente.
5. El profesor inserta la URL del LMS.
6. El profesor inserta el Path del LMS.
7. El profesor inserta el usuario del LMS.
8. El profesor inserta la contraseña del LMS.

9. El profesor pulsa el botón de enviar para guardar la información introducida.
10. Se ejecuta en la actividad el método `setOnClickListener` que realiza el código asociado a dicho botón.
11. Se inicia la clase asíncrona `saveData` que realiza la petición al servicio web.
12. Se ejecuta el método `setConfigurationFTP` en el servicio web para almacenar los datos capturados.
13. Al finalizar el almacenamiento envía un `Ok` a la aplicación para que continúe el proceso principal.
14. Tras recibir el `Ok` notifica al usuario de que se ha guardado la nueva cuenta de forma correcta.

4.5.5 Mostrar y cambiar entre cuentas Twitter

A continuación se muestra las acciones realizadas para seleccionar una cuenta Twitter de entre las registradas en el dispositivo. Aunque el actor que se indica es el Profesor, el diagrama es válido también para el Alumno.

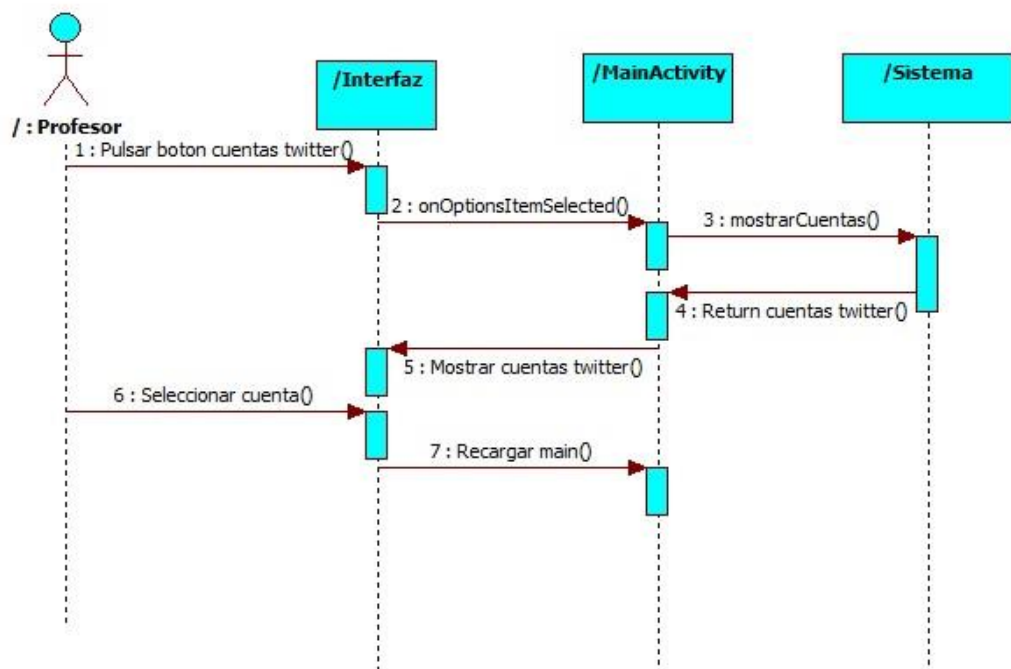


Ilustración 73: Diagrama de secuencia Cambiar cuenta Twitter

1. El profesor pulsa la opción Añadir nueva cuenta.

2. En la clase MainActivity se ejecuta el método `onOptionsItemSelected`, que se inicia cada vez que se pulsa en una de las opciones principales.
3. Se inicia el método `mostrarCuentas` que se encarga de recuperar los datos asociados a las cuentas Twitter del usuario.
4. Una vez recuperada la información se envía a la aplicación.
5. Se muestran las cuentas en una ventana emergente en la parte central de la pantalla.
6. El profesor selecciona una de las cuentas.
7. La aplicación se recarga actualizando los datos de la nueva cuenta.

4.5.6 Mostrar el enunciado de un ejercicio

A continuación se muestra las acciones realizadas para seleccionar un ejercicio y visualizar el contenido del enunciado. De nuevo, este diagrama es válido tanto para el Profesor y el Alumno.

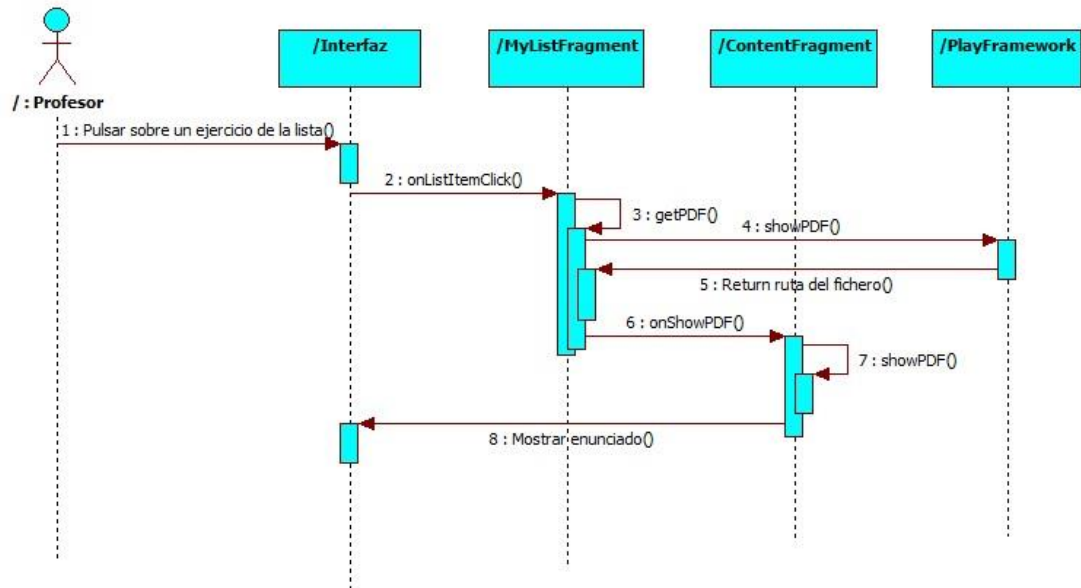


Ilustración 74: Diagrama de secuencia Mostrar enunciado

1. El profesor pulsa sobre un ejercicio de la lista.
2. En la clase `MyListFragment` se ejecuta el método `onListItemClick` que se inicia cada vez que se selecciona un ítem del `ListView`.

3. Se inicia el método `getPDF` para enviar la solicitud de recuperar un fichero del servidor.
4. En el servicio web se ejecuta el método `showPDF` para recuperar el contenido de dicho fichero.
5. Una vez que ha finalizado la tarea envía la ruta a la aplicación.
6. Se ejecuta el método `onShowPDF` de la clase `MyListFragment` para enviar la ruta recibida al fragment que mostrará el contenido.
7. Se inicia el método `showPDF` del `ContentFragment` para visualizar el contenido del enunciado
8. Se visualiza el contenido del fichero con la posibilidad de hacer zoom y navegar en él.

4.5.7 Mostrar la solución de un ejercicio

En el siguiente diagrama se muestra las acciones realizadas para seleccionar un ejercicio y visualizar el contenido del enunciado.

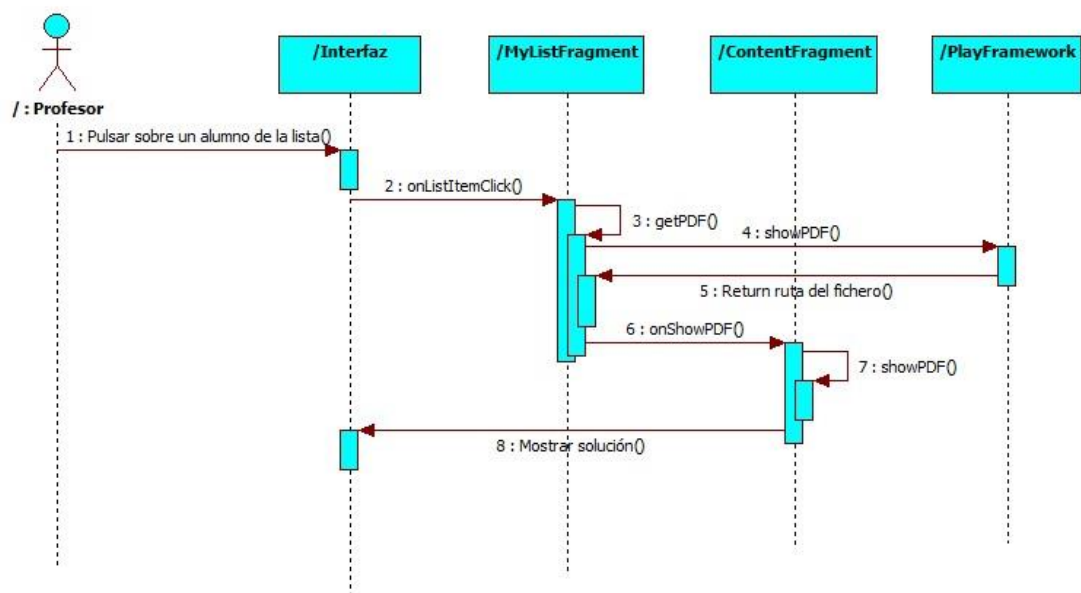


Ilustración 75: Diagrama de secuencia Mostrar solución

1. El profesor pulsa sobre un alumno de la lista.

2. En la clase MyListFragment se ejecuta el método onListItemClick que se inicia cada vez que se selecciona un ítem del ListView.
3. Se inicia el método getPDF para enviar la solicitud de recuperar un fichero del servidor.
4. En el servicio web se ejecuta el método showPDF para recuperar el contenido de dicho fichero.
5. Una vez que ha finalizado la tarea envía la ruta a la aplicación.
6. Se ejecuta el método onShowPDF de la clase MyListFragment para enviar la ruta recibida al fragment que mostrará el contenido.
7. Se inicia el método showPDF del ContentFragment para visualizar el contenido de la solución.
8. Se visualiza el contenido del fichero con la posibilidad de hacer zoom y navegar en él.

4.5.8 Enviar solución de un ejercicio

En el siguiente diagrama se muestra los pasos que permiten enviar una solución en la aplicación del alumno.

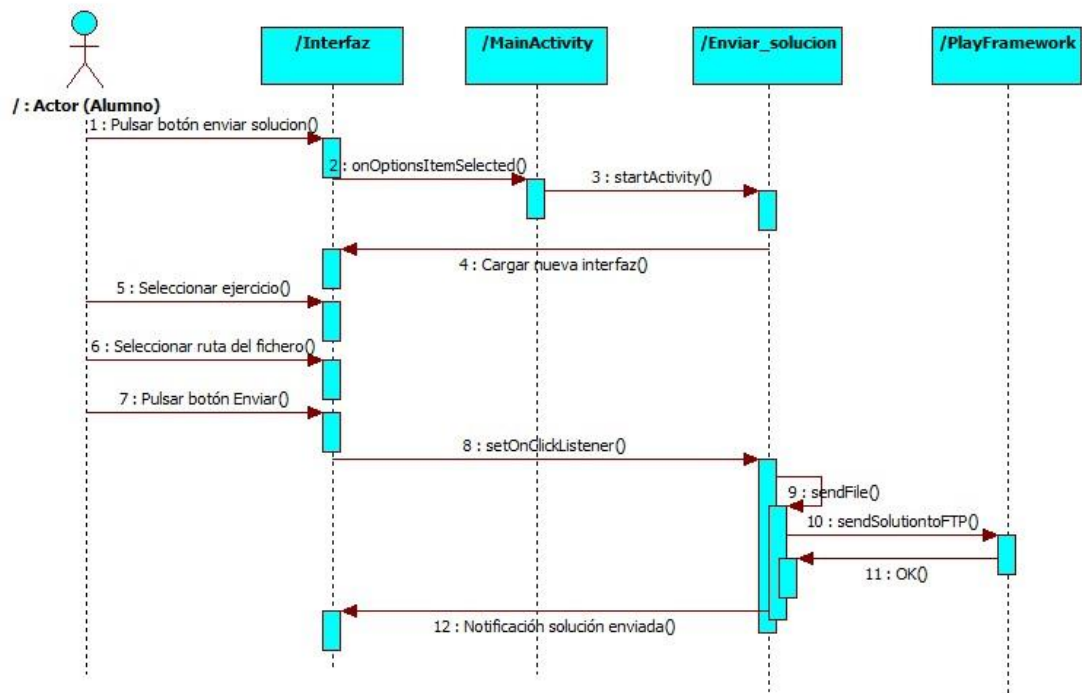


Ilustración 76: Diagrama de clase Enviar Solución

1. El alumno pulsa la opción Subir solución.
2. En la clase MainActivity se ejecuta el método onOptionsItemSelected, que se inicia cada vez que se pulsa en una de las opciones principales.
3. Se ejecuta el método startActivity que inicia la actividad Enviar_solucion.
4. La aplicación carga la interfaz correspondiente.
5. El alumno selecciona el ejercicio deseado.
6. El alumno localiza el fichero en el dispositivo.
7. El alumno pulsa el botón para enviar el archivo seleccionado.
8. Se ejecuta en la actividad el método setOnClickListener que realiza el código asociado a dicho botón.
9. Se inicia la clase asíncrona sendFile que realiza la petición al servicio web.
10. Se ejecuta el método sendSolutionToFTP en el servicio web para enviar el fichero al servidor FTP.
11. Al finalizar la subida envía un Ok a la aplicación para que continúe el proceso principal.
12. Tras recibir el Ok notifica al usuario de que se ha enviado el fichero de forma correcta.

4.5.9 Añadir asignatura de un profesor

En este diagrama se muestra las acciones realizadas por la aplicación del alumno para añadir y guardar diferentes asignaturas para acceder a sus recursos compartidos.

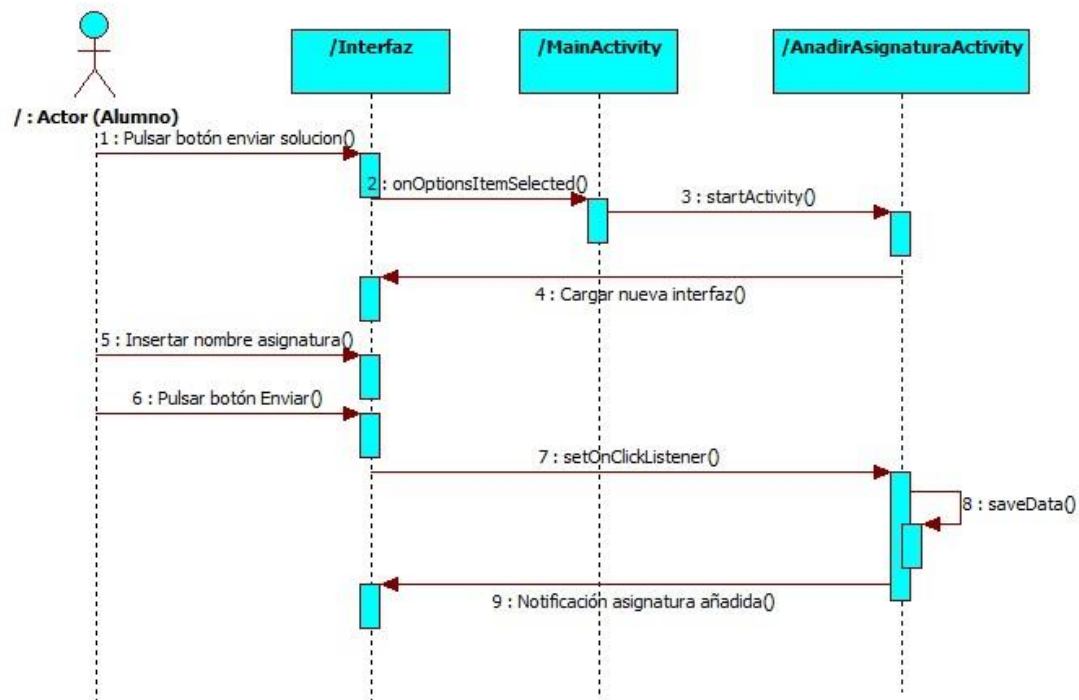


Ilustración 77: Diagrama de clase Añadir Asignatura

1. El alumno pulsa la opción Añadir asignatura.
2. En la clase MainActivity se ejecuta el método `onOptionsItemSelected()`, que se inicia cada vez que se pulsa en una de las opciones principales.
3. Se ejecuta el método `startActivity()` que inicia la actividad `AnadirAsignaturaActivity`.
4. La aplicación carga la interfaz correspondiente.
5. El alumno inserta el nombre de la asignatura a la que desea acceder.
6. El alumno pulsa el botón para añadir los datos introducidos.
7. Se ejecuta en la actividad el método `setOnClickListener()` que realiza el código asociado a dicho botón.
8. Se inicia el método `saveData()` que almacena en base de datos la nueva asignatura.
9. Tras guardar los datos se notifica al usuario de que se ha añadido la asignatura correctamente.

Capítulo 5

5. Pruebas

En este capítulo se definen los casos de pruebas unitarias y de sistema realizados para determinar el correcto funcionamiento de la aplicación.

5.1 Pruebas unitarias

Para verificar que la aplicación funciona de forma esperada se han diseñado diferentes pruebas unitarias para comprobar la funcionalidad de los métodos que se han implementado en las clases.

Para ello se ha creado un proyecto independiente compuesto de clases que al ejecutarlas como Android JUnit Test sirven de prueba para la aplicación del profesor y del alumno. Cada una de estas clases engloba los métodos comunes a algún aspecto de la aplicación.

En primer lugar, se describen las pruebas asociadas a la aplicación del profesor. La primera clase que se ha creado se denomina NetworkTest e incluye las siguientes pruebas:

- testInternetConexion: Verifica si el dispositivo está conectado a Internet.
- testWebServiceConexion: Comprueba que la aplicación se ha conectado correctamente con el servicio web que realiza las peticiones.

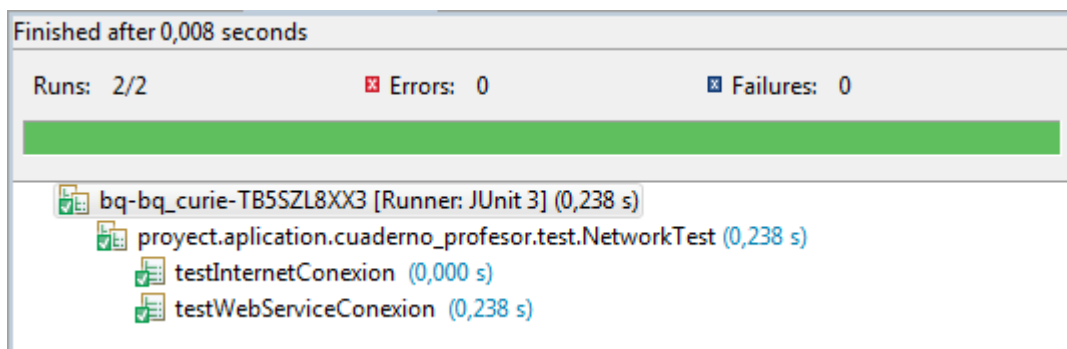


Ilustración 78: Pruebas unitarias NetworkTest (Profesor)

Se ha diseñado una segunda clase denominada ListFragmentTest que comprueba que se recupera la información requerida de las peticiones al servicio web para mostrarlas posteriormente al profesor. En ella se realizan las siguientes pruebas:

- testListaAlumnos: Es correcto si tras realizar la petición de mostrar la lista de alumnos al servicio web, este devuelve una lista que contenga la información requerida y no vacía.
- testListaEjercicios: Verifica que tras realizar la petición de mostrar la lista de ejercicios enviados, el servicio web devuelve una lista que contenga la información requerida y no vacía.
- testSolucionesEnviadas: Comprueba que el método que recupera las soluciones enviadas a un ejercicio devuelve una lista que no esté vacía.
- testEjerciciosResueltos: Es correcto si la lista que devuelve el servicio web tras recuperar los ejercicios resueltos por un alumno contiene la información deseada.
- testMostrarEnunciado: Es correcto si la ruta obtenida del fichero seleccionado coincide con la esperada.

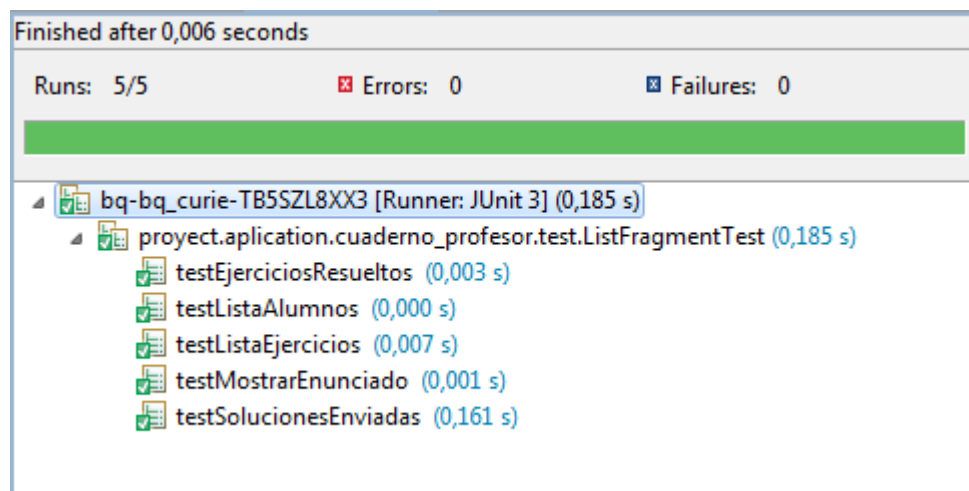


Ilustración 79: Pruebas unitarias ListFragmentTest (Profesor)

En segundo lugar, se describen las pruebas asociadas a la aplicación del alumno. Las clases que se han creado son análogas a las anteriormente descritas a excepción de algunos métodos que varían entre una aplicación y otra. La clase NetworkTest incluye las siguientes pruebas:

- testInternetConexion: Verifica si el dispositivo está conectado a Internet.
- testWebServiceConexion: Comprueba que la aplicación se ha conectado correctamente con el servicio web que realiza las peticiones.

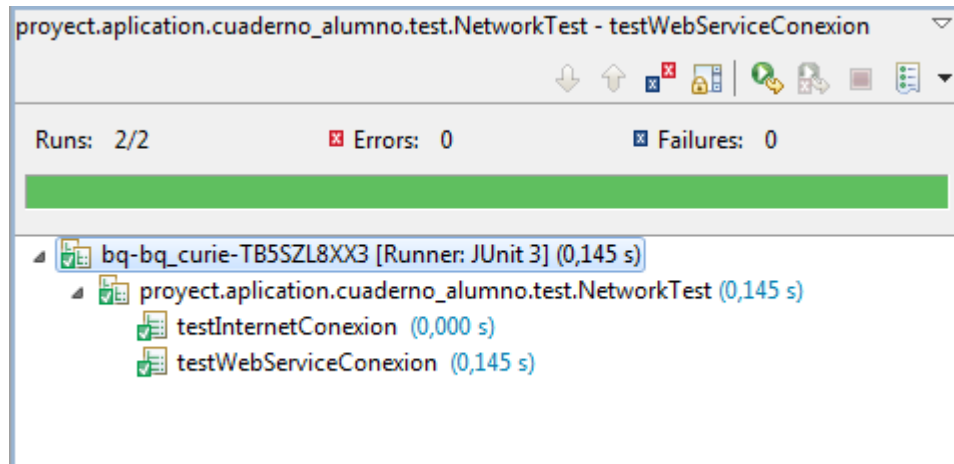


Ilustración 80: Pruebas unitarias NetworkTest (Alumno)

En la clase ListFragmentTest se prueba que la aplicación del alumno recibe de forma correcta los datos solicitados por el alumno desde el servicio web.

- testListaEjercicios: Verifica que tras realizar la petición de mostrar la lista de ejercicios enviados, el servicio web devuelve una lista que contenga la información requerida y no vacía.
- testEjerciciosResueltos: Es correcto si la lista que devuelve el servicio web tras recuperar los ejercicios resueltos por un alumno contiene la información deseada.
- testMostrarEnunciado: Es correcto si la ruta obtenida del fichero seleccionado coincide con la esperada.

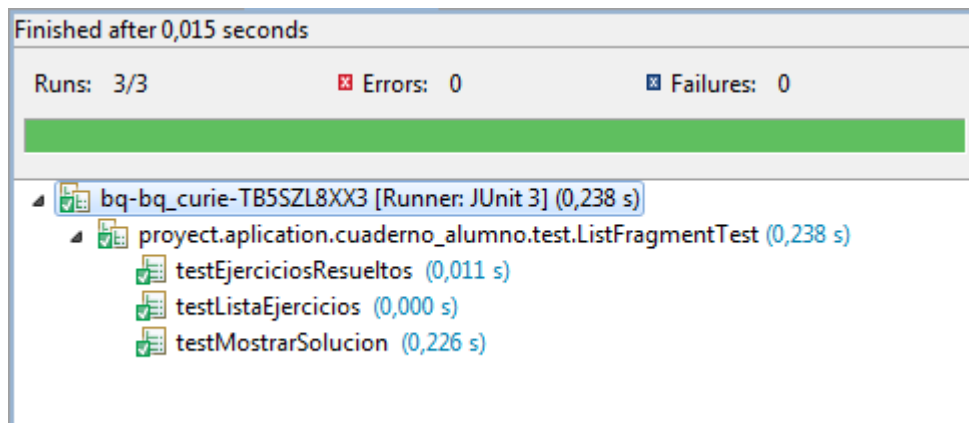


Ilustración 81: Pruebas unitarias ListFragmentTest (Alumno)

Por último se ha implementado una clase que contiene una batería de pruebas que verifican la comunicación entre las aplicaciones y el servidor FTP. Las pruebas que se han diseñado son las siguientes:

- testCambiarDirectorio: Comprueba que el sistema puede acceder al directorio indicado desde una ruta diferente.
- testConexion: Es correcto si la aplicación realiza correctamente la conexión con el servidor FTP.
- testCrearDirectorio: Verifica que se añade un nuevo directorio a la ruta establecida.
- testLeerFichero: Testea que el resultado esperado al leer un fichero es el mismo que se obtiene.
- testListarFicheros: Es correcto si la lista que recupera el nombre de los ficheros de un directorio no está vacía.
- testSubirEnunciado: Comprueba que al enviar la ruta de un fichero y definir el directorio de destino se añade dicho archivo al directorio.

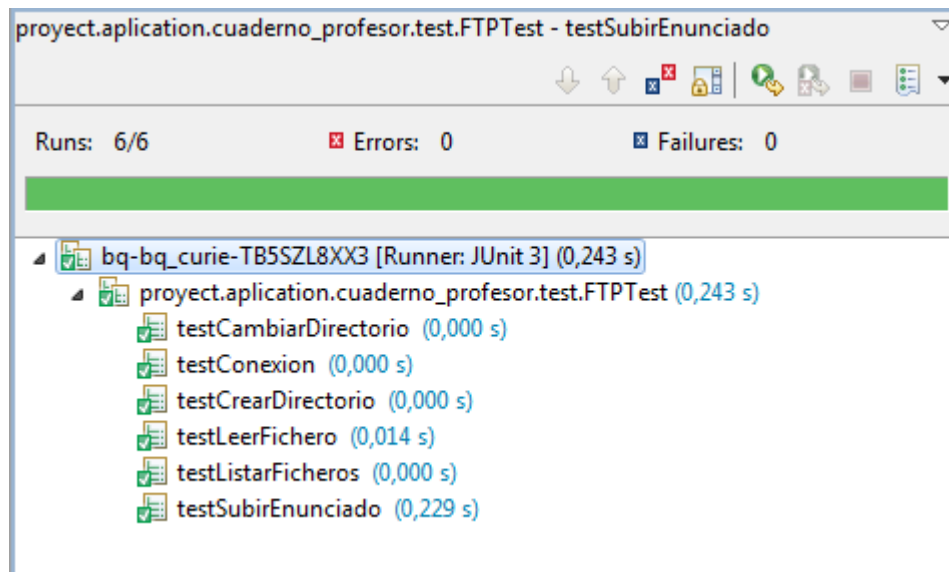


Ilustración 82: Pruebas unitarias FTPTest

5.2 Pruebas de sistema

Las pruebas de sistema permiten probar la aplicación conjuntamente con otros sistemas con los que tiene que interactuar para verificar que los requisitos se cumplen. A través de estas pruebas se garantiza que la navegación entre interfaces y el procesamiento y recuperación de datos se hagan de forma correcta.

Por ello, es esencial que las pruebas de sistema se enfoquen en los requisitos definidos en los casos de uso descritos anteriormente. Además se obtiene una visión muy similar al comportamiento que se espera en un entorno real.

Cada una de las pruebas va a venir definida por la siguiente tabla:

- ID: Número que identifica cada prueba.
- Descripción: Campo para definir en qué consiste la prueba.
- Pasos: Los sucesivos pasos que se han seguido para realizar la prueba.
- Errores posibles: Se describen los errores que pueden surgir al realizar los pasos del atributo anterior.
- Requisitos: Requisitos de la aplicación que se ven involucrados en la prueba.
- Resultado final: El resultado que se obtiene tras realizar cada prueba.

- Estado: Si el resultado de la prueba es el correcto, OK, o si por el contrario, ha ocurrido un problema, ERROR.

ID	
Descripción	
Pasos	
Errores posibles	
Requisitos	
Resultado final	
Estado	

Tabla 34: Plantilla pruebas de sistema

En primer lugar se detallan las pruebas de sistema de la aplicación del profesor. Esta primera prueba verifica que se visualizan de forma correcta el nombre y apellido de cada alumno.

ID	PS001
Descripción	Ver lista de alumnos
Pasos	1. Iniciar la aplicación
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído
Requisitos	CP003, CP004
Resultado final	Aparece un listado de los alumnos de la asignatura
Estado	OK

Tabla 35: PS001

Esta prueba es correcta si se muestra adecuadamente un listado de los ejercicios que ha subido el profesor al LMS.

ID	PS002
Descripción	Ver lista de ejercicios
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar en la pestaña de Ejercicios
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web

	<ul style="list-style-type: none"> • El servidor FTP está caído
Requisitos	CP003, CP005
Resultado final	Se muestra un listado de los ejercicios disponibles
Estado	OK

Tabla 36: PS002

Esta tercera prueba verifica que el profesor puede enviar un nuevo ejercicio al servidor.

ID	PS003
Descripción	Subir un ejercicio
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre el botón el primer botón del menú superior 3. Introducir el nombre de la entrada del ejercicio 4. Pulsar sobre el botón Buscar para localizar el fichero 5. Seleccionar el fichero deseado del dispositivo 6. Pulsar sobre el botón Enviar
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído • El fichero está dañado
Requisitos	CP003, CP008, CP011
Resultado final	El fichero queda almacenado en el repositorio para consultar su contenido posteriormente.
Estado	OK

Tabla 37: PS003

En la siguiente tabla se muestra la prueba que se ha realizado para comprobar la funcionalidad de añadir diferentes cursos en la aplicación.

ID	PS004
Descripción	Añadir un curso
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre el segundo botón del menú superior 3. Pulsar en el botón de Añadir repositorio 4. Seleccionar la identidad de Twitter deseada

	<ol style="list-style-type: none"> Introducir la URL del servidor Teclear el Path necesario para acceder a los recursos Insertar Usuario Insertar Contraseña Pulsar el botón de Añadir
Errores posibles	<ul style="list-style-type: none"> El dispositivo no tiene conexión a Internet La aplicación no se comunica con el servicio web Formato de los datos introducidos incorrecto
Requisitos	CP003, CP013
Resultado final	Los datos del repositorio quedan guardados en la configuración del servicio web
Estado	OK

Tabla 38: PS004

De manera análoga, se crea una prueba para verificar que los datos de configuración de un repositorio se pueden modificar.

ID	PS005
Descripción	Modificar datos de un curso
Pasos	<ol style="list-style-type: none"> Iniciar la aplicación Pulsar sobre el segundo botón del menú superior Pulsar en el botón de Editar configuración Seleccionar la identidad de Twitter deseada Modificar la URL del servidor Teclear el Path necesario para acceder a los recursos Modificar Usuario Modificar Contraseña Pulsar el botón de Añadir
Errores posibles	<ul style="list-style-type: none"> El dispositivo no tiene conexión a Internet La aplicación no se comunica con el servicio web Formato de los datos introducidos incorrecto
Requisitos	CP003, CP013
Resultado final	Los datos del repositorio quedan guardados en la configuración del servicio web

Estado	OK
---------------	----

Tabla 39: PS005

A continuación se testea que el profesor puede cambiar entre las diferentes cuentas Twitter registradas en el dispositivo.

ID	PS006
Descripción	Cambiar cuenta Twitter
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre el tercer botón del menú superior 3. Seleccionar un ID de Twitter
Errores posibles	<ul style="list-style-type: none"> • La aplicación no puede recuperar los datos de Twitter
Requisitos	CP001, CP002
Resultado final	La aplicación se reinicia y carga la configuración asociada a dicha identidad, mostrándola en la barra superior
Estado	OK

Tabla 40: PS006

En la siguiente tabla se comprueba que el profesor puede visualizar qué ejercicios han sido resueltos por un alumno en concreto.

ID	PS007
Descripción	Mostrar un listado de los ejercicios resueltos por un alumno
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre un alumno
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído
Requisitos	CP003, CP006
Resultado final	Aparece un listado de los ejercicios que ha resuelto un alumno anteriormente seleccionado
Estado	OK

Tabla 41: PS007

En la siguiente prueba se testea que la aplicación muestra un listado de los alumnos que han enviado una solución a un ejercicio en particular.

ID	PS008
Descripción	Mostrar listado de los alumnos que han resuelto un ejercicio
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar en la pestaña de Ejercicios 3. Pulsar sobre un ejercicio
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído
Requisitos	CP003, CP007
Resultado final	Aparece un listado de los alumnos que ha resuelto un ejercicio anteriormente seleccionado
Estado	OK

Tabla 42: PS008

En la octava prueba se verifica que se puede visualizar el contenido de un ejercicio enviado por el profesor.

ID	PS009
Descripción	Ver enunciado de un ejercicio
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar en la pestaña de Ejercicios 3. Seleccionar un ejercicio
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído • El contenido del fichero no puede mostrarse
Requisitos	CP003, CP009
Resultado final	Aparece el contenido del enunciado de dicho ejercicio
Estado	OK

Tabla 43: PS009

En la última tabla, se comprueba la funcionalidad de ver la solución de un ejercicio enviada por un alumno.

ID	PS010
Descripción	Ver solución de un ejercicio
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre un alumno 3. Seleccionar un ejercicio
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído • El contenido del fichero no puede mostrarse
Requisitos	CP003, CP010
Resultado final	Aparece el contenido de la solución de dicho ejercicio
Estado	OK

Tabla 44: PS010

De manera análoga a las descritas anteriormente, se describen a continuación, las pruebas de sistema que se han realizado en la aplicación del alumno:

En esta prueba se verifica que el alumno puede visualizar todos los ejercicios que el profesor ha compartido.

ID	PS011
Descripción	Ver lista de ejercicios disponibles
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar en la pestaña de Ejercicios
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído
Requisitos	CA003, CA004
Resultado final	Se muestra un listado de los ejercicios compartidos por el profesor
Estado	OK

Tabla 45: PS011

Se muestra la comprobación de que el alumno puede añadir diferentes asignaturas en su aplicación.

ID	PS012
Descripción	Añadir una asignatura
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre el segundo botón del menú superior 3. Pulsar en el botón de Añadir asignatura 4. Introducir la identidad social de la nueva asignatura 5. Pulsar el botón de Añadir
Errores posibles	<ul style="list-style-type: none"> • Formato de los datos introducidos incorrecto
Requisitos	CA003, CA009
Resultado final	Se añade la nueva asignatura a la aplicación para seleccionarla posteriormente
Estado	OK

Tabla 46: PS012

Comprueba que el alumno tiene la opción de cambiar entre sus cuentas de Twitter.

ID	PS013
Descripción	Cambiar cuenta Twitter
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre el tercer botón del menú superior 3. Seleccionar un ID de Twitter
Errores posibles	<ul style="list-style-type: none"> • La aplicación no puede recuperar los datos de Twitter
Requisitos	CA001, CA002
Resultado final	La aplicación se reinicia y carga la configuración asociada a dicha identidad, mostrándola en la barra superior
Estado	OK

Tabla 47: PS013

Verifica que el alumno puede subir una solución a un ejercicio determinado.

ID	PS014
Descripción	Subir solución
Pasos	<ol style="list-style-type: none"> 1. Iniciar aplicación 2. Pulsar sobre el primer botón del menú superior 3. Seleccionar el ejercicio deseado 4. Pulsar sobre el botón de Buscar

	5. Localizar el fichero en el dispositivo 6. Pulsar sobre el botón Enviar
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído El fichero está dañado
Requisitos	CA003, CA005, CA006
Resultado final	El fichero queda almacenado en el repositorio
Estado	OK

Tabla 48: PS014

Comprueba que el alumno visualiza el contenido del enunciado de un ejercicio.

ID	PS015
Descripción	Ver enunciado de un ejercicio
Pasos	1. Iniciar la aplicación 2. Seleccionar un ejercicio
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído • El contenido del fichero no puede mostrarse
Requisitos	CA003, CA007
Resultado final	El contenido del enunciado se muestra en la aplicación
Estado	OK

Tabla 49: PS015

Verifica que el alumno puede visualizar el contenido de una solución que haya subido con anterioridad.

ID	PS016
Descripción	Ver solución de un ejercicio
Pasos	<ol style="list-style-type: none"> 1. Iniciar la aplicación 2. Pulsar sobre un ejercicio resuelto 3. Pulsar sobre el botón de Mostrar solución
Errores posibles	<ul style="list-style-type: none"> • El dispositivo no tiene conexión a Internet • La aplicación no se comunica con el servicio web • El servidor FTP está caído • El contenido del fichero no puede mostrarse
Requisitos	CA003, CA008
Resultado final	El contenido de la solución se muestra en la aplicación
Estado	OK

Tabla 50: PS016

Capítulo 6

6. Planificación y presupuesto

En este apartado se describe la planificación que se ha seguido durante la realización del proyecto detallando los requisitos que se han implementado en cada una de las iteraciones. Por otro lado, se hará una descripción de los costes que han supuesto la realización de todo el trabajo de fin de grado.

6.1 Planificación

Como ya se ha indicado en apartados anteriores, el desarrollo del proyecto se ha enfocado en las metodologías ágiles, por lo que la planificación que se ha seguido ha sido adaptativa e iterativa. Es decir, a diferencia de las metodologías tradicionales, solo se planifica en detalle la siguiente iteración de manera que, al comienzo de una nueva iteración, se determinan los requisitos a abordar en dicha iteración y se definen las tareas necesarias para el desarrollo de los mismos. En concreto, en este proyecto se han realizado 7 iteraciones.

A continuación se muestra de forma detallada cuáles han sido las tareas que se definieron en cada iteración, y los requisitos que se cumplían al finalizar la iteración:

6.1.1 Primera iteración

Se llevó a cabo un análisis del problema donde se definieron los objetivos principales del proyecto y se realizaron los primeros prototipos del diseño. Además se llevó a cabo un primer estudio de los sistemas PLE y se implementaron las primeras interfaces y actividades de la aplicación basadas en dicho prototipo.

Tarea	Fecha inicio	Fecha fin
Análisis y estudio PLE	04/02/2013	05/02/2013
Análisis del problema	06/02/2013	06/02/2013
Diseño prototipos	07/02/2013	08/02/2013
Implementación interfaces profesor	09/02/2013	13/02/2013
Implementación interfaces alumno	15/02/2013	19/02/2013

Tabla 51: Tareas de la primera iteración

6.1.2 Segunda iteración

En esta fase, se definió el nuevo diseño de la interfaz donde la pantalla se divide en dos fragmentos separados, y se establecieron nuevos requisitos. Por ello, en esta iteración, la principal tarea fue modificar el código de las interfaces y actividades de la aplicación para adaptarlo al nuevo diseño. Además, se hizo el primer análisis de las diferentes aplicaciones existentes en el mercado.

Tarea	Fecha inicio	Fecha fin
Análisis de mercado	20/02/2013	21/02/2013
Implementar fragment izquierdo del profesor	04/03/2013	05/03/2013
Implementar fragment derecho del profesor	06/03/2013	07/03/2013
Añadir las pestañas al fragment izquierdo	08/03/2013	13/03/2013
Implementar fragment izquierdo del alumno	18/03/2013	19/03/2013
Implementar fragment derecho del alumno	20/03/2013	21/03/2013

Tabla 52: Tareas de la segunda iteración

6.1.3 Tercera iteración

Una vez que se acordó el diseño y las interfaces definitivas se definieron tareas relacionadas con la implementación del LMS para recuperar y mostrar la información asociada a los alumnos y los ejercicios, creando una clase en Java que contuviera los métodos asociados al intercambio de datos con el servidor FTP.

Tarea	Fecha inicio	Fecha fin	Requisitos
Leer fichero listado de alumnos del FTP	02/04/2013	03/04/2013	CP004
Mostrar el listado de alumnos	03/04/2013	03/04/2013	CP004
Crear método listar directorio del FTP	09/04/2013	09/04/2013	CP005
Mostrar la lista de todos los ejercicios	09/04/2013	09/04/2013	CP005, CA004
Crear método para listar soluciones de los alumnos de un ejercicio	10/04/2013	12/04/2013	CP007
Notificar qué alumnos han resuelto un ejercicio	15/04/2013	15/04/2013	CP007

Crear método para listar los ejercicios que ha resuelto un alumno	23/04/2013	24/04/2013	CP006
--	------------	------------	-------

Tabla 53: Tareas de la tercera iteración

6.1.4 Cuarta iteración

En la iteración siguiente se definieron nuevos requisitos y se amplió la funcionalidad de la aplicación implementando el código correspondiente a compartir un fichero y mostrar su contenido.

Tarea	Fecha inicio	Fecha fin	Requisitos
Crear método para enviar un fichero al FTP	10/06/2013	10/06/2013	CP008, CP011, CA005, CA006
Crear método para buscar y recuperar un fichero del FTP	11/06/2013	11/06/2013	CP009, CP010, CA007, CA008
Mostrar el contenido del fichero	12/06/2013	12/06/2013	CP009, CP010, CA007, CA008

Tabla 54: Tareas de la cuarta iteración

6.1.5 Quinta iteración

Tras analizar varias alternativas de diseño, se decidió por implementar la plataforma relativa al PLE en un servicio web, por lo que, de nuevo, se realizaron las modificaciones oportunas en el diseño y en el código, definiendo con anterioridad las tareas asociadas a este requisito.

Tarea	Fecha inicio	Fecha fin	Requisitos
Crear servicio web utilizando Play Framework	24/06/2013	27/06/2013	CP003
Mover los métodos del FTP de la aplicación al servicio web	03/07/2013	05/07/2013	-
Establecer la comunicación entre el dispositivo y el servicio web	08/07/2013	09/07/2013	CP003
Respuestas del servicio web en formato JSON	15/07/2013	19/07/2013	-

Adaptar el código de la aplicación para capturar las respuestas y mostrar los diferentes resultados a través de la interfaz	24/07/2013	30/07/2013	-
--	------------	------------	---

Tabla 55: Tareas de la quinta iteración

6.1.6 Sexta iteración

En la siguiente fase, y tras realizar pruebas unitarias verificando el correcto funcionamiento del servicio web y de la comunicación entre éste y el servidor FTP, realizando las funciones de LMS, se establecieron las tareas relativas a la autenticación e identificación de los usuarios mediante la identidad social de Twitter.

Tarea	Fecha inicio	Fecha fin	Requisitos
Implementación alternativa librería externa	06/08/2013	12/08/2013	CP001, CA001
Modificación y adaptación del código para el diseño final	13/08/2013	14/08/2013	-
Acceder y recuperar las cuentas del usuario registradas en el dispositivo	15/08/2013	15/08/2013	CP001, CA001
Mostrar en un popup las cuentas	15/08/2013	15/08/2013	CP002, CA002
Implementar método para añadir cuentas al servicio web	16/08/2013	17/08/2013	CP013
Implementar método para añadir y acceder a los recursos de una asignatura	20/08/2013	21/08/2013	CA003, CA009

Tabla 56: Tareas de la sexta iteración

6.1.7 Séptima iteración

Por último, en la última iteración se realizaron el resto de pruebas unitarias y las pruebas de sistema garantizando el correcto funcionamiento del sistema en conjunto.

Tarea	Fecha inicio	Fecha fin
Pruebas unitarias del profesor	20/08/2013	23/08/2013
Pruebas unitarias del alumno	26/08/2013	30/08/2013
Pruebas de sistema del profesor	02/09/2013	04/09/2013
Pruebas de sistema del alumno	05/09/2013	06/09/2013

Tabla 57: Tareas de la séptima iteración

En cada una de las iteraciones descritas, se ha procedido de la siguiente manera: en primer lugar, realizar un análisis de las posibles alternativas, después, diseñar e implementar el código asociado, y, por último, llevar a cabo las pruebas de aceptación necesarias para conseguir el software funcional y estable requerido en cada iteración. Mencionar, por otro lado, que la realización de la documentación se ha llevado a cabo de forma paralela al desarrollo de las dos aplicaciones. En la siguiente tabla se muestra el tiempo en horas que se ha empleado para la realización de este proyecto.

Fecha comienzo del proyecto	14 de enero 2013
Fecha comienzo del proyecto	13 de septiembre 2013
Días empleados	120 días laborables
Dedicación al día	3 horas
Tiempo total dedicado	360 horas

Tabla 58: Tiempo dedicado al proyecto

6.2 Presupuesto

En este apartado se definen los gastos asociados al desarrollo de este proyecto. En primer lugar se definen los costes derivados del personal. Se han incluido al desarrollador principal, Álvaro Mateos, y a los dos tutores que han computado horas por las sucesivas reuniones de cada iteración y por la ayuda que han ofrecido en el desarrollo del proyecto.

Personal	Cargo	Salario	Horas	Coste (€)
Álvaro Mateos Jiménez	Ingeniero Junior	15 €/h	360	5400
Alberto Heredia García	Ingeniero Senior	30 €/h	30	900
Javier García Guzmán	Ingeniero Senior	30 €/h	10	300

Tabla 59: Costes de personal

Por otro lado, se debe tener en cuenta el coste asociado a las amortizaciones de los dispositivos que se han utilizado durante el desarrollo del proyecto como son un ordenador portátil donde se ha llevado a cabo toda la implementación y una Tablet Android para realizar todas las pruebas.

Dispositivo	Precio Unidad	Tiempo de vida útil	Tiempo de uso	Coste en el proyecto
Ordenador portátil Toshiba, Procesador Intel Core i5 2.30 Ghz Memoria 4 GB, Windows 7	700 €	48 meses	7 meses	102 €
Tablet BQ Curie	170 €	24 meses	7 meses	50 €

Tabla 60: Costes amortizaciones

Finalmente el coste total que ha supuesto la realización de este proyecto es la suma de los costes de personal y de las amortizaciones dando un total de 6.752 €:

PRESUPUESTO	COSTE
Personal	6.600
Amortizaciones	152 €
COSTE TOTAL	6.752€

Tabla 61: Coste total del proyecto

Capítulo 7

7. Conclusiones y líneas futuras

7.1 Conclusiones

La realización de este proyecto ha permitido el desarrollo de una plataforma que utiliza la red social de Twitter para facilitar a los usuarios el aprendizaje online, accediendo a recursos de diferentes plataformas, que sirven como complemento a los contenidos dados por los profesores en las clases presenciales.

Los dos objetivos que se definieron al inicio del proyecto se han cumplido. El primero de ellos, crear una plataforma que permita al profesor compartir ejercicios de clase se ha llevado a cabo realizando una aplicación que realiza la conexión a un PLE que permite enviar ficheros al servidor al LMS. De forma análoga, el segundo objetivo se ha cumplido implementando una aplicación para el alumno con la que poder acceder a los ejercicios compartidos por el profesor a través del PLE y enviar las soluciones al LMS.

Para implementar el PLE, se ha creado un servicio web desde donde se gestionará a qué repositorio tiene que acceder la aplicación para enviar y compartir un recurso. Esto se lleva a cabo debido a que el usuario, al iniciar la aplicación, tiene que acceder haciendo uso de su identidad de Twitter. Esta identificación va a permitir que el servicio web acceda a una plataforma online u otra. Respecto al LMS que permite el almacenamiento y gestión de los recursos, se ha optado por un servidor FTP, aunque esté abierta la posibilidad, para futuros desarrollos, de añadir otro servidor diferente.

El desarrollo de este proyecto me ha permitido llevar a la práctica los conocimientos que he adquirido a lo largo de los cuatro años del grado de ingeniería informática, en especial en asignaturas como Ingeniería del Software, Dirección de Proyectos de Desarrollo Software y Seguridad en Dispositivos Móviles. Por otro lado, me ha servido para ampliar dichos conocimientos en aspectos que, o no se han dado durante la carrera o simplemente se han impartido de forma general, como ha sido el desarrollo de un servicio web que alojara la comunicación con el servidor FTP,

conocer las funcionalidades principales que aporta el API de Twitter y las diferentes clases y métodos desarrollados en Java que permiten la comunicación mediante el protocolo FTP.

Mencionar por otro lado, que a pesar de que en este año académico he llevado a cabo el desarrollo de una práctica en Android, este proyecto me ha ofrecido la oportunidad de implementar una aplicación, con todas sus funcionalidades, partiendo desde cero. Para ello se ha tenido que pasar por todas las etapas que componen el desarrollo de un proyecto software como son el análisis, diseño, implementación y pruebas. En este aspecto, seguir una metodología ágil de trabajo me ha permitido ir desarrollando la aplicación de manera progresiva adaptando el diseño a los requisitos del cliente definidos en cada reunión. Esto ha facilitado la comunicación entre ambas partes, de forma que las modificaciones que había que realizar, en la mayoría de las ocasiones, no conllevaban mucho tiempo llevarlas a cabo.

Realizar esta aplicación me ha ayudado a conocer la situación actual del mercado en relación a los sistemas operativos móviles, llegando a la conclusión de que haber elegido Android, me ha aportado muchas cosas útiles que seguramente pueda aprovechar al salir la universidad ya que, hoy en día, es un sistema que está en auge. Algunos de estos aspectos a destacar, son el hecho de haber profundizado en la arquitectura de Android y conocer los componentes que forman este sistema, así como en lo relativo a elementos de la programación, como son el uso de fragments, para dividir la pantalla con contenido diferente, tabhost para añadir pestañas, o el cómo modificar los colores por defecto que componen un layout.

Personalmente, creo que uno de los aspectos más importantes de este proyecto es el hecho de, a diferencia de estos cuatro años en los que siempre he tenido un compañero o un grupo de trabajo, haber realizado este trabajo solo. Ha sido un reto compaginar el proyecto con las clases, las prácticas, los exámenes y el trabajo. Y pienso que llevar a cabo el desarrollo de un proyecto como este, de inicio a fin, es la mejor forma de mejorar los conocimientos adquiridos durante estos cuatro años y poner punto y final a un grado de informática muy completo.

7.2 Líneas futuras

En este apartado señalo aquellos aspectos que pueden mejorarse en un futuro con el objetivo de conseguir una aplicación competente con las aplicaciones que actualmente se ofertan, de forma gratuita o de pago, en los mercados más importantes.

Uno de los objetivos que se han perseguido en la realización de este proyecto es el de unificar en una misma aplicación el acceso a diferentes plataformas de aprendizaje online. Sin embargo, sólo se ha implementado el acceso a un servidor FTP. Sería interesante que se añadieran otras APIs correspondientes a los LMS más importantes, como son Moodle o Dropbox. De esta forma, se dotaría a la aplicación de una mayor flexibilidad y facilitaría la compartición de recursos de diferentes fuentes entre el profesor y el alumno.

Respecto a las funcionalidades, cabe mencionar que esta aplicación está limitada a compartir ficheros y ver el contenido de los mismos. Sería conveniente añadir nuevas funcionalidades que permitieran al profesor realizar tareas como evaluar el contenido de una solución, mostrar las notas obtenidas por un alumno, o tener la opción de calcular la media que han conseguido los alumnos en un ejercicio determinado.

Aprovechando el uso que se hace de Twitter, se podría dar la opción al profesor de enviar un tweet de forma automática notificando que está disponible un nuevo ejercicio. Esto permitiría a los alumnos no tener que abrir la aplicación cada vez que quieran ver si hay un nuevo ejercicio para resolver. Además, sería muy útil el poder intercambiar mensajes privados entre el profesor y el alumno para enviar dudas o comentarios.

Por último, otro aspecto importante del proyecto a mejorar, sería el de ofrecer al usuario la posibilidad de utilizar no sólo la identidad social de Twitter, sino la de otras redes sociales como Facebook o Google+, de forma que el usuario pueda elegir el perfil que más se adapte a sus necesidades.

Conclusions and futures researches

Conclusions

The carrying out of this project has allowed the development of a platform that uses the social network Twitter in order to facilitate the online learning for users. Having access to different platform's resources which can be used as a complement for the contents given by teachers in face to face instruction.

Both objectives described at the beginning of this project have been accomplished. The first of them, creating a platform that allows the teacher sharing exercises, has been fulfilled by developing an application which makes a connection to a PLE that, meanwhile, allows sending files to the LMS server. Likewise, the second objective has been satisfied by implementing an application for the student that allows the access to the exercises shares by the teacher through the PLE and send the solutions to the LMS.

To implement the PLE, it has been shaped a web service. From that web service, it will be managed which repository the application will have to access to, to send and share a resource.

That is carried out when the user, by initiating the application, has to use his or her Twitter account to access. The identification allows the access of the web service to different online platforms. In relation to the LMS, that consents the storage and management of the resources, an FTP server has been chosen. Even though there is still the chance to future researches, to add a different server.

The development of this project has given me the chance to practice the different knowledge that I have learnt in four years in the Bachelor's Degree in Computer Science and Engineering. Particularly the knowledge acquired in subjects such as Software Engineering, Software development Projects Management and Mobile Devices Security. On the other hand, it has amplified this knowledge in some ways that I could not do during the degree or knowledge of which we only had an overview. E.g. developing a web service which hosted the communication with the FTP server, the principal functions that the API provides from Twitter and the diverse kinds and methods made in Java that allow communication through the FTP protocol.

It bears mentioning that, although during the last academic year I have carried out a practice in Android, this project has provided me with the opportunity of implementing an application, with all its functions, starting from zero. To reach that goal, it has been necessary through all the stages that make up the development of a software project such as the analysis, the design, implementation and proofs. In this regard, following an agile methodology of work, has given me the opportunity to develop the application in a progressive way, maturing the design with the client's requirements, previously defined in each meeting. That has made easier the communication between both parts, in that way, the modification that had to be done, did not take a long time to accomplish.

To realize this application, has allowed me to know the market's real situation in connection with the mobile operating systems. All of this leads me to the conclusion that having chosen Android, has offered me many useful things that I will certainly use when I will finish my studies at college. That is because, nowadays, it's a rising system. Some of the aspects I would like to highlight are, in the first place, having the chance to study deeply the Android's architecture. Secondly, knowing which components form a system. Finally, but not least, the elements for programming, like: using fragments to divide the screen with a different content, tabhost to add tabs, or how to modify predetermined colours in a layout.

Personally, I believe that one of the most relevant aspects of this project is the fact that, in contrast with the past four years, where I always had a partner or a group of work, I did this on my own. It has been a continuous challenge to combine the time I spent carrying out the project with lessons, practices, exams and my job. Furthermore, I think that doing a project such as this one, from the beginning to the end, is the best way to improve the previously acquired knowledge during the past four years. And finally, to draw a line under a very complete Bachelor's Degree in Computer Science and Engineering.

Future researches

In this section I am going to point the aspects that could be improved in the future, in order to get a competent application that could actually compete with the applications that are offered, free of charge or paying an specific amount, in the most important markets.

One of the objectives that have been followed in the realization of this project is to unify in one application the access to different platforms to online learning. However, it has only been implemented the access to a FTP server. It would be really interesting to add some other APIs corresponding to the most important LMS, like Moodle or Dropbox. In that way, the application would be provided with more flexibility and would make easier to share resources from different sources between the teacher and the student.

In relation to the functionalities, it is important to underline that this application will be limited to share files and see its contents. It would be convenient to add some new functions which allowed the teacher to make some tasks like: evaluate the content of a solution, show each student's marks or having the option to calculate the student's average score in a specific exercise.

Taking advantage of the Twitter platform, the teacher could have the choice to send a tweet automatically to notify that a new exercise is available. In that way, students would not have to open the application each time they wanted to check if there is a new exercise to solve. Furthermore, it would be very useful to have the possibility to exchange private messages between the teacher and the student to solve doubts or comments.

Finally, another important part of the project to improve, would be to offer the user the chance to use, not only the Twitter identity, but the one from other social networks like Facebook or Google+, in order that the user could chose the most convenient profile according to his needs.

Anexo

Bibliografía

[1] Graham Attwell. Personal learnign environments. Consultado en abril de 2013.

<http://www.slideshare.net/GrahamAttwell/personal-learning-environments-46423>

[2] Graham Attwell. Personal learnign environments. The future of eLearning.

Consultado en abril de 2013.

<http://elearningeuropa.info/files/media/media11561.pdf>

[3] Mark van Harmelen. Personal learnign environments. Consultado en abril de 2013.

http://wiki.ties.k12.mn.us/file/view/PLEs_draft.pdf/282847312/PLEs_draft.pdf

[4] Universidad de Salamanca. Mobile Personal Learning Environments.

Consultado en abril de 2013. <http://grial.usal.es/node/239>

[5] José H. Canós, Patricio Letelier y M^a Carmen Penadés. Metodologías Ágiles en el Desarrollo de Software. Consultado en mayo de 2013.

http://noqualityinside.com.ar/nqi/nqifiles/XP_Agil.pdf

[6] IBM. Desarrollo ágil. Consultado en mayo de 2013.

<http://www-01.ibm.com/software/rational/agile/>

[7] ¿Qué es SCRUM? Consultado en mayo de 2013.

<http://www.proyectosagiles.org/que-es-scrum>

[8] Introducción a Android. Consultado en mayo de 2013.

<http://www.android.com/about/>

[9] Arquitectura de Android. Consultado en junio de 2013.

http://www.tutorialspoint.com/android/android_architecture.html

- [10] Arquitectura de Android. Los conceptos clave del sistema operativo Android. Consultado en junio de 2013. <http://www.android-app-market.com/android-architecture.html>
- [11] Google Play. Notebook. Consultado en junio de 2013. <https://play.google.com/store/apps/details?id=com.amstapps.xnotebookdemo&hl=en>
- [12] Google Play. Cuaderno del profesor. Consultado en junio de 2013. <https://play.google.com/store/apps/details?id=com.labeldroid.cuaderno.lite&hl=en>
- [13] Google Play. AndroClass - Teacher. Consultado en junio de 2013. https://play.google.com/store/apps/details?id=de.apenschi.androclass_t.main&hl=en
- [14] BlackBoard. Consultado en junio de 2013. <https://www.blackboard.com/platforms/mobile/overview.aspx>
- [15] iTunes. iDoceo, Cuaderno de notas del profesor. Consultado en junio de 2013. <https://itunes.apple.com/es/app/idoceo-cuaderno-notas-del/id477120941?mt=8>
- [16] Edmodo. Consultado en junio de 2013. <https://www.edmodo.com/>
- [17] Slideshare. Consultado en junio de 2013. <http://www.slideshare.net/>
- [18] Harry McCracken. iOS vs Android. Consultado en julio de 2013. <http://techland.time.com/2013/04/16/ios-vs-android/>
- [19] Ranking de sistemas operativos móviles más usados en 2013. Consultado en julio de 2013. <http://blog.uchceu.es/informatica/ranking-de-sistemas-operativos-mas-usados/>
- [20] Ventas de smartphones. Consultado en julio de 2013. <http://www.marketingdirecto.com/especiales/marketing-movil/las-ventas-de-smartphones-crecen-durante-el-primer-trimestre-de-2013-con-android-y-samsung-como-lideres/>

