

---

Diseño de una aplicación  
Learning Management System (LMS)  
para dispositivos móviles

---



TRABAJO FIN DE GRADO EN  
INFORMÁTICA/SOFTWARE

Daniel Serrano Torres  
David García Álvarez  
Juan Luís Pérez Valbuena

Departamento de Ingeniería de Arquitectura  
de Computadores y Automática  
Facultad de Informática  
Universidad Complutense de Madrid

Junio 2015



Diseño de una aplicación  
Learning Management System  
(LMS)  
para dispositivos móviles

*Memoria que presenta para optar al título  
de Graduado en Ingeniería en Informática/Software*

**Daniel Serrano Torres  
David García Álvarez  
Juan Luis Pérez Valbuena**

*Dirigida por el Doctor*

**José Miguel Montañana Aliaga**

**Departamento de Ingeniería de Arquitectura  
de Computadores y Automática  
Facultad de Informática  
Universidad Complutense de Madrid**

**Junio 2015**



*Sólo podemos ver poco del futuro, pero lo suficiente  
para darnos cuenta de que hay mucho que hacer.  
Alan Turing (1912-1954)*



# Autorización de difusión y utilización

Se autoriza a la Universidad Complutense a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la propia memoria, como el código, la documentación y/o el prototipo desarrollado.

Madrid, 18 de Junio de 2015

Fdo. Daniel Serrano Torres

Fdo. David García Álvarez

Fdo. Juan Luis Pérez Valbuena



# Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a nuestros padres (Miguel Ángel y Ana María, José Ramón y Lucita, José Luís y Antonia) por habernos proporcionado el apoyo necesario para poder llegar donde nos encontramos ahora.

También queremos agradecer a todas las personas que han colaborado directa o indirectamente con este proyecto.

A todos de corazón. . . ¡gracias!



# Resumen

Este proyecto trata el desarrollo de una herramienta para el soporte educativo del tipo LMS (Learning Management System) para dispositivos móviles, aportando un grado mayor de ubicuidad al proceso educativo orientado a la ingeniería. La herramienta se ha desarrollado en sobre tres plataformas móviles (Android<sup>TM</sup>, iOS<sup>TM</sup> y Windows Phone<sup>®</sup>) que abarcan la mayoría de la cuota de mercado, de esta forma se pretende llegar al máximo número de usuarios.

La herramienta consta de varias secciones principales: tablón, se muestran noticias publicadas por los usuarios; salas de chat, sistema de comunicación instantánea para trabajos colaborativos; exámenes, sección donde se pueden realizar exámenes de distintos tipos de preguntas; documentos, permite consultar y compartir tanto apuntes como exámenes.

El proyecto incluye el desarrollo de una aplicación Web en el lado servidor la cual se encarga de proveer y manipular los datos a través de un API que es utilizada por las aplicaciones móviles.

**Palabras Clave**— Educación, Plataformas móviles, Learning Management System, Bases de Datos, Aprendizaje Ubicuo.



# Abstract

This project is about the development of a Learning Management System for mobile devices, providing greater ubiquity of the educational process oriented engineering. The system has been developed in three different platforms (Android<sup>TM</sup>, iOS<sup>TM</sup> and Windows Phone<sup>®</sup>) covering the most of the market share, it is intended to reach the maximum number of users.

The system consists of several main sections: timeline, they are shown news publication from the users; chat rooms, a real-time messaging for collaborative works; exams, section which allows to perform exams with different kind of questions; documents, which allows to consult and share either exams and notes.

The project includes the development of an application web in server side which provides and manipulates the data through an API which is used by the mobile applications.

**Keywords**— Education, Mobile Platforms, Learning Management System, Databases, Ubiquitous Learning.



# Índice

<b>Agradecimientos</b>	<b>IX</b>
<b>Resumen</b>	<b>XI</b>
<b>Abstract</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
<b>2. Trabajo relacionado</b>	<b>5</b>
<b>3. Propuesta de trabajo</b>	<b>9</b>
3.1. Motivación . . . . .	9
3.2. Objetivos de la propuesta . . . . .	10
3.3. Secciones de la aplicación a implementar . . . . .	12
3.3.1. Tablón . . . . .	12
3.3.2. Chat . . . . .	13
3.3.3. Archivos . . . . .	15
3.3.4. Exámenes . . . . .	15
3.3.5. Web . . . . .	17
<b>4. Herramientas de trabajo y organización</b>	<b>19</b>
4.1. Documentación . . . . .	19
4.2. Control de versiones para el código fuente . . . . .	20
4.3. Integración continua . . . . .	22
4.4. Entornos de desarrollo . . . . .	23
4.4.1. Servidor . . . . .	23
4.4.2. Android™ . . . . .	25
4.4.3. iOS™ . . . . .	26
4.4.4. Windows Phone® . . . . .	31
<b>5. Decisiones de diseño</b>	<b>41</b>
5.1. Arquitecturas . . . . .	41

5.1.1.	Cliente-servidor . . . . .	41
5.1.2.	Arquitectura en tres capas . . . . .	42
5.1.3.	Arquitectura orientada a servicios . . . . .	43
5.1.4.	Elección de arquitectura . . . . .	43
5.2.	Application Programming Interface (API) . . . . .	43
5.3.	Sistemas de comunicación . . . . .	44
5.3.1.	XML . . . . .	44
5.3.2.	JSON . . . . .	44
5.4.	Estructura, diseño e implementación . . . . .	45
5.4.1.	Servidor . . . . .	45
5.4.2.	Aplicaciones móviles . . . . .	48
<b>6.</b>	<b>Trabajo Realizado</b>	<b>49</b>
6.1.	Servidor . . . . .	49
6.1.1.	Fundamentos básicos del API desarrollado . . . . .	49
6.1.2.	Sistema de Chat . . . . .	51
6.1.3.	Despliegue . . . . .	51
6.2.	Aplicaciones móviles . . . . .	52
6.2.1.	Primera ejecución . . . . .	53
6.2.2.	UC1: Registrar Usuario . . . . .	53
6.2.3.	UC2: Autenticación de usuario . . . . .	55
6.2.4.	UC3: Obtener mensajes tablón . . . . .	55
6.2.5.	UC4: Enviar mensaje tablón . . . . .	56
6.2.6.	UC5: Marcar mensaje favorito . . . . .	56
6.2.7.	UC6: Borrar mensaje . . . . .	57
6.2.8.	UC7: Agregar/Borrar asignatura a favoritos . . . . .	58
6.2.9.	UC8: Realizar examen . . . . .	58
6.2.10.	UC9: Visualizar estadísticas . . . . .	62
6.2.11.	UC10: Subir un examen/apuntes . . . . .	63
6.2.12.	UC11: Consultar examen/apuntes . . . . .	63
6.2.13.	UC12: Modificar perfil usuario . . . . .	64
6.2.14.	UC13: Crear sala de chat . . . . .	65
6.2.15.	UC14: Enviar mensajes chat . . . . .	66
6.2.16.	UC15: Recibir mensaje chat . . . . .	66
6.3.	Contribución al proyecto . . . . .	67
6.3.1.	Daniel Serrano Torres . . . . .	68
6.3.2.	David García Álvarez . . . . .	69
6.3.3.	Juan Luís Pérez Valbuena . . . . .	70
<b>7.</b>	<b>Discusión y conclusión</b>	<b>71</b>
7.1.	Discusión sobre objetivos . . . . .	71

---

7.1.1. Objetivos principales . . . . .	71
7.1.2. Objetivos extraordinarios . . . . .	72
7.2. E-Learning, futuro . . . . .	73
7.3. Trabajo futuro . . . . .	73
7.4. Conclusión . . . . .	74
<b>8. Discussion and conclusion</b>	<b>75</b>
8.1. Discussion on objectives . . . . .	75
8.1.1. Main objectives . . . . .	75
8.1.2. Extraordinary objectives . . . . .	75
8.2. E-Learning, future . . . . .	76
8.3. Future work . . . . .	77
8.4. Conclusion . . . . .	78
<b>Glosario</b>	<b>79</b>
<b>Acrónimos</b>	<b>81</b>
<b>A. ORM API en Django</b>	<b>83</b>
A.1. Modelos para el ejemplo . . . . .	83
A.2. Creando un nuevo registro en la base de datos . . . . .	84
A.3. Consultando todos los registros de la entidad Entry . . . . .	84
A.4. Relaciones entre tablas a través de los modelos . . . . .	84
<b>B. Enlaces de interés</b>	<b>85</b>



# Índice de figuras

1.1. Datos sobre cantidad de dispositivos extraídos de la fuente: [1].	2
1.2. Datos de cuota de mercado extraídos de la fuente: [2].	2
3.1. Tablón en Android™, iOS™ y Windows Phone®	12
3.2. Casos de Uso de Tablón	13
3.3. Casos de Uso de Salas de chat	14
3.4. Chat en Android™, iOS™ y Windows Phone®	14
3.5. Casos de Uso de Archivos	15
3.6. Archivos en Android™, iOS™ y Windows Phone®	16
3.7. Exámenes en Android™, iOS™ y Windows Phone®	16
3.8. Casos de Uso de Exámenes	17
3.9. Panel web	18
4.1. Centralizado	21
4.2. TravisCI	23
4.3. Eclipse™ con el complemento de PyDev™	24
4.4. Android: Servicio de Chat	26
4.5. Xcode® iOS™ IDE	27
4.6. Diagrama MVC en iOS™	27
4.7. StoryBoard en iOS™	28
4.8. Autolayout	29
4.9. User defaults	30
4.10. Core Data	30
4.11. Patrón delegado	31
4.12. Visual Studio	32
4.13. Nokia Lumia 520	32
4.14. Capas del MVVM	34
4.15. Ejemplo de IValueConverter	35
4.16. Descripción gráfica del IsolatedStorage	36
4.17. Ejemplo de ApplicationSettings	37

4.18. Explicación de la caché de imágenes y su interacción con el servidor . . . . .	38
4.19. Características almacenamiento interno WP . . . . .	39
5.1. Cliente-servidor . . . . .	42
5.2. Arquitectura en tres capas . . . . .	42
5.3. Inicio de sesión . . . . .	46
5.4. Inicio de sesión . . . . .	47
6.1. Estructura sistema . . . . .	50
6.2. Conexión al Chat . . . . .	52
6.3. Mensajes enviados por el servidor . . . . .	52
6.4. Mensajes enviados por los dispositivos . . . . .	53
6.5. Pantalla de inicio Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	54
6.6. Proceso de registro en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	54
6.7. Tablón en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	56
6.8. Escribir mensaje en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	57
6.9. Borrar mensaje en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	58
6.10. Agregar asignatura a favoritos en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	59
6.11. Selección de un tema para realizar un examen en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	59
6.12. Examen tipo text única respuesta Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	60
6.13. Examen tipo text multi-respuesta Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	61
6.14. Examen tipo respuestas cortas Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	62
6.15. Examen tipo parejas Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	62
6.16. Gráficas en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	63
6.17. Subir examen Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	64
6.18. Subir examen Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	65
6.19. Cambiar perfil Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	65
6.20. Chat en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	66
6.21. Chat en Android <sup>TM</sup> , iOS <sup>TM</sup> y Windows Phone® . . . . .	67

# Capítulo 1

## Introducción

La educación es uno de los pilares más importantes de la sociedad actual, puesto que cuantos más conocimientos y experiencia tiene una persona, más facilidad tendrá para encontrar un trabajo. Hoy en día estamos viviendo en la era digital, la tecnología ha hecho que muchas tareas de la vida cotidiana cambien y la educación está entre ellas.

Una de las tecnologías que más está destacando en este momento es la tecnología de los dispositivos móviles, en la cual podemos incluir desde smartphones, tablets, phablets hasta los nuevos smartwatch. Podemos decir que el auge de esta tecnología empezó en 2007 con la salida del primer iPhone<sup>TM</sup> [3] con iOS<sup>TM</sup>, desarrollado por Apple®. Un año más tarde, con la aparición de Android<sup>TM</sup> en el mercado [4] como sistema operativo open source para dispositivos móviles, otras compañías como Samsung<sup>TM</sup> se animaron a diseñar grandes terminales para aquellos tiempos, como por ejemplo el Samsung<sup>TM</sup> Galaxy S1. Viendo que el mercado de la tecnología móvil evolucionaba favorablemente, Microsoft<sup>TM</sup> también se incorporó anunciando su sistema operativo Windows Phone® para sus dispositivos.

En este momento se estima que 1.75 miles de millones de personas alrededor del mundo están usando al menos un dispositivo móvil [1] y la tendencia es que este valor crezca con el paso del tiempo, como podemos ver en la gráfica a continuación (figura 1.1):

La figura 1.2 muestra el porcentaje abarcado por cada sistema móvil del año 2014, se puede observar que el 99% [2] de la cuota de mercado está comprendido por los sistemas Android<sup>TM</sup>, iOS<sup>TM</sup>, Windows Phone®.

El término Electronic Learning (E-Learning) aparece en 1999 [5], siendo utilizado este término por primera vez en los seminarios de sistemas CBT (Cognitive Behavioral Therapy).

El primer sistema de E-Learning fue introducido en los años setenta,

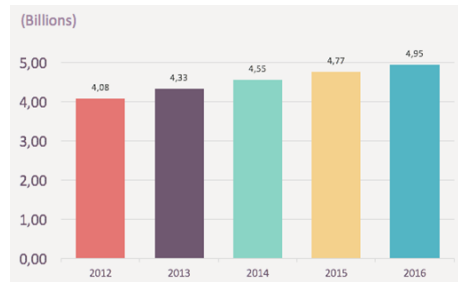


Figura 1.1: Datos sobre cantidad de dispositivos extraídos de la fuente: [1].

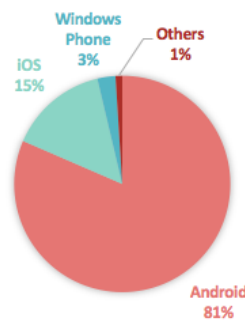


Figura 1.2: Datos de cuota de mercado extraídos de la fuente: [2].

construido para distribuir información entre los estudiantes. A partir de entonces los sistemas E-Learning comienzan a ser más interactivos.

Con la expansión de la computación e Internet las herramientas de E-Learning se expandieron, junto con la llegada de computadores para uso doméstico (el primer MAC®) se facilitó el aprendizaje de temas particulares y desarrollo de habilidades por parte de los usuarios. Durante los siguientes diez años estos sistemas de enseñanza prosperaron notablemente. Llegando al entorno laboral sistemas de aprendizaje a distancia a partir del año 2000.

Dejando a un lado la historia del E-Learning se van a exponer algunas de las ventajas más notables que este sistema de aprendizaje aporta.

- No existen restricciones de espacio ni de tiempo para aprender.
- Un curso diseñado para ser interactivo hace el aprendizaje más atractivo y distendido.
- Disminuye costes para ambas partes implicadas en el curso.

Como ya desde el pasado ha venido ocurriendo una convergencia educa-

ción y tecnología, lo que ha llevado a la aparición del término E-Learning, el cual trata sobre cómo integrar la tecnología en general con la educación de los estudiantes de la forma más efectiva posible. Algunos autores se han atrevido incluso a crear subcategorías de E-Learning según la tecnología que se use para enseñar a dichos estudiantes, algunos de los ejemplos son:

- **M-Learning:** estilo de enseñanza en la cual el medio son los dispositivos móviles.
- **U-Learning:** tiene cierta similitud con el M-Learning, con la diferencia de que es más amplio, no solo se basa en dispositivos móviles, y busca una experiencia más distribuida en el tiempo y el espacio [6, 7].
- **T-Learning:** enseñanza vía TV gracias a la multiplexación de canales que nos ofrece la TDT [8].

Existen ya muchas tecnologías E-Learning en uso y de forma muy extendida como pueden ser las plataformas Moodle<sup>TM</sup> y Sakai<sup>®</sup> definidas como Learning Manager System (LMS), existen más que las mencionadas pero no se van a tratar.

Moodle es una plataforma de código abierto para gestionar los cursos de alumnos matriculados en una institución de enseñanza, permite administrar las asignaturas de forma independiente, gestionar material de aprendizaje, comunicación entre estudiantes y profesores, etc. El carácter de proyecto libre otorga el beneficio de las aportaciones de la comunidad, esto hace que Moodle<sup>TM</sup> pueda mejorar como plataforma gracias a los complementos y cambios informporados por la comunidad de desarrolladores. Si bien es una plataforma completa de gestión para la enseñanza carece del carácter ubicuo que puede otorgar una aplicación para sistemas móviles, ya que Moodle depende de utilizar un navegador de internet y no siendo adaptable de manera adecuada a pantallas de dispositivos móviles.

Por otro lado está Sakai<sup>®</sup>, es una plataforma similar a Moodle<sup>TM</sup> que aporta alguna característica nueva como puede ser el directorio de compartición de recursos y la capacidad de compartir mayor variedad de recursos (audio, portfolios, etc). Pero al igual que Moodle carece de la adaptación correcta a los dispositivos móviles que facilite su uso desde cualquier lugar.

Por las carencias descritas en las plataformas LMS Moodle<sup>TM</sup> y Sakai<sup>®</sup> se ha decidido desarrollar el proyecto cuya propuesta se expone más a delante.

Se ha explicado cuando surge el E-Learning, que es e incluso plataformas LMS que muestran el potencial de este en la educación. El E-Learning

realiza una gran aportación a la educación eliminando la barrera temporal. Con este estilo de herramientas los estudiantes pueden aprender en el momento que deseen, puesto que no necesitan un profesor presencial, el propio dispositivo electrónico les brinda lo que necesitan. Si a esto le añadimos que las tecnologías móviles, a diferencia de los PC y las TV, son más manejables eliminando así la barrera espacial. Con estas dos barreras eliminadas los alumnos podrán estudiar cuándo y dónde quieran.

Desde un punto de vista social se debe prestar atención al fenómeno de crecimiento masivo, que el uso de las tecnologías móviles ha provocado en las Redes Sociales, de esta forma han llegado a su máximo esplendor. No es posible saber cuál y cuándo apareció la primera Red Social en Internet, pero lo que sí podemos mencionar es que una de las redes sociales pioneras y más grandes en la actualidad a nivel mundial es Facebook<sup>TM</sup>. Facebook<sup>TM</sup> fue fundada en el 2004 [9] y ahora mismo es la red social con más usuarios registrados en el mundo, teniendo más del 80 % de los usuarios de Internet [10].

Aunque Facebook<sup>TM</sup> sea posiblemente la red social más importante en estos momentos, no podemos dejar de mencionar a otras redes sociales que también poseen una gran cantidad de usuarios registrados y que además tienen un estilo diferente al de Facebook<sup>TM</sup>. Estas redes sociales son Youtube<sup>TM</sup> (el 60 % de los usuarios de internet tienen cuenta), Google+<sup>TM</sup> (60 %), Twitter<sup>TM</sup> (53 %).

Dada la aceptación por parte de los usuarios de Internet hacia las redes sociales, es interesante tener cuenta su enfoque a la hora de desarrollar una aplicación, sobre todo si denota algún carácter interactivo y/o grupal. Además el componente social es un elemento que puede favorecer en gran medida el E-Learning [11, 12, 13, 10], es una gran idea el hecho de introducir características (como por ejemplo sociales, colaborativas, etc...) en un entorno E-Learning; por un lado los usuarios están familiarizados con estas funcionalidades y por otro fomenta el intercambio, transferencia y desarrollo del conocimiento.

Por ello este proyecto incorpora un componente social para que los usuarios colaboren, difundan y se comuniquen en el entorno educativo.

## Capítulo 2

# Trabajo relacionado

A continuación se van a tratar trabajos previos en E-learning, la motivación y objetivos de dichos trabajos antes de abordar la propuesta de nuestro proyecto.

Actualmente es un momento más favorable para el uso de E-Learning como paradigma de aprendizaje, Chris Jones y sus compañeros de trabajo han realizado un estudio [14] con estudiantes de lo que han denominado “digital-natives” o “net-generation”, donde han tratado de buscar claros patrones mientras desarrollan estudios que integran el E-Learning. Este estudio compara dos grupos de estudiantes de primer curso de carrera donde unos usan algunas nuevas tecnologías y el otro conjunto realiza un uso extenso de las nuevas tecnologías, dándose cuenta que los estudiantes nacidos a partir de 1983 pertenecen a un sola generación. Si bien concluyen que de entre los estudiantes de primer año hay variaciones significativas en cuanto al uso de las nuevas tecnologías seguimos pudiendo afirmar que son una nueva generación que ha asimilado la tecnología de una manera más natural y profunda, y ello nos lleva a tener una generación más favorable en la adopción del E-Learning como parte de su proceso de aprendizaje. El artículo recoge datos del acceso de los estudiantes a medios digitales que pueden facilitar el uso de tecnologías E-Learning (ordenadores de escritorio, ordenadores portátiles), pero hoy día el 99% [2] de los estudiantes ya disponen de un smartphone propio lo que nos permite llevar el E-Learning un poco más lejos y por lo cual surge este proyecto.

Por otro lado ya han surgido muchas iniciativas de E-Learning hasta hoy, y la gran mayoría acaba por tener un uso discontinuado por parte de los estudiantes. El autor de [15] Ming-Chi Lee ha realizado un estudio para explicar por qué ocurre esta discontinuación y para tratar de predecirla, en el estudio se afirma que la aceptación inicial de un sistema E-Learning es fundamental para que se produzca algún tipo de continuación en su uso y

que por tanto es necesario comprender los factores que afectan a los usuarios/estudiantes en la continuación del uso de tecnologías E-Learning. Como se comenta en el primer párrafo la nueva generación de estudiantes también tiene una nueva manera de funcionar, tienen más dificultad para seguir un proceso lineal de aprendizaje ya que tienen una mayor capacidad de realizar multitarea, necesitan un rápido acceso a la información y un sistema de recompensas para mantener el interés. Es por ello que nuestro proyecto está enfocado a las tecnologías móviles, otorga así un rápido acceso al estudiante en cualquier momento y lugar, así como la obtención inmediata de sus calificaciones, fallos y aciertos.

Los investigadores Charles Calderwood, Phillip L. Ackerman, Erin Marie Conklin en su trabajo [16] han estudiado los problemas que ha generado la tecnología en los estudiantes, siendo uno de estos la multitarea, durante las clases o los procesos de estudio.

Durante el seguimiento de un proceso de estudio de 3 horas de un grupo de estudiantes, los datos obtenidos determinan que, de media, los estudiantes sufren 35 distracciones de aproximadamente 3 segundos durante el periodo de estudio ocupando 26 minutos en total de las 3 horas. De este grupo el 59% de los estudiantes escuchan música como parte del proceso de estudio, la media de la cantidad de tiempo gastado en esta distracción aumenta a 73 minutos del proceso de estudio (sobre el 40% de la sesión). El trabajo concluye que los estudiantes que consiguen mayor concentración durante el proceso de estudio son aquellos que durante este sufren el menor número de distracciones y de menor duración.

Conocer por qué muchas soluciones dentro del campo del E-Learning acaban fracasando es importante para el desarrollo de soluciones en este campo, los factores que afectan al abandono de estos métodos de aprendizaje han sido estudiados por los autores del artículo [17], Pei-Chen Sun, Ray J. Tsai, Glenn Finger, Yueh-Yang Chen, Dowming Teh. En dicho estudio han encontrado factores como la ansiedad del estudiante, la predisposición del profesor a estas tecnologías y metodologías, la calidad y flexibilidad del curso ofertado, etc.

Los estudiantes de este estudio se han visto envueltos en 16 cursos diferentes de dos universidades en Taiwán, de los datos obtenidos a través de encuestas, de donde el 45,7% de las respuestas han sido válidas para el estudio y denotan que los factores indicados anteriormente y algunos más (consultar el documento para más información) son críticos a la hora de afectar al estudiante en términos de satisfacción para que no abandone el proceso de aprendizaje.

---

Particularizando el tema de la multitarea, mencionado en los párrafos anteriores, durante los procesos de aprendizaje nos encontramos con la situación particular del uso de ordenadores portátiles en las aulas. En el estudio[18] de los autores Faria Sana, Tina Weston, Nicholas J. Cepeda se analiza el rendimiento de los estudiantes que realizan procesos multitarea directamente, indirectamente y de los que no lo hacen a través de sus calificaciones en los test. A través del estudio de 44 estudiantes de una universidad Canadiense, en dos experimentos diferentes de comprensión lectora los autores concluyen que los estudiantes que no realizan multitarea tiene mejores calificaciones frente a los que si lo hacen en sus tests.

Los móviles ahora son también parte del E-Learning, Jorge L. V. Barbosa, Rodrigo Hahn, Débora N. F. Barbosa, Willian Segatto han realizado un estudio a través del uso intensivo de las tecnologías móviles [19] en el proceso de aprendizaje. Dentro del estudio se han realizado tres proyectos diferentes (un juego donde los estudiantes tienen que desplazarse para completar las tareas con objetos reales, control de robots a través del móvil que interactúan con el medio y aplicaciones para aprender diferentes asignaturas) de los cuales extraen conclusiones donde los estudiantes refuerzan sus conocimientos sobre robótica a través de las tecnologías móviles, mencionan que han conseguido generar motivación en el estudiante al realizar procesos más prácticos de enseñanza.

El término Mobile and Ubiquitous Learning descrito en el capítulo uno, es estudiado en el anterior artículo y en otros más [19, 20, 21, 22, 23]. Haciendo referencia al hecho de poder llevar la enseñanza más allá de las aulas e incluso del ordenador, otorgando esto muchas más posibilidades para el desarrollo de métodos de enseñanza diferentes de los tradicionales.

La utilización de la computación como herramienta en la educación cada vez es más interesante y produce mejores resultados para las personas que los utilizan, esto se da gracias a la evolución de la tecnología y los sistemas informáticos como por el hecho de la necesidad evolutiva de los métodos y modelos de enseñanza. Al igual que sistemas informáticos basados en agentes [6], sistemas que ayudan al aprendizaje de las materias de los alumnos gracias a la asistencia interactiva a través del ordenador personal, otras utilidades sobre nuevas plataformas (tablets, teléfonos móviles, etc) son de gran ayuda para mejorar los resultados de los alumnos [7]. Los sistemas desarrollados en estos estudios han dado resultados a sus autores que muestran su eficiencia y que los estudiantes que lo usaron aprendían más contenidos.

Otro estudio por parte de Natasa Hoic-Bozic, Verdán Mornar, Ivica Boticke [24] describe un enfoque metodológico para conseguir un modelo de

enseñanza mixto y transitorio hacia el E-Learning, manteniendo el soporte de al alumno del método tradicional de enseñanza y un aprendizaje activo con las nuevas tecnologías, lo cual radica en el aumento de la motivación de los estudiantes. Obteniendo como resultados del estudio la mejora en las calificaciones de los participantes.

Una parte muy importante de cualquier sistema informático es la privacidad, y más cuando se trata de información sensible sobre las personas. Como se discute en este artículo [25] de los autores Vanja Bevanda, Jasmin Azenović, Denis Mušić es muy complejo desarrollar grandes sistemas que mantengan un alto grado de seguridad de la información y que proteja el contenido, en este caso, de los estudiantes. En este trabajo exponen la necesidad de utilizar sistemas como las bases de datos hipocráticas y enumeran diez principios a seguir para tratar de garantizar la máxima seguridad en el sistema desarrollado.

Concluyen la necesidad de que las bases de datos de los entornos E-Learning deben cumplir con las formas normales para garantizar seguridad del modelo de datos.

## Capítulo 3

# Propuesta de trabajo

Se ha decidido desarrollar una aplicación Learning Management System (LMS) para dispositivos móviles.

Será multiplataforma (Android™, iOS™, Windows Phone®) lo cual hace que esté disponible para el 99% [2] de los usuarios de dispositivos móviles.

Al ser desarrollado este proyecto para dispositivos móviles favorece una característica descrita anteriormente que es muy útil de cara a una mejor adopción de una aplicación M-Learning, se trata de la ubicuidad. Este factor favorece el uso de la aplicación ya que el usuario no depende de un lugar y una estación de trabajo fijos para aprender. Existiendo iniciativas europeas como la red Eduroam [23], que permite que los alumnos de las universidades tengan conexión a Internet de forma inalámbrica en todo el campus universitario.

### 3.1. Motivación

La motivación de desarrollar este proyecto va ligada directamente con los objetivos descritos en los capítulos uno y dos. El proyecto pretende ser una herramienta de ayuda al estudio, para mejorar calificaciones, disminuyendo esos factores ya estudiados en el capítulo dos (multitarea, distracciones, etc), aprovechar que la nueva generación de estudiantes traen un conocimiento intrínseco de las tecnologías para llegar a ellos de mejor manera en la enseñanza, aprovechar las nuevas tecnologías que pueden brindar nuevas maneras de enseñar.

Por otro lado y desde el punto de vista tecnológico, los participantes en el desarrollo de este proyecto hemos aprovechado para aprender sobre desarrollo móvil, nuevos lenguajes de programación, arquitecturas, patrones, etc. Elementos que han comprendido e involucrados durante todo el proceso

de desarrollo.

### 3.2. Objetivos de la propuesta

A continuación se va a describir los objetivos que este proyecto trata de abarcar.

Se quiere crear un sistema de noticias que se llamará tablón, donde los usuarios podrán compartir novedades y votar las noticias más relevantes. Además el usuario podrá elegir si su mensaje se publica de forma anónima.

Un sistema de mensajería instantánea que permitirá salas de conversación públicas y privadas (la información de las conversaciones no se almacena), que pueden estar constituidas por un gran número de usuarios.

Otro objetivo es crear un sistema para que el usuario pueda realizar exámenes, controles, etc. Estos serán de varios tipos: de preguntas con una única respuesta correcta, preguntas con múltiples respuestas correctas, preguntas con respuestas cortas y preguntas de emparejamiento de respuestas.

En la aplicación se quiere que cada usuario tenga un espacio donde ver su información y gestionar cambios de facultad y contraseña. Para ello se tiene como objetivo crear una sección de perfil de usuario donde permitir las funciones anteriores, mostrar las estadísticas de los exámenes y generales, así como permitir cambiar la fotografía identificativa del usuario.

Para que los profesores puedan crear exámenes y administrar la información podrán acceder a un panel administrativo desde la aplicación, este solo estará accesible para los usuarios que son profesores estrictamente.

Y como último objetivo principal se quiere crear un sistema de compartición de ficheros (formato imagen) para que se puedan compartir exámenes, apuntes e información docente.

Estos seis objetivos son los que se consideran necesarios para desarrollar el proyecto con las características deseadas, pero además se desea llegar a abarcar una serie de objetivos que proceden de circunstancias derivadas del uso de la tecnología en las aulas y problemas derivados de los sistemas E-Learning.

Uno de los objetivos extraordinarios, el cual ya ha sido expuesto en el capítulo dos, como factor que afecta negativamente a un proceso de enseñanza E-Learning, es tratar de reducir la distracción de los alumnos durante

las lecciones y así también la multitarea en estas. Uno de los elementos que genera distracciones en la aulas actualmente es el dispositivo móvil, que como se ha mencionado anteriormente abarca el 99 % de los estudiantes, por ello el hecho de tener una plataforma de enseñanza en estos que apoye las lecciones del profesor puede ayudar a minimizar el uso de los dispositivos móviles para tareas ajenas a las lecciones. Y la multitarea es abordada desde el punto que un profesor puede preparar controles o exámenes fuera de las clases permitiendo que los alumnos no necesiten utilizar una fuente externa para los enunciados o copiarlos mientras el profesor prosigue la lección para realizar finalmente la tarea.

Una de las dificultades que a veces se presenta en el ámbito de estudio es el trabajo en equipo, en colaboración. Siendo uno de los factores que llevan a que se convierta en una dificultad el hecho de necesitar un ordenador para desarrollar el trabajo o reunir al grupo de trabajo, por ello el desarrollo de este proyecto se realiza sobre plataformas móviles, pretendiendo así conseguir que realizar el trabajo pueda ser lo más ubicuo posible.

Otro de los objetivos que se ha pretendido abordar en el proyecto es facilitar la difusión de conocimiento en el ámbito de estudio, creando para ello un sistema de compartición de ficheros (imágenes) estructurado y categorizado para facilitar su búsqueda. De esta manera se puede compartir información de manera sencilla y completamente relacionada con el campo de estudio.

Uno de los puntos expuestos en el capítulo dos es el abandono de sistemas E-Learning transcurrido un tiempo de uso. Esto está en parte relacionado, como ya se ha expuesto, con la motivación y otros sentimientos del individuo, pero también con el hecho de la tarea de aprender sobre sistemas digitales deba realizarse en lugares concretos a los cuales un usuario no puede recurrir cuando desee. Por ello reincidimos en la decisión de hacer el proyecto sobre dispositivos móviles, que puede ayudar a que si, un usuario desea realizar en cualquier momento una tarea o compartir/obtener información, lo pueda hacer desde su dispositivo móvil; incluso aprovechando momentos del día en los que de otra forma no utilizaría, como, por ejemplo, desplazamientos en transporte público.

### 3.3. Secciones de la aplicación a implementar

Hemos dividido la funcionalidad en distintos bloques, que cada plataforma móvil adaptará en función de los patrones de diseño propios de dicha plataforma. Es decir, la funcionalidad de cada plataforma móvil será idéntica, pero la experiencia de uso de cada una de ellas puede variar.

#### 3.3.1. Tablón

Un tablón es un lugar donde se asocian mensajes cortos que han sido enviados por los usuarios. Cada facultad tiene asociado un tablón, pudiendo un mensaje sólo estar asociado a una facultad y a un tablón.

Fomentamos el aprendizaje colaborativo a través de un tablón donde los usuarios podrán preguntar sus dudas o simplemente comentar cualquier tema que les parezca oportuno, e incluso compartir noticias sobre los temas que deseen (figura 3.1). Cada facultad tiene asociado un tablón, pudiendo un mensaje sólo estar asociado a una facultad y a un tablón.

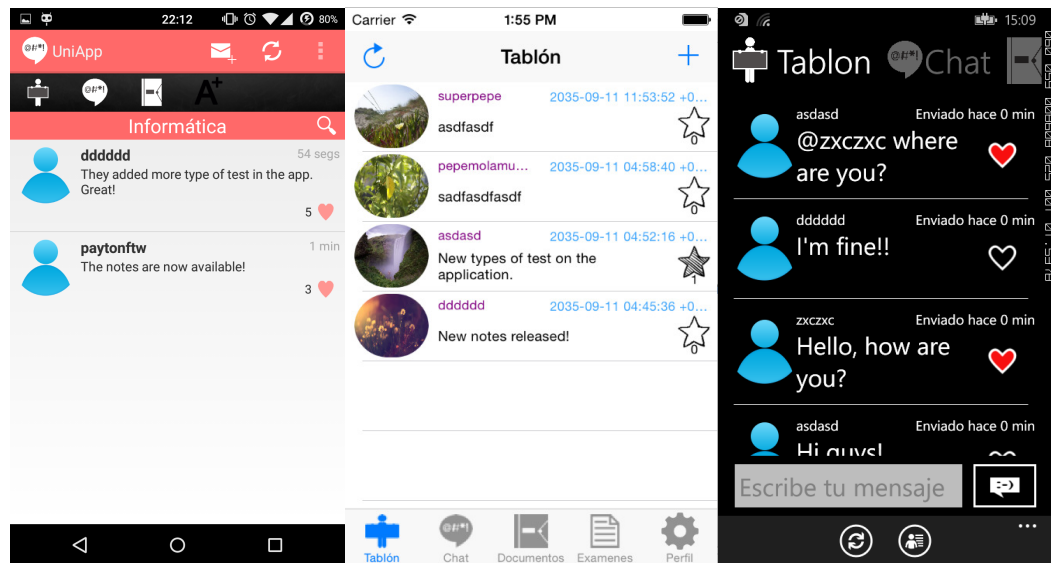


Figura 3.1: Tablón en Android™, iOS™ y Windows Phone®

Los usuarios podrán interactuar con el tablón de las siguientes maneras, como se aprecia en el diagrama 3.2:

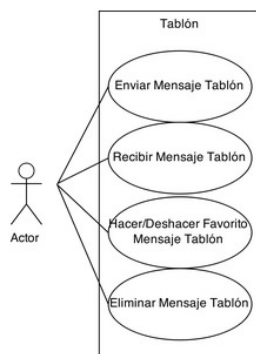


Figura 3.2: Casos de Uso de Tablón

Un usuario envía un mensaje a un tablón, el usuario compone un mensaje de texto corto ( menos de 140 caracteres), elige si desea enviar el mensaje de forma anónima o no. Se pulsa el botón de enviar para que el mensaje aparezca en el tablón.

Un usuario recibe mensajes de un tablón cuando inicia sesión o pulsa el botón de actualizar mensajes tablón.

Un usuario hace/deshace favorito un mensaje de un tablón pulsando el botón de hacer/deshacer favorito un mensaje.

### 3.3.2. Chat

Esta sección será diseñada para fomentar el trabajo en equipo y colaborativo, permitirá que los usuarios trabajen con un sistema de comunicación en tiempo real para realizar tareas en grupo. Es por ello que habilitaremos un sistema de mensajería instantánea (figuras 3.4) que nos permitirá comunicarnos con quien deseemos de manera instantánea.

El sistema de chat se se basará en el Internet Relay Chat [26] para proveer una integración de de chat en nuestra aplicación, incluyendo mejoras de autenticación (se utiliza la propia autenticación de la aplicación) y el uso del protocolo JSON como método para el recepción y envío de datos.

Se ha determinado el nombre de “Sala” a cada uno de los canales definidos por el protocolo IRC [26].

Los usuarios podrán interactuar con el Chat de las siguientes maneras, detalladas en el siguiente diagrama 3.3:

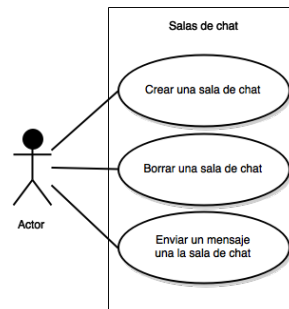


Figura 3.3: Casos de Uso de Salas de chat

Un usuario, simplemente por el hecho de conectarse a la aplicación, se conectará a la Sala de su Facultad.

Por otra parte, se permitirá la creación de Salas con nombre personalizado. Si al introducir el nombre no existía, se creará una nueva y sino, simplemente se unirá a dicha sala.

Finalmente el usuario, a través de la interfaz, elegirá de que Sala quiere recibir/enviar mensajes.

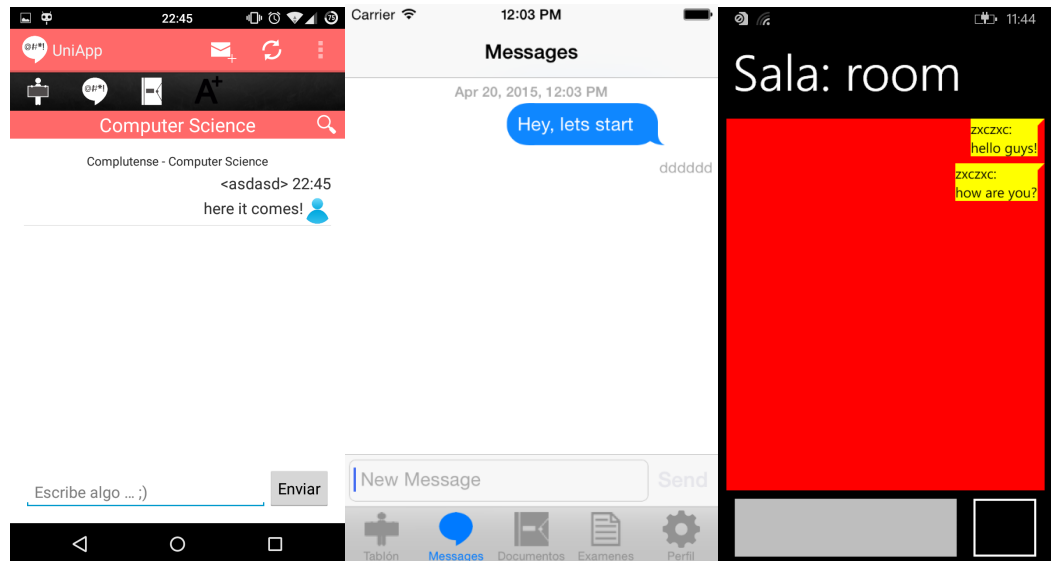


Figura 3.4: Chat en Android™, iOS™ y Windows Phone®

### 3.3.3. Archivos

Esta sección complementará las dos anteriores proporcionando un sistema de compartición de ficheros para que los alumnos puedan compartir documentación relevante al estudio.

Se distingue entre apuntes y exámenes de años anteriores.

Cada archivo pertenece a una asignatura o a un tema de una asignatura para facilitar la búsqueda. También se añadirá una fecha para que el usuario pueda tener en cuenta la antigüedad del archivo subido (figuras 3.6).

Los usuarios podrán interactuar con los Archivos de las siguientes maneras (diagrama 3.5):

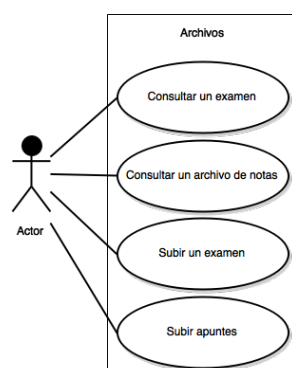


Figura 3.5: Casos de Uso de Archivos

Un usuario podrá visualizar archivos, ya sean apuntes o exámenes, a través de una asignatura o tema.

Un usuario podrá subir archivos, indicando los datos del mismo y eligiendo, a través de la interfaz, la imagen o imágenes que desea subir.

### 3.3.4. Exámenes

En este apartado, se incluye la parte más académica de nuestro LMS. Los profesores podrán componer preguntas para que los alumnos puedan responderlas. La idea es que sean preguntas conceptuales, que resaltan conceptos clave de la asignatura en cuestión y su modo de responder sea intuitivo y fácil.

No tenemos intención de obligar al alumno a resolver un problema complejo a través de un dispositivo móvil, sino de resolver muchos problemas sencillos con conceptos clave.

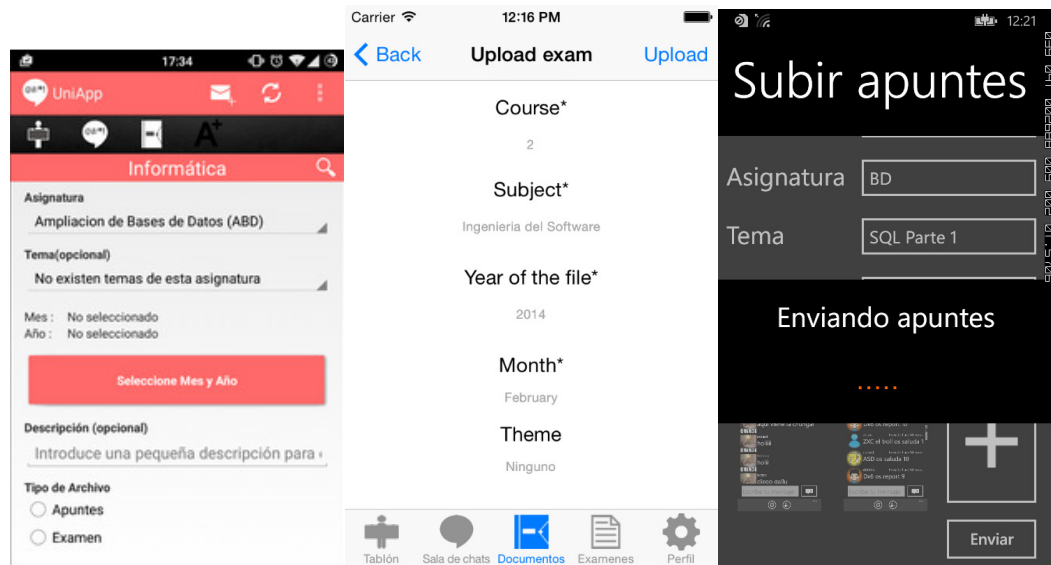


Figura 3.6: Archivos en Android™, iOS™ y Windows Phone®

Se implementarán cuatro tipos de tests, preguntas de única respuesta, preguntas de respuesta múltiple, preguntas de respuesta breve y por último preguntas de emparejamiento de respuestas (figuras 3.7).

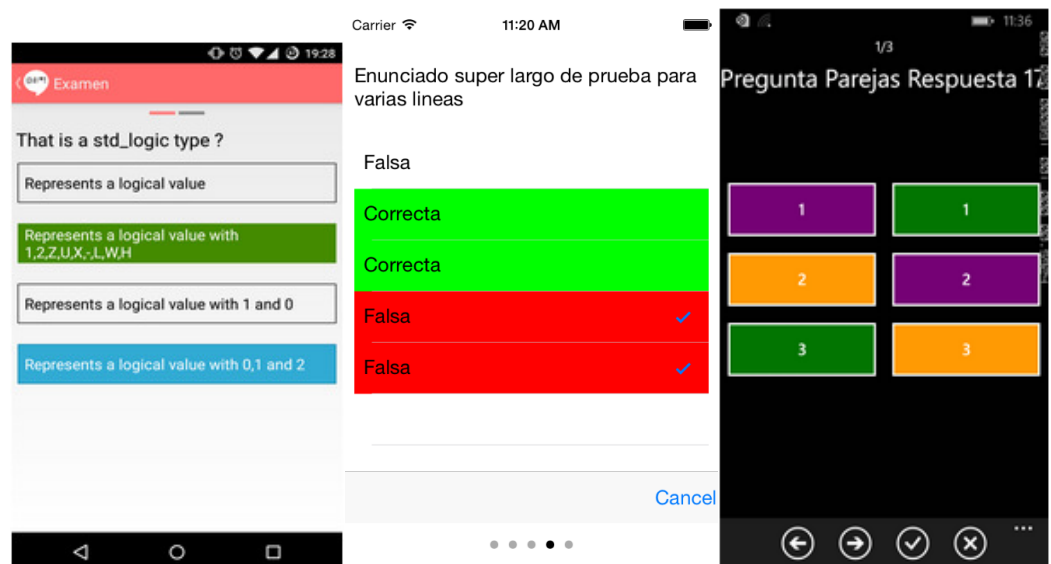


Figura 3.7: Exámenes en Android™, iOS™ y Windows Phone®

Los usuarios podrán interactuar con los Exámenes de las siguientes maneras, como muestra el diagrama 3.8:

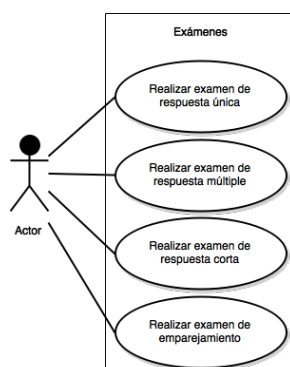


Figura 3.8: Casos de Uso de Exámenes

Un usuario podrá acceder a la sección de exámenes y elegir asignaturas de la lista habiendo seleccionado previamente el curso de la asignatura. Posteriormente aparece una lista de temas de la asignatura seleccionada, al elegir uno de estos el usuario podrá escoger el tipo de examen a realizar y enviar los resultados cuando termine de realizarlo.

### 3.3.5. Web

Por último se desarrollará un panel de administración (figura 3.9) el cual permitirá, de forma cómoda, la gestión del sistema, tanto para administradores del sistema como para profesorado. Esta parte Web, estará disponible dentro del sistema (no haría falta una aplicación externa para la gestión), pero para mayor comodidad, se permite también el acceso desde un dispositivo cualquiera conectado a Internet.

Un usuario con privilegios (profesor) podrá:

Componer cualquier tipo de preguntas  
Obtener los resultados de los test realizados

Un usuario administrador podrá:

Realizar cualquier gestión relacionada con el sistema, ya que tendrá acceso total al mismo.

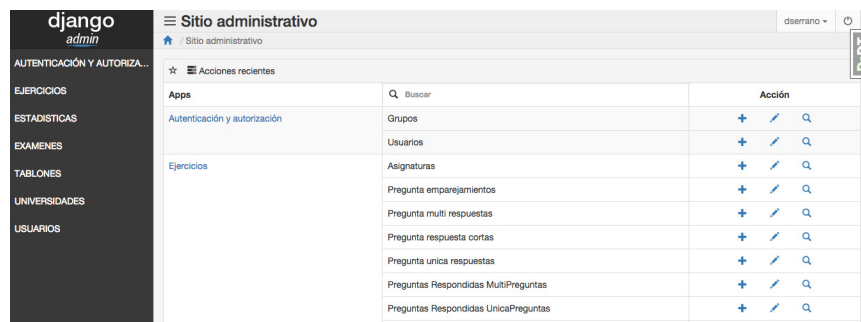


Figura 3.9: Panel web

## Capítulo 4

# Herramientas de trabajo y organización

En este capítulo se explicarán las distintas herramientas de trabajo y organización usadas por el equipo.

Al ser un trabajo realizado por más de una persona, la organización y el trabajo en equipo es fundamental. Es por ello que se haya hecho uso de múltiples herramientas para facilitar la tarea conjunta.

### 4.1. Documentación

Se ha hecho uso de dos herramientas para la generación de la distinta documentación requerida por el trabajo.

Para realizar documentación interna, se ha decidido utilizar el Servicio de almacenamiento en línea Google Drive<sup>TM</sup>.

Dicha plataforma otorga muchas ventajas para el trabajo en equipo. Por un lado permite la compartición de documentos de forma sencilla y manteniendo la privacidad de estos a usuarios ajenos al proyecto. también proporciona un sistema de control de cambios sobre cada documento que hace mucho más fácil volver a una versión anterior o simplemente conocer qué cambios exactos en el documento se han realizado.

Por otra parte y para la generación de documentación externa, como por ejemplo la realización de esta memoria, se ha utilizado Overleaf<sup>TM</sup> que nos permite, de forma colaborativa, editar un documento de tipo  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$  de manera gratuita (esta memoria ha sido confeccionada en  $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ ).

Como característica quizá más útil e importante (al menos para este proyecto) es la capacidad que permiten el Servicio de almacenamiento en línea Google Drive™ y Overleaf™ de edición en colaborativa en tiempo real, permitiendo así que varios de los componentes de este proyecto trabajen en un mismo documento al mismo tiempo.

## 4.2. Control de versiones para el código fuente

Se ha utilizado Git como herramienta de control de versiones para todos los códigos fuente desarrollados en este proyecto.

En este apartado se intenta explicar Git y sus fundamentos básicos.

Git es un sistema de control de versiones diseñado para tratar con los proyectos Software sea cual sea su tamaño con rapidez y eficacia.

Git tiene múltiples ventajas con respecto a otros sistemas de administración de código fuente, ya puedan ser Subversion, CVS, Perforce u otros.

- Branchs (ramas): Se puede disponer de múltiples ramas, una para el código en producción, otra para pruebas y otras para distintas características que se estén desarrollando para la aplicación.
- Ligero y rápido: muchas de las operaciones son realizadas de forma local, dando ventaja a los sistemas centralizados que necesitan comunicarse con un servidor constantemente.
- Seguridad de datos: para evitar corrupción en los datos, cada subida de código tiene una suma de comprobación que se realiza antes de subir en el cliente y una vez subido en el servidor.
- Software libre, y por supuesto, gratuito.
- Distribuido: Una de las mejores características de Git es que necesitas “clonar” el repositorio entero. Esto ofrece múltiples ventajas:
  - Múltiples copias de seguridad
  - Se permite usar cualquier “flujo de trabajo” ya sea:
    - Centralizado: todos los desarrolladores suben cambios al mismo repositorio (digrama 4.1).

- “Blessed Repository”: los desarrolladores suben cambios a su repositorio privado y solicitan a un usuario (manager de integración) que suba sus cambios al “Blessed Repository”. Se usa para repositorios de GitHub y otros de código abierto.
- “Dictador” y “tenientes”: Parecido al anterior pero hay varios usuarios responsables (“tenientes”) de distintos subsistemas que aprueban los cambios que otros desarrolladores han subido. Existe un usuario por encima de los tenientes que sube los cambios al “Blessed Repository”.

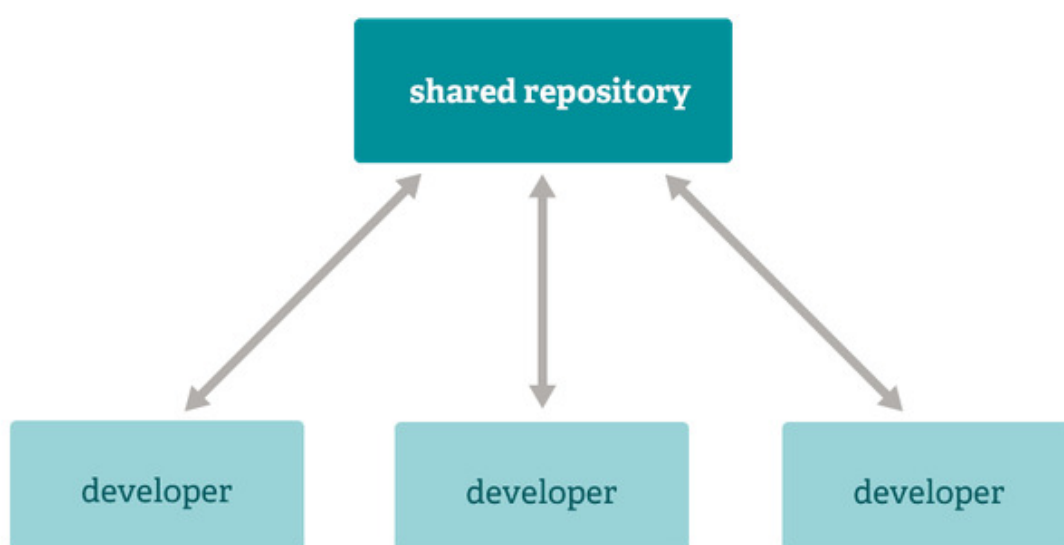


Figura 4.1: Centralizado

GIT dispone de múltiples herramientas gráficas, pero dispone de una potente línea de comandos donde poder realizar todas las operaciones.

A diferencia de otros sistemas de control de versiones como SVN, GIT guarda y procesa la información de forma muy diferente, aunque a través de la interfaz de usuario sea parecida.

**Instantáneas, no diferencias:** Esto afecta en como GIT guarda los datos. Conceptualmente los otros sistemas (SVN, etc.) guardan la información como una lista de cambios en los archivos. Estos sistemas tienen la información como un conjunto de archivos y los cambios que se han ido realizando a lo largo del tiempo, así funciona SVN y similares.

Sin embargo, GIT no guarda la información de esta manera. GIT almacena distintas instantáneas de un pequeño sistema de archivos. Cada vez que

se añada cambios o se guarda el estado del proyecto en GIT, realiza una ‘instantánea’ de todos los archivos hasta el momento y guarda una referencia a esta ‘instantánea’. Para ser eficiente, si algún archivo no ha sido modificado, se guarda una referencia al archivo anteriormente guardado.

**Prácticamente todas las operaciones son en local:** La mayoría de las operaciones en GIT solo necesitan archivos locales y recursos para funcionar. En la mayoría de los casos no es necesaria información de otros ordenadores de tu red debido a que se dispone de todo el historial del proyecto en nuestro disco local.

**GIT tiene integridad:** Antes de ser guardado, toda la información pasa por una suma de comprobación, esto hace que sea imposible de cambiar los contenidos de cualquiera de los archivos o directorios sin que GIT lo sepa. El mecanismo que usa GIT para la suma de comprobación es SHA-1. Se genera una cadena de 40 caracteres hexadecimales (0-9 y de la a-f) calculado según los contenidos de un archivo o directorio.

### 4.3. Integración continua

En el desarrollo de software la automatización de procesos es fundamental, incluso más importante durante el ciclo de desarrollo ya que permite agilizar y simplificar este. Por ello para este proyecto se ha incorporado un sistema de integración continua que permite comprobar que la aplicación compila y termina satisfactoriamente la ejecución de las pruebas.

Dicha herramienta es TravisCI, que a su vez se integra con una cuenta de GitHub permitiendo de esta forma acceder al repositorio donde se aloja el proyecto para realizar los procesos de compilado y ejecución de las pruebas unitarias. En la figura 4.2 se puede ver un fragmento de la interfaz de TravisCI usando el repositorio de este proyecto.

El sistema de integración continua permite una configuración muy amplia para un proyecto, en el caso de este proyecto se ha decidido utilizar para comprobar que realiza correctamente la fase de compilado y pruebas. En caso de detectar cualquier error lo notifica por email a los usuarios que se hayan configurado en el fichero de configuración para TravisCI (`.travis.yml`, denotando el punto primero un fichero oculto en sistemas Unix) que debe permanecer en el directorio raíz del proyecto.

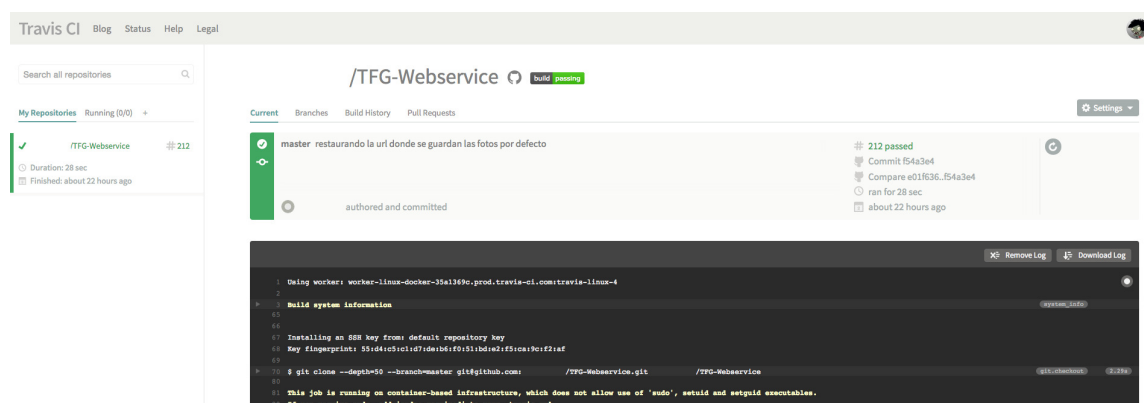


Figura 4.2: TravisCI

## 4.4. Entornos de desarrollo

### 4.4.1. Servidor

Cuando se va a desarrollar un interfaz de programación de aplicación o Application Programming Interface (API) hay varios factores a tener en cuenta, estos son: el lenguaje de programación, el sistema operativo del servidor sobre el que se ejecutará y la base de datos que se usará entre otros múltiples factores. En este apartado nos centraremos en el Integrated Development Environment (IDE) elegido para desarrollo de código del servidor, el framework seleccionado y el sistema gestor de base de datos.

#### 4.4.1.1. IDE: Eclipse™

Un entorno de desarrollo integrado (IDE) ofrece al programador un espacio de trabajo amigable, que además en la mayoría de los casos reconoce la sintaxis del código en tiempo real e informan al usuario de errores compilación. Esto descrito anteriormente es lo mínimo que se espera de un IDE, pero además es muy de agradecer que el IDE sea multiplataforma, tenga soporte para diversos lenguajes, integración con sistemas de control de versiones, reconocimiento de la sintaxis usada por cada lenguaje y facilidad de integración con los frameworks más populares de cada lenguaje. Eclipse™ es uno de los IDE mas usados a nivel mundial [27], no solo por su buena integración con Java™, si no por ser de código abierto y permitir a la comunidad crear complementos para interactuar con otros lenguajes. Además cumple con todos los requisitos descritos anteriormente. Por ese motivo se ha optado por Eclipse™ para trabajar con Python™ y su complemento PyDev™.

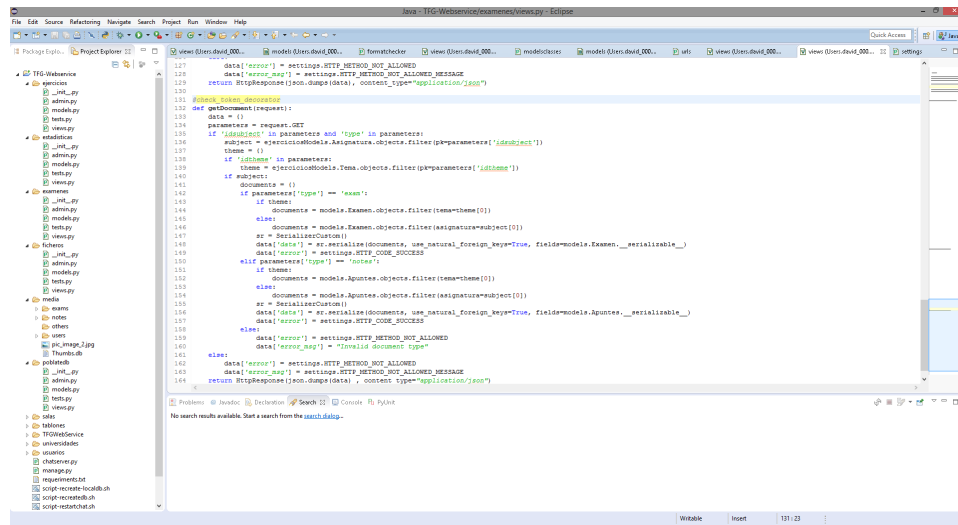


Figura 4.3: Eclipse™ con el complemento de PyDev™

#### 4.4.1.2. Framework: Django™

Un framework tiene el objetivo resolver un nuevo problema basándose en un conjunto de conceptos estandarizados, prácticas y criterios que han servido a la hora de resolver un problema similar en el pasado. Esto llevado al campo de la programación consiste en ayudar al programador a crear un nuevo software basándose en una estructura general ya definida para proyectos de la misma índole.

Para el caso de un API, lo que sería deseable de un framework es que facilite el trabajo con la comunicación HTTP, abstracción de la conexión a bases de datos, un ORM (Mapeado objeto-relacional) que sea fácil de utilizar, serialización y deserialización de objetos por mencionar algunas de las principales características.

Django™ es un framework que cumple con estas características además de tener una buena integración con Eclipse™ y su complemento PyDev™.

#### 4.4.1.3. Base de datos: PostgreSQL®

La base de datos es el corazón de un API, puesto que contiene la información con la que interactuar. En este caso se ha optado por una base de datos SQL, relacional, por el hecho de que los objetos que se van a almacenar en ella están relacionados entre sí, por ejemplo: un usuario está relacionado con los mensajes que ha escrito.

En un gestor de base de datos se valora muy positivamente que sea cumpla con las características ACID, esto quiere decir que conste de atomicidad, consistencia, aislamiento y durabilidad. Además de tener librerías que im-

plementen la conexión a estos gestores para un abanico amplio de lenguajes. Si a esto le sumamos que sea Open Source y además gratuito el resultado es PostgreSQL® como gestor de base de datos elegido para el proyecto.

#### 4.4.2. Android™

Android™ dispone de unas potentes herramientas que pone a disposición de los desarrolladores de manera gratuita, junto a una extensa documentación que describe todos los métodos, e incluye códigos “precocinados” listos para ser incluidos una aplicación.

En este caso, se ha utilizado el entorno de desarrollo integrado de JetBrains denominado Android Studio el cual cubre todas las fases del desarrollo de la aplicación, desde el programación, pruebas, hasta generación de una aplicación desplegable firmada y lista para ser publicada en la tienda de Aplicaciones de Google™ (Google Play).

Aunque Android Studio proporciona herramientas para la emulación de un dispositivo, el equipo de desarrollo prefirió una experiencia de desarrollo más nativa utilizando un dispositivo real, dado que otorga mayor fidelidad de los resultados a la hora de realizar pruebas. El dispositivo de desarrollo utilizado ha sido un “Google™ Nexus™ 4”, teléfono que tiene unas características muy interesantes y que posee la última versión de Android™, que a la fecha la escritura de esta memoria se trata de Android™ 5.1, lo cual permite obtener todas las funcionalidades que nos ofrece Android™ actualmente.

##### 4.4.2.1. Características especiales de la plataforma

###### Base de Datos Local

Para evitar una redundancia de las peticiones, se almacenan en una base de datos local tipo SQLite parte de la información que se obtiene a través del API. En concreto, se persiste en local información de:

- Mensajes de Tablón
- Usuarios
- Mensajes de Chat

En concreto para los mensajes de chat (figura 4.4), se utiliza de una manera muy especial. Hay un servicio corriendo que está escuchando mensajes de chat del socket. Una vez recibe uno, no se procesa en la interfaz de usuario, a no ser que el usuario esté con la sala de chat abierta en el momento de recibir el mensaje (Android™ no permite la modificación de una vista que no está siendo mostrada) sino que se persiste en la base de datos local.

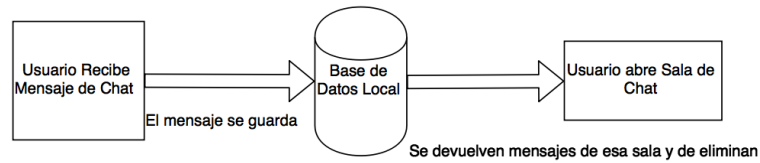


Figura 4.4: Android: Servicio de Chat

Una vez el usuario accede a la correspondiente sala de chat, se realiza una consulta en la base de datos para obtener mensajes correspondientes a esa sala.

Una vez se han obtenido, se borran de la base de datos local.

Al iniciar la aplicación todos los mensajes se borran, para evitar mensajes demasiado antiguos y aumentar innecesariamente el tamaño de la base de datos local.

#### SharedPreferences

Android proporciona métodos nativos para guardar información de la aplicación. En concreto, se ha utilizado una clase denominada `SharedPreferences`, que nos permite guardar cualquier tipo de objeto en la misma.

En concreto, cuando accedemos a la aplicación, se guardan en las preferencias los siguientes datos:

- Nuestro usuario
- Nuestra facultad
- Nuestro “token” de autenticación

Esto permite que, una vez conectado en la aplicación, no nos vuelva a preguntar por el usuario y contraseña sino que comprueba el valor de la autenticación y nos permite acceder si esta es válida.

#### 4.4.3. iOS™

iOS™, mencionado anteriormente, es propiedad de Apple®, cuyo ecosistema de desarrollo comparte las mismas herramientas, realizar aplicaciones para iOS™ y OS X™ gira entorno al entorno de desarrollo Xcode® (figura 4.5).

El entorno de desarrollo contiene todas las herramientas necesarias para llevar a cabo el desarrollo de un aplicación para iOS™ sin necesidad de recurrir a utilidades externas, desde lo básico, gestor de ficheros, editor de código, documentación, depurador, hasta control de versiones, etc.

Para el desarrollo de esta aplicación también se ha dispuesto de un dispositivo, iPhone 3GS, para realizar las pruebas de la aplicación desarrollada en hardware real. Bien sin olvidar que el dispositivo es antiguo dentro de la

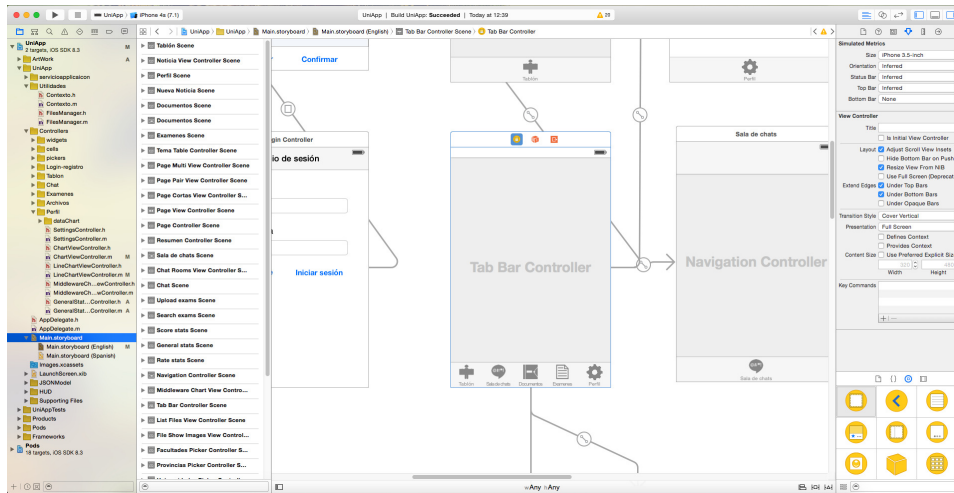


Figura 4.5: Xcode® iOS™ IDE

gama de teléfonos de Apple® y está dos versiones atrás en cuanto al sistema operativo actual y requerido para publicar en la App Store.

#### 4.4.3.1. Características especiales de la plataforma

A continuación se van a exponer las características principales que marcan la diferencia a la hora de desarrollar en esta plataforma (iOS™).

##### Modelo Vista Controlador

Es un patrón muy utilizado en desarrollo de software cuando intervienen interfaces de usuario. Este patrón permite realizar abstracción entre el comportamiento y apariencia de la vista con el modelo o datos que está usará, no queda aquí la definición del patrón MVC (diagrama 4.6), sino que también define cómo y con quién se van a comunicar estos tres elementos a la hora de interactuar entre sí.

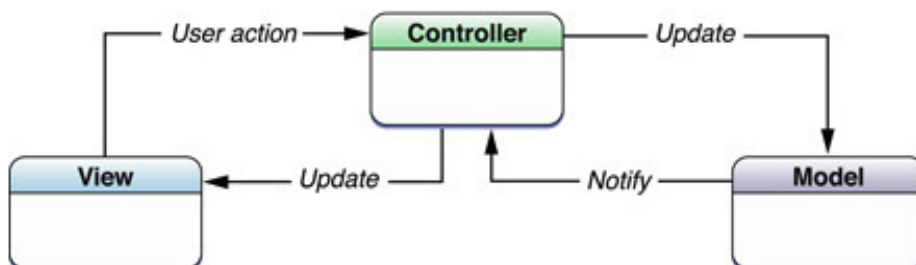


Figura 4.6: Diagrama MVC en iOS™

## Storyboard

Storyboard (figura 4.7) es una herramienta dentro de Xcode® que permite crear las interfaces de usuario colocando los elementos de manera visual, también permite interconectar diferentes vistas para definir el flujo de la aplicación, e incluso propiedades y datos para estos elementos que de otra forma se debería hacer programando directamente.

Por supuesto, esto facilita en gran medida crear interfaces de usuario con cierto comportamiento. Aunque por otro lado existe, como se ha mencionado antes, la forma de conectar los elementos visuales a los datos de la aplicación se realiza de una forma fácil pero que no queda reflejada en el código y puede llegar a hacer difícil encontrar errores y/o ver clara la conexión entre ciertos elementos.

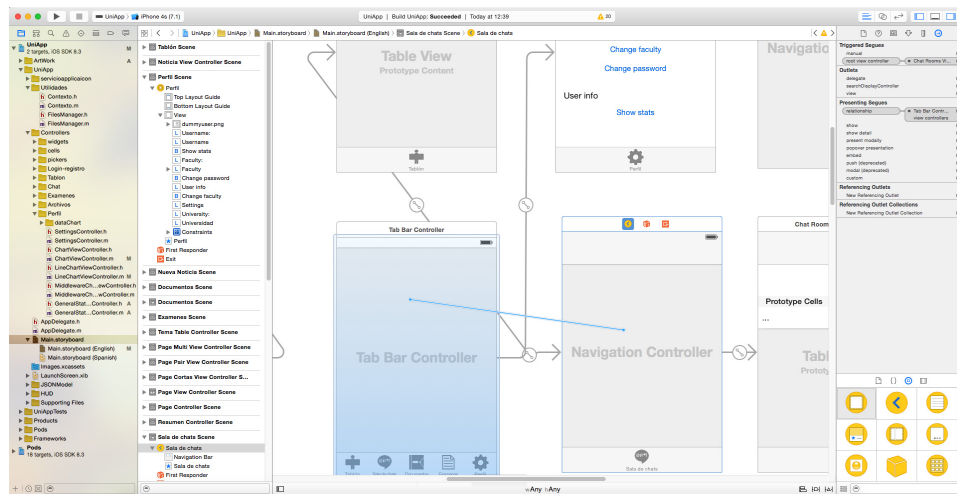


Figura 4.7: StoryBoard en iOS™

### Autolayout

Esta es una característica realmente especial de iOS™ por la cual los elementos de la interfaz responden adecuadamente adaptándose a cambios de orientación y tamaño de la pantalla del dispositivo.

Funciona a base de restricciones matemáticas que se pueden definir a un elemento en relación a la vista en la que está y/o al resto de elementos.

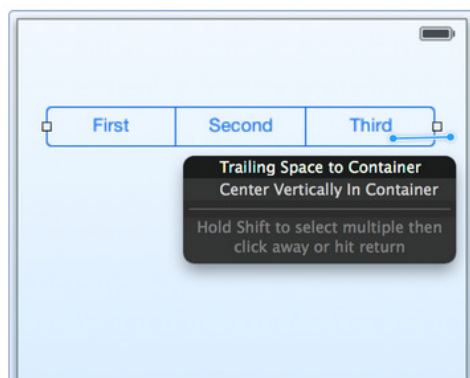


Figura 4.8: Autolayout

En la figura 4.8 se aprecia como añadir una restricción de posición al elemento en la vista que se está editando.

### Almacenamiento interno

En iOS existen varias formas de almacenamiento en el dispositivo móvil, estas son: espacio de usuario para almacenar ficheros, user defaults (almacenamiento de propiedades al estilo contenedor asociativo, clave-valor) y core data (ORM - Object-Relational Mapping para SQLite).

### User defaults

Este primer sistema permite almacenar las preferencias de la aplicación en términos de elementos clave-valor (clave1="valor2", etc). De esta manera se mantiene un sistema simple de persistencia para datos sencillos y sin relaciones entre sí de referencia directa ni de integridad.

En la figura 4.9 se puede apreciar como se registra a la propiedad Cache-DataAgressively con un valor booleano YES.

Este tipo de almacenamiento sobrevive al cierre de la aplicación, apagado del dispositivo, etc.

### Core data

Es un framework para el acceso a base de datos utilizando persistencia de objetos (diagrama 4.10), esto permite definir por ejemplo objetos con relaciones entre sí, listas de elementos, etc... que se pueden almacenar directamente en la base de datos sin necesidad de acceder a la información del

```

- (NSDictionary*)getUserPreferencesByArrayKeys:(NSArray*)keys {
    NSUserDefaults *preferencias = [NSUserDefaults standardUserDefaults];
    NSMutableDictionary* dic = [[NSMutableDictionary alloc] init];
    for (NSString* key in keys) {
        NSString* prop = [preferencias objectForKey:key];
        if (prop) {
            [dic setObject:prop forKey:key];
        }
        else {
            NSLog(@"Key no encontrada %@", key);
        }
    }
    return dic;
}

```

Figura 4.9: User defaults

objeto ni construir un sentencia de SQL.

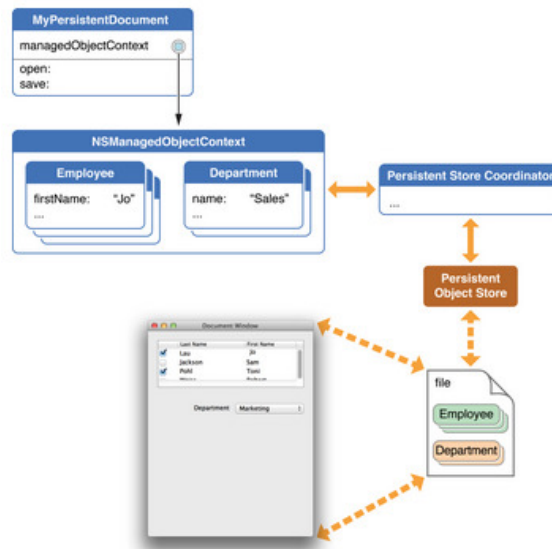


Figura 4.10: Core Data

Por supuesto la mayor este sistema de almacenamiento, al igual que user defaults, es persistente.

### Paso de información entre vistas (delegados)

En este caso se puede hacer de varias formas, o instanciando vía programación el controlador de la vista que se va a mostrar y pasando los datos explícitamente, o usando el patrón delegado.

El patrón delegado (diagrama 4.11) permite que un objeto actúe de com-

portamiento de otro o el proveedor de este.

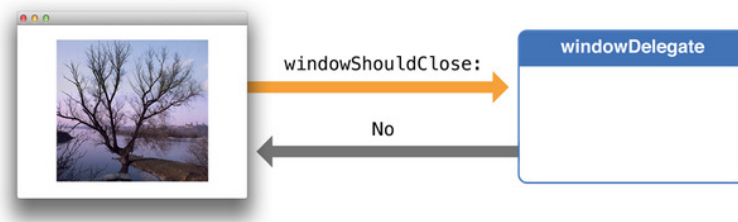


Figura 4.11: Patrón delegado

Por lo tanto es tan simple como definir un objeto delegado que contenga la información de la vista que se va a crear y que esta lo use para obtenerla. Al final la ventana creada tiene una referencia a la ventana padre que la invoca o al objeto delegado que se asigne.

#### 4.4.4. Windows Phone®

Windows Phone® (WP) es propiedad de Microsoft®, por este motivo todo el entorno de desarrollo gira alrededor del sistema operativo para computadora Windows®. En este caso la aplicación a desarrollar es para WP 8.1 y esto pide el requisito mínimo de tener instalado en el PC Windows® 8.1 [28].

El IDE (Integrated Development Environment) recomendado por Microsoft® para realizar aplicaciones para Windows Phone® es Microsoft Visual Studio® [29], del cual Microsoft® también es propietario.4.12

La aplicación a desarrollar es una aplicación para móviles, por ese motivo aparte de necesitar una computadora sobre el que desarrollar la aplicación se necesita un terminal que tenga el sistema operativo Windows Phone® 8.1. Aunque el IDE Visual Studio® incorpora una máquina virtual que simula ser un móvil WP para que puedas ejecutar tus aplicaciones, el equipo de desarrollo optó por la compra de un terminal Windows Phone® para tener un dispositivo real en el que ejecutar la aplicación.

Debido al bajo presupuesto del que se dispone se ha optado por comprar un terminal de gama baja, el Nokia Lumia 520 (figura 4.13). Este modelo de terminal será sobre el que se ejecute la aplicación y se harán las pruebas.

##### 4.4.4.1. Características especiales de la plataforma

En este apartado se procederá a exponer las principales diferencias de Windows Phone® con las otras plataformas sobre las que se han creado la aplicación.

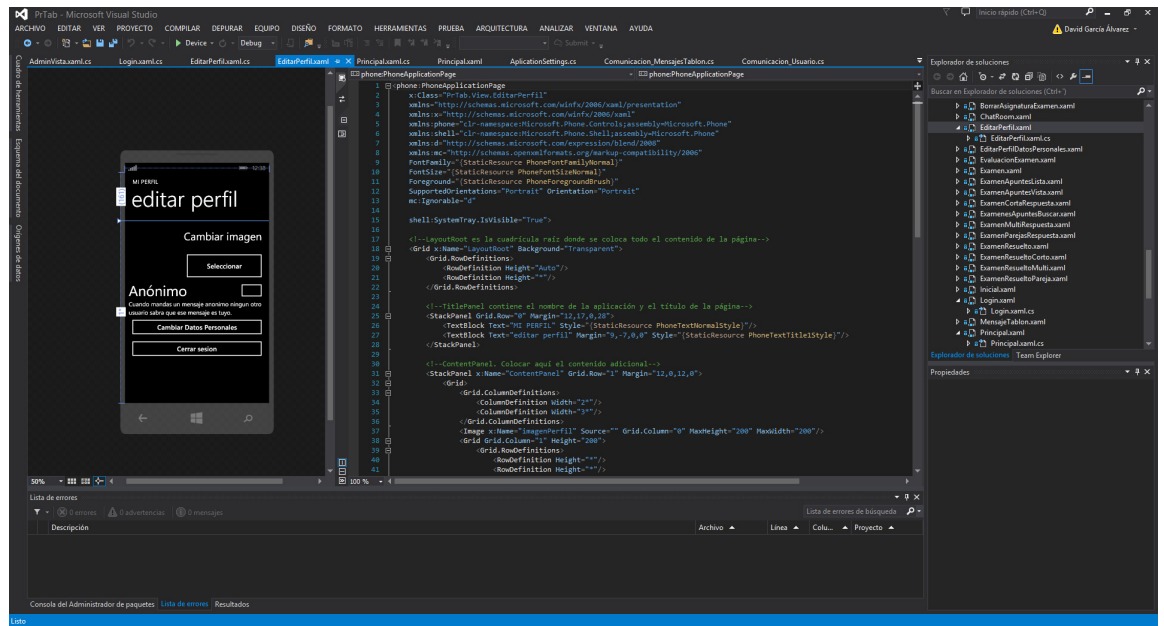


Figura 4.12: Visual Studio



Figura 4.13: Nokia Lumia 520

### Model-View-ViewModel

A lo largo del desarrollo de la aplicación se han usado muchos patrones de diseño puesto que como es bien conocido ayudan al desarrollo de la aplicación, la escalabilidad y a su posterior mantenimiento [30]. El objetivo de

este apartado no es hacer hincapié en estos patrones bien conocidos como: Factoría abstracta, Singleton, Adapter o Decorator, puesto que estos patrones se pueden usar en la mayoría de los lenguajes de programación. Lo que se va a describir en este apartado es un patrón exclusivo de Microsoft<sup>TM</sup> para desarrollo de aplicaciones para Windows® y Windows Phone®: Model-View-ViewModel.

Model-View-ViewModel (también denominado MVVM) es un patrón de diseño de aplicaciones para desacoplar código de interfaz de usuario y código que no pertenezca a dicha interfaz. En funcionamiento de este patrón es bastante sencillo, se define la interfaz de usuario mediante XAML y usas el marcado de enlace de datos para vincularla a otras capas que contengan datos. Esta infraestructura de enlace de datos proporciona un acoplamiento débil que mantiene sincronizada la interfaz del usuario con los datos a los que está vinculado.

Usando este patrón el código queda estructurado de tal forma que es posible cambiar cualquiera de las partes de forma individual sin que el resto de las partes se vea afectada. Esto nos ofrece varias ventajas como:

- Permitir un estilo de codificación exploratorio e iterativo.
- Simplifica las pruebas unitarias.
- Permite aprovechar mejor algunas herramientas como Expression Blend.
- Admite la colaboración en equipos.

Como dice el nombre del patrón, al utilizarlo la aplicación se divide en tres capas (diagrama 4.14):

- **Capa del modelo** (Model) incluye todo el código que implementa la lógica principal de la aplicación y define los tipos requeridos para modelar el dominio de la aplicación.
- **Capa de la vista** (View) define la interfaz de usuario en un lenguaje declarativo, en este caso XAML. El marcado de enlace de datos define la conexión entre componentes de la interfaz y diversos miembros de la vista.
- **Capa de modelo de vista** (ViewModel) proporciona destinos de enlace de datos para la vista. En la mayoría de los casos el ViewModel expone a la View el modelo o miembros encapsulados del modelo. El ViewModel también es el encargado de definir un seguimiento de datos que son relevantes para la interfaz de usuario pero no para el modelo, como por ejemplo el orden en que aparecen estos datos en la interfaz.

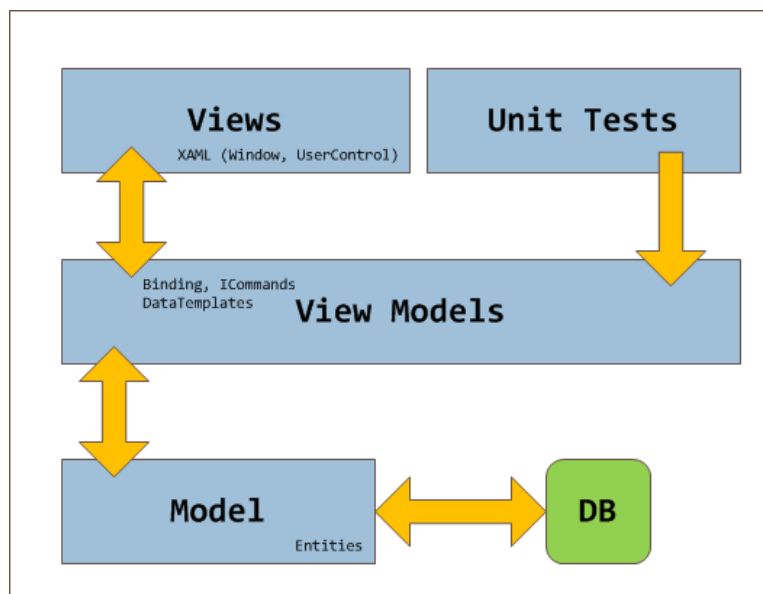


Figura 4.14: Capas del MVVM

Esta división en capas ofrece la ventaja de facilitar la comprensión del código, esta ventaja se viene dada a que el código de características específicas a menudo es independiente del otro código, lo que facilita su uso y reutilización. Otra ventaja que nos ofrece esta separación es la realización de pruebas unitarias automatizadas del código que no es parte de la interfaz de usuario.

Como conclusión podemos ver que esta arquitectura desacoplada permite aislar el impacto de cambios que se realicen en el código y su reutilización, lo cual es una característica muy importante a tener en cuenta a la hora de desarrollar un proyecto software.

### ValueConverter

En el apartado anterior se vio como un Model mediante un ViewModel queda vinculado a una View, es decir el ViewModel le pasa los datos del Model a la View para que este los muestre. En el caso más sencillo, por ejemplo una cadena de caracteres, solo tendría que mostrar en un cuadro de texto esa cadena de caracteres, pero podemos encontrarnos con un problema si, por ejemplo, el color de texto de un párrafo depende de un valor booleano (verdadero o falso), puesto que esta conversión no será directa.

Para estas situaciones tenemos los ValueConverter, se vincularía este valor booleano al color de letra del texto y se le diría que tiene que pasar por el ValueConverter que se ha definido, el cual transformará el valor booleano a un color. Para crear un ValueConverter solo hay que crearse una clase que implemente la interfaz IValueConverter. Esta interfaz obliga a implementar

dos funciones `Convert(...)` y `ConvertBack(...)`. En `Convert` se recibe el parámetro que se quiere convertir y se devuelve en el formato que se quería, `ConvertBack` ocurre lo contrario.

En el proyecto esto se ha utilizado por ejemplo para saber si un usuario le dio favorito a un mensaje y que aparezca el corazón pintado o no y transformar fechas en formato Unix Timestamp a un formato más legible por el ser humano. Esto se muestra de forma esquemática en la 4.15.

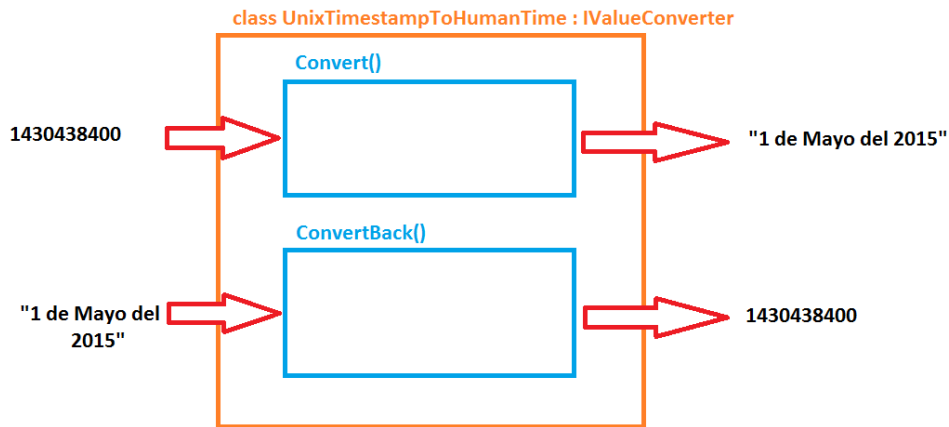


Figura 4.15: Ejemplo de `IValueConverter`

### Almacenamiento interno

Existen diversas formas de almacenar de forma interna en la aplicación datos e información. Para este proyecto se ha optado por 2 formas distintas almacenar la información, según el tipo de datos.

#### IsolatedStorage

Es una herramienta que viene por defecto instalada en el SDK de Windows Phone®, permite copiar, mover o reemplazar archivos y directorios a una carpeta local y exclusiva de tu aplicación. Esto se divide en tres apartados como se ve en el diagrama 4.16.

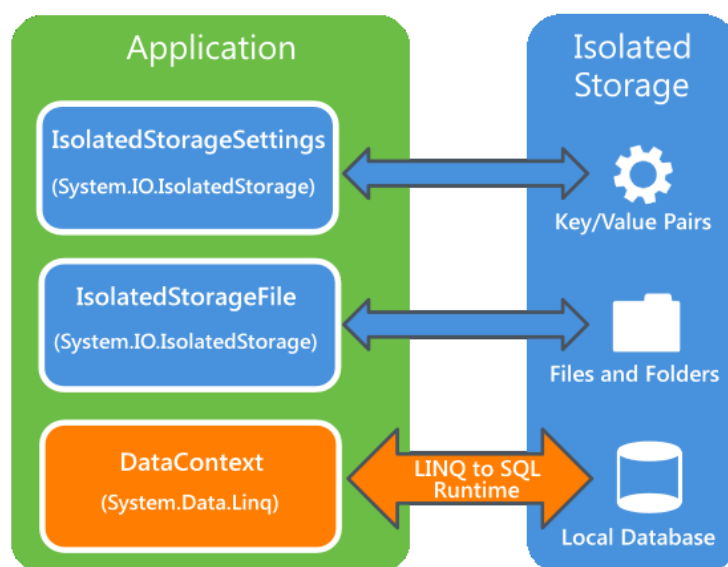


Figura 4.16: Descripción gráfica del IsolatedStorage

#### Parejas Clave/Valor

Es una forma de almacenar información en formato clave valor. La información solo se puede guardar como cadena de caracteres. Muy útil y sencilla de utilizar, pero tiene una gran limitación, puesto que solo se podrán guardar cadenas de caracteres u objetos serializados en cadena de caracteres. Este tipo de almacenamiento se usa en la aplicación para guardar información con tipo de dato sencillo (cadena de caracteres, enteros, decimales, booleanos) que puede ser necesaria de acceder desde cualquier punto de la aplicación. Por ejemplo el token de sesión, el ID del usuario registrado, el ID de la facultad del usuario, si los mensajes de tablón se ponen de forma anónima, etc. Para acceder a esta información descrita anteriormente se usa una clase

estática que se llama `ApplicationSettings` la cual tiene las funciones GET y SET para acceder o modificar esta información. A continuación en el diagrama 4.17, para guardar y coger el token.

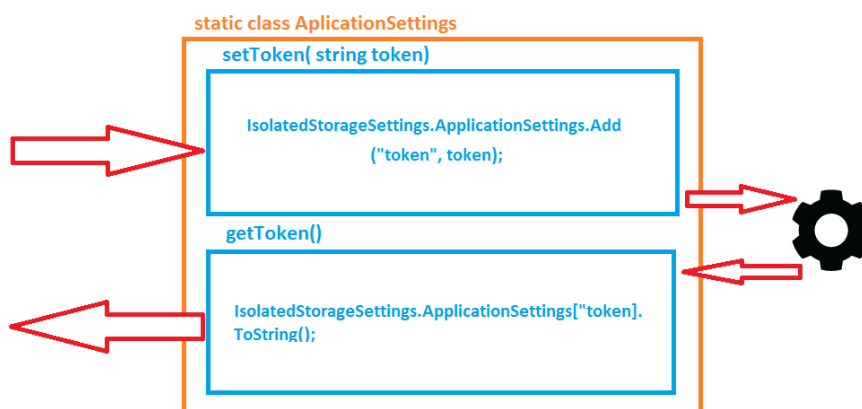


Figura 4.17: Ejemplo de `ApplicationSettings`

### Archivos y carpetas.

Este apartado es el que te permite crear carpetas dentro de la aplicación y archivos. A estas carpetas y archivos solo tendrá acceso la aplicación que las creó, de esta forma son privadas de la aplicación. Para esta aplicación esto se usará para guardar las fotos de los usuarios, de esta forma no será necesario descargarlas del servidor siempre que quieran mostrarse. Cada vez que se quiere mostrar la foto de un usuario pueden suceder 3 cosas (diagrama 4.18):

- La foto no está. La foto no se encuentra descargada en el terminal y hay que descargarla y almacenarla. A parte de guardar la foto se guarda la fecha de última modificación que te da el servidor en la cabecera de la petición HTTP.
- La foto está y es la actual. La foto se encuentra ya descargada en la aplicación y es la actual. Para saber si la foto es la actual se manda un mensaje HTTP HEAD al servidor, el cual te responde con varios parámetros (sin incluir la imagen que es la parte más pesada de la comunicación), entre esos parámetros está la fecha de última modificación. Si la fecha de última modificación que se tenía almacenada de la foto coincide con la respuesta del servidor significa que tenemos la foto actualizada.
- La foto está y no es la actual. Como se describió en el apartado anterior

se pregunta al servidor por la fecha de última modificación. Si esto no coincide se procede a descargar la nueva foto.

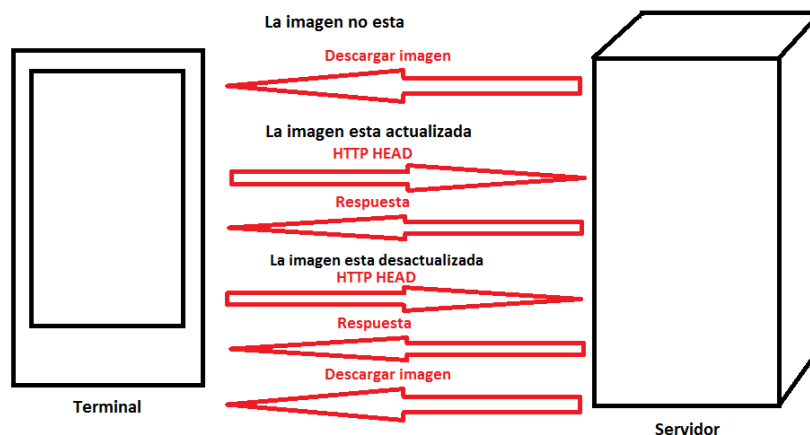


Figura 4.18: Explicación de la caché de imágenes y su interacción con el servidor

### Base de datos.

En el apartado anterior vimos como almacenar datos sencillos en formato cadena de caracteres de forma permanente en nuestra aplicación. No siempre queremos almacenar datos sencillos, en ciertas ocasiones queremos guardar objetos y esto no lo permite la clase `IsolatedStorageSettings`. Para estas ocasiones se necesita una base de datos interna en el terminal. Para Windows Phone® hay dos tipos de bases de datos diferentes disponibles: SQL Server CE® y SQLite.

Ambas tienen la misma característica a la hora de la creación, esta característica es denominada con 'code first', es decir 'código primero'. Esto consiste en que las tablas no se crean con un asistente o con consultas SQL, sino con las clases que ya están creadas en nuestra aplicación.

Las principales diferencias entre SQL Server CE® y SQLite las podemos encontrar resumidas en la tabla 4.19 que se encuentra debajo.

Se puede ver que SQLite tiene más ventajas que SQL Server CE® y las que no tiene, como por ejemplo funcionar en Windows Phone® 7.1 o encriptación de Base de Datos, no son algo demasiado importante para el desarrollo de esta aplicación.

En cambio a SQL Server CE® sí que le faltan algunas características muy interesantes para el proyecto, por ejemplo el código que se cree para

	SQL Server CE	SQLite
Windows Phone 7.1	Sí	No
Windows Phone 8	Sí	Sí
Aplicaciones 'Metro' de Windows 8	No	Sí
Sintaxis de LINQ	Sí	Sí
Ejecución de sentencias SQL	No	Sí
Multi plataforma	No	Sí
Soporte de Vistas	Sí	Sí
Soporte de Claves Foráneas	Sí	Sí
Soporte de Procedimientos Almacenados	No	No
Soporte de Triggers	No	Sí
Encriptación de Base de Datos	Sí	No

Figura 4.19: Características almacenamiento interno WP

Windows Phone® en relación a la base de datos no valdría para la versión de Windows® para escritorio, además SQL Server CE® no admite consultas SQL solo consultas mediante LINQ. A esto anterior se le suma que SQLite es multiplataforma (una base de datos en SQLite podrá ser usada tanto en Android, iOS y WP), la ventaja principal que tiene esto es que la comunidad de desarrolladores con SQLite es mucho más grande y hay más documentación ante posibles problemas.

Por los motivos descritos a la hora de elegir una base de datos para almacenar información se optó por implementar una base de datos interna SQLite.

Parte de la información que almacena la base de datos de la aplicación es información que tiene que pedirle al servidor y que difícilmente cambie en el servidor, como por ejemplo: las provincias, universidades, facultades, asignaturas por curso y temas de cada asignatura. Aun así hay mecanismos implementados para poder refrescar esta información por si cambia en el servidor, pero mientras no cambie se ahorran peticiones.

Otra parte de la información que se almacena en la base de datos sí que tiene más probabilidad de cambiar en el servidor o de añadir nueva información, por ese motivo esta información contiene una capa de control que se explicara en el apartado de casos de uso. Esta información de la que estamos hablando es: mensajes del tablón, caché de imágenes para las fotos de usuario y exámenes realizados.

**Paso de información entre vistas.** El paso de información entre vistas (también denominadas Pages) se hace mediante parámetros en la URL al invocar a la vista. Esto tiene una gran ventaja, su simplicidad puesto que si quieres pasar datos solo tienes que escribirlos en el siguiente formato:

*vistaDestino.xaml?dato1=info1&dato2=info2*

Donde *vistaDestino.xaml* es el nombre de la vista a la que se quiere invocar, *dato1* y *dato2* el nombre de los parámetros que se le pasan, e *info1* e *info2* los datos que se le pasan. La vista destino podrá recoger esta información sobrescribiendo el método *OnNavigatedTo* y recogiendo la información de la siguiente forma:

*NavigationContext.QueryString['dato1']*

El problema que tiene esta forma de pasarse datos es que solo permite pasar información en formato cadena de caracteres y esto tiene muchísimas limitaciones, puesto que nunca se podrán pasar objetos sin serializar.

## Capítulo 5

# Decisiones de diseño

En este capítulo se va a discutir las decisiones de diseño tomadas en el proyecto, para comunicar las aplicaciones móviles con el servidor, y la arquitectura interna de este.

### 5.1. Arquitecturas

Existen varias soluciones a nivel de arquitectura para proyectos de software, de comunicación, diseño de software con patrones. Exponiendo las elecciones tomadas.

#### 5.1.1. Cliente-servidor

Esta arquitectura se divide en dos partes fundamentalmente, el cliente es la parte activa de la comunicación y realiza las peticiones de información al servidor quedando a la espera de una respuesta. Normalmente este cliente es utilizado directamente por un usuario, por lo tanto contiene la interfaz de usuario.

Por el otro lado está el servidor, la división cliente servidor no siempre ni necesariamente debe ser a nivel físico, es decir, muchas veces se realiza por diseño o el servidor se puede estar ejecutando en la misma plataforma que el cliente; este es la parte pasiva en esta arquitectura (diagrama 5.1) ya que se dedica a esperar las peticiones de los clientes y responder con la información solicitada.

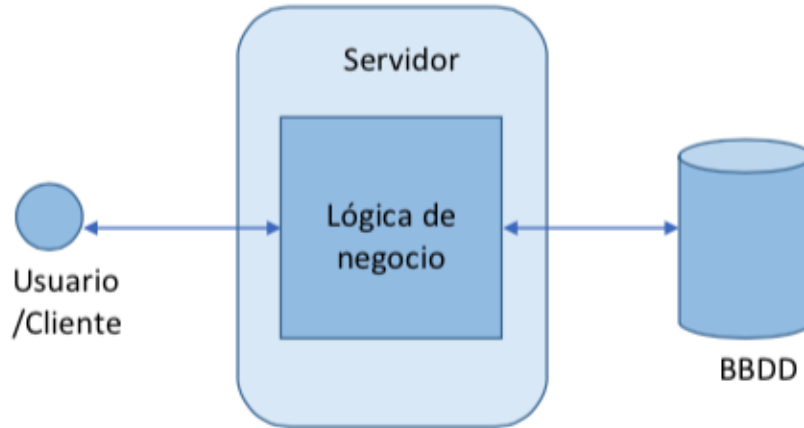


Figura 5.1: Cliente-servidor

### 5.1.2. Arquitectura en tres capas

Esta arquitectura es un derivado de la arquitectura cliente-servidor (diagrama 5.2) pero haciendo hincapié en una mayor diferenciación de las tareas o funcionalidades del servidor, separando la lógica de negocio del almacenamiento de datos.

Esto convierte el desarrollo bajo esta arquitectura en un desarrollo con mayor capacidad de extensibilidad, mantenibilidad y menor acoplamiento.

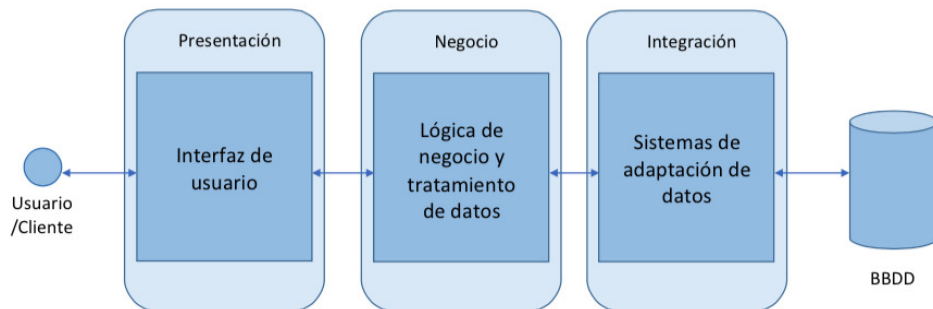


Figura 5.2: Arquitectura en tres capas

Capa de presentación: esta capa es la que interactúa con el usuario y solicita información al servidor.

Capa de negocio: esta capa se encarga de manejar los datos que el cliente envía y que se devuelven, a su vez gestiona las comprobaciones necesarias

para que los datos enviados al sistema de integración sean correctos.

Capa de datos/integración: es la capa que gestiona las particularidades del sistema de integración para la persistencia de los datos, ya sea el uso de ficheros, las consultas para la base de datos, etc.

### 5.1.3. Arquitectura orientada a servicios

Esta arquitectura está pensada para el desarrollo de sistemas distribuidos, permite que se construyan sistemas de información altamente escalables, flexibles y de fácil implantación. Los servidores pueden ser varios e incluso estar separados geográficamente entre si, repartiendo sus tareas bajo necesidad o diseño del proyecto.

En este caso servicio es entendido como una tarea a desempeñar sin estado, auto contenida, y bien definida en términos del interfaz utilizado para comunicarse con este.

Esta arquitectura suele utilizarse en sistemas de gran envergadura.

### 5.1.4. Elección de arquitectura

De las arquitecturas descritas anteriormente y obviando muchas otras que no se ajustan al tipo de proyecto desarrollado se ha optado por utilizar arquitectura en tres capas. Ello se debe a que facilita el desarrollo al separar los subsistemas de manera clara en el servidor, hace que el mantenimiento sea más sencillo, y se maximice la simplicidad de las tareas desarrolladas al solo encargarse de su competencia. Se ha descartado la arquitectura cliente-servidor por los motivos anteriores y la arquitectura orientada a servicios por sobrepasar en magnitud la capacidad de este proyecto y porque añadiría mayor complejidad al desarrollo, intrínsecamente el desarrollo de aplicaciones distribuidas es más complejo.

## 5.2. Application Programming Interface (API)

Dentro de la arquitectura multicapa de tres niveles se ha decidido construir las capas de negocio e integración a través de un interfaz de programación de aplicación (API). A diferencia de una arquitectura multicapa donde la comunicación desde la capa de presentación está restringida dentro del mismo sistema desarrollado, un diseño a través de un API permite desacoplar las capas de negocio e integración de la capa de presentación permitiendo de esta manera construir un sistema mucho más reutilizable, escalable y abstraído del cliente que solicita la información.

Por las razones descritas se ha decidido optar por diseñar el sistema del servidor como un API.

### 5.3. Sistemas de comunicación

Existen varios sistemas de comunicación que se pueden elegir para las respuestas proporcionadas por un API a los clientes, se van a tratar los más utilizados y conocidos para exponer su estructura y valorar el adecuado para escoger en este proyecto.

#### 5.3.1. XML

XML es un lenguaje de marcado muy extendido que se ha convertido en un estándar, desarrollado por el World Wide Web Consortium. Sirve para representar información de forma legible, comunicación entre sistemas de forma independiente del lenguaje o tecnología utilizados, e incluso da soporte para bases de datos.

Este lenguaje has sido y sigue siendo ampliamente utilizado, sobre todo en el entorno web. Por ejemplo se utiliza en sistemas SOAP en los cuales se definen como dos objetos de diferentes sistemas pueden comunicarse mediante XML.

Un desventaja que se puede encontrar en el uso de XML al ser independiente del sistema es que cada lenguaje debe utilizar un sistema de interpretación de este lenguaje para convertir la información en una estructura de datos manejable por el lenguaje utilizado.

#### 5.3.2. JSON

JSON es un formato de datos en representación literal de los objetos de JavaScript, estructura de datos también conocida como diccionario o contenedor asociativo. Una de las características de esta representación de los datos es la ligereza (en término de tamaño) que otorga el formato, es decir, para representar la información se necesitan menos caracteres que en lenguajes como XML.

Por otro lado la simplicidad de este formato de datos implica directamente en que el analizador que interpreta la información en una estructura JSON es más sencillo de construir.

De los sistemas expuestos arriba para este proyecto se ha decidido utilizar JSON, dado que minimiza los datos necesarios a enviar en la comunicación, siendo esto fundamental ya que ayuda notablemente a minimizar el consumo de datos en dispositivos móviles.

## 5.4. Estructura, diseño e implementación

En este apartado se expone la estructura y el diseño del servidor y de las aplicaciones móviles desarrolladas, incluyendo donde sea necesario descripciones sobre decisiones de implementación.

### 5.4.1. Servidor

Como ya se ha mencionado antes el servidor está diseñado bajo una arquitectura multicapa (más concretamente modelo-vista-controlador) y con un API como sistema de comunicación con los clientes. Toda la información es enviada y recibida en formato JSON a excepción del panel web, que se trata de HTML. A continuación se exponen aspectos de diseño e implementación que se han considerado relevantes.

#### Inicio de sesión

La primera decisión de diseño viene dada por incorporar el sistema en API al servidor, esta trata sobre el inicio de sesión de usuario. Ya que el diseño de un API proporciona una serie de consultas independientes entre si al servidor, se necesita garantizar que el cliente que las realiza lo hace siendo el usuario correcto y autorizado. Para ello se opta por un sistema de token, un identificador único que se obtienen del servidor al realizar la llamada de inicio de sesión y que es requerido para el resto de consultas.

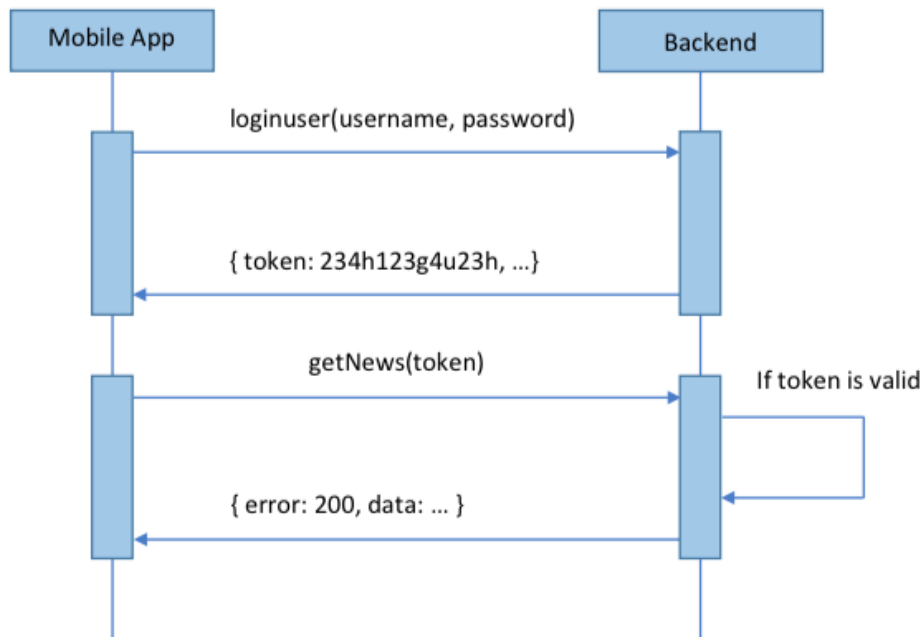


Figura 5.3: Inicio de sesión

En el diagrama 5.3 se muestra como el sistema inicio de sesión es seguro dado que usa tokens, mencionado anteriormente, si no son válidos o han caducado no se permitirá la comunicación con el backend, en caso contrario las peticiones de datos transcurren de forma esperada.

### Decoradores o capa de middleware

Dada la necesidad mencionada en el sistema de inicio de sesión de utilizar un token para la obtención de información se genera la necesidad de comprobar que el token de sesión o identificador único de sesión es válido para la consulta a realizar.

Todas las consultas que se pueden realizar al API requieren un token, por lo que la comprobación de dicho token debe realizarse siempre antes de la consulta. Para solucionar esto se ha decidido utilizar una característica del lenguaje Python que se llama decorador, esto permite resolver exactamente la necesidad expuesta, se trata de una función que se ejecutará siempre antes de las consultas al API comprobando la validez del token y permitiendo que se continúe con la consulta sólo en el caso de que lo sea (diagrama 5.3).

En el diagrama 5.4 se muestra como la capa de middleware funciona en el caso de las comprobaciones del token. Pero también nos permite registrar accesos al sistema para obtener estadísticas.

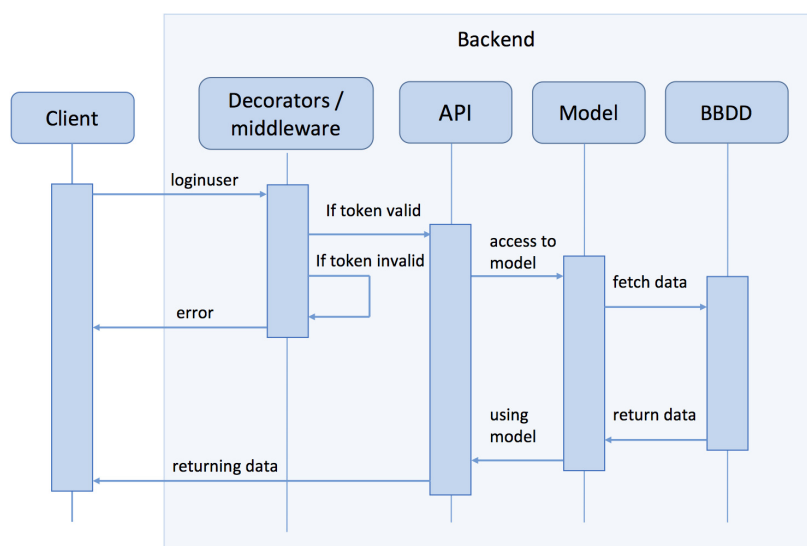


Figura 5.4: Inicio de sesión

### Arquitectura modelo-vista-controlador

El sistema desarrollado como API hace uso del framework Django el cual utiliza una arquitectura modelo-vista-controlador (MVC), esto permite separar la definición de los datos del sistema de la forma en la que se almacenan y del comportamiento de este (albergado en el controlador), también a su vez se abstrae de cómo se van a mostrar los datos en el cliente (vistas).

En este sistema en particular las vistas están desarrolladas en las aplicaciones móviles y el acceso en forma de API solo conlleva involucrar al controlador, modelo e integración. A excepción por su puesto del panel de administración web que sigue este modelo fielmente.

### ORM API

Otra de las características que incorpora Django es un API para abstraer el tratamiento de datos a nivel de base de datos, es decir, permite que no se necesite utilizar lenguajes como SQL para realizar las consultas y modificaciones de los datos en el sistema de integración (véase en el apéndice A sección ORM API en Django).

### 5.4.2. Aplicaciones móviles

Dado que cada aplicación móvil ha sido desarrollada en una plataforma diferente se da el caso donde todo el ecosistema cambia de una a otra, herramientas de desarrollo, lenguaje, plataforma, diseño de interfaces y sus elementos, etc. Por ello es que el desarrollo de cada aplicación ha tenido que adaptarse o respetar las particularidades de su plataforma.

#### **Interfaz de usuario**

Este es uno de los puntos más diferenciales entre las aplicaciones, dado que en cada sistema existen unos elementos de interfaz de usuario diferentes, distintos métodos para gestionar las vistas y el paso de información entre ellas.

Por lo tanto se ha optado por tomar la decisión de que cada aplicación móvil resuelva los diseños y problemas en base a los recursos particulares del sistema. Pero siempre manteniendo la misma funcionalidad.

#### **Consumo de datos**

Las aplicaciones desarrolladas, como se ha mencionado anteriormente en el capítulo dos y tres, tienen un carácter ubicuo; esto lleva a considerar que los recursos de los dispositivos móviles son limitados (datos y batería).

Teniendo en cuenta la limitación de recursos en el desarrollo del proyecto se ha buscado agrupar la información al máximo posible en la comunicación entre cliente y servidor para disminuir el consumo de ambos factores.

#### **Sistema de almacenamiento**

En el caso del sistema de almacenamiento del dispositivo móvil también se encuentran cambios significativos, tomando de ejemplo iOS no hace uso de sentencias SQL ya que tiene un sistema ORM llamado Core Data (consulte el apéndice A, la sección de almacenamiento interno para cada plataforma).

Por lo cual las técnicas de almacenamiento persistente en cada dispositivo móvil se han implementado de forma diferente tal y como cada sistema resuelve dicha situación.

## Capítulo 6

# Trabajo Realizado

En este capítulo se explicará el trabajo que se ha realizado. Se ha dividido en dos secciones : el servidor de aplicaciones (API) y los distintos clientes en aplicaciones móviles.

### 6.1. Servidor

Para gestionar el backend del sistema M-Learning utilizando el lenguaje de programación Python (v2.7) y el framework Django (v1.7).

Para lograr una mayor escalabilidad y evitar problemas de conexión en las redes universitarias o privadas se ha utilizado el protocolo HTTP para las conexiones con el backend, ya que otro tipo de protocolos pueden tener dificultades en este tipo de redes para operar con normalidad debido a las restricciones de seguridad.

#### 6.1.1. Fundamentos básicos del API desarrollado

La comunicación de la aplicación móvil se realiza mediante las peticiones HTTP comentadas anteriormente con el backend y este responde en formato JSON, este formato es muy ligero en consumo de datos, fácil de tratar y modelar, y que las tres aplicaciones móviles tratan perfectamente.

El API desarrollada tiene una serie de funciones definidas, las cuales realizan un trabajo expecificado para cumplir con los requisitos del sistema. Estas funciones reciben una serie de parámetros en formato HTTP GET o HTTP POST, según corresponda. Por ejemplo, si la llamada requiriere el envío de una fichero únicamente puede realizarse vía POST, y para esta llamada todos los parámetros se envían por la misma vía. En cambio si no se requiriese el envío de ningún archivo todos los parámetros de la llamada se incluirán en el HTTP GET.

El formato de una función con parametros HTTP GET es:

<http://www.bsodsoftware.me/funcion?parametro1=dato1&parametro2=dato2>

Los parámetros que recibe el API al realizar una llamada pueden ser de dos tipos: parámetros obligatorios o parámetros opcionales.

Antes de realizar cualquier acción, las funciones del servidor con parámetros obligatorios comprueban de su existencia. Se indicará que faltan parámetros obligatorios en la respuesta para ayudar a los desarrolladores y control de errores.

La mayoría de las funciones que se invocan tienen una dependencia del usuario que lo hace, por ese motivo incluyen un parámetro denominado “token” que es el identificador de sesión de un usuario. El “token” es una cadena alfanumérica aleatoria y única, lo obtiene el usuario al registrarse manteniéndolo hasta su próxima conexión con el sistema.

Como la comprobación del parámetro “token” es recurrente, ya que para la mayor parte de las invocaciones es necesario estar autenticado en el sistema para hacer uso de ellas, se decidió implementar un “decorador” (capítulo cinco para más detalles acerca de decoradores) para la comprobación de su existencia en las invocaciones a funciones.

Esto es una función que “envuelve” a otra, es decir que se ejecutara antes y podrá abortar la ejecución en caso necesario. En nuestro caso la función invocada no se puede ejecutar si falta el “token”. Además esto proporciona una gran cantidad de ventajas como se puede ver en [31].

En el diagrama 6.1 se puede observar la arquitectura del sistema M-Learning, donde el usuario que interactúa con la aplicación móvil crea una comunicación desde esta hacia el backend y el backend a su vez con la base de datos. Generando de esta manera un canal de comunicación bidireccional.

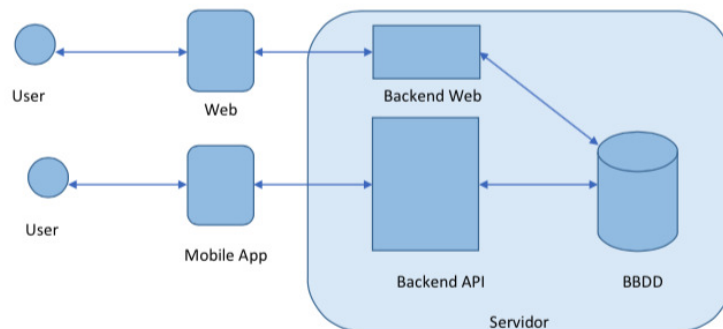


Figura 6.1: Estructura sistema

### 6.1.2. Sistema de Chat

Para la implementación del chat se ha optado por la programación orientada a eventos, en este caso los eventos son las nuevas conexiones TCP al puerto 8000 para el posterior envío/recepción de información.

Se ha hecho uso de Twisted, que es un motor de comunicación orientado a eventos de red escrito en Python [32].

El primer paso necesario para utilizar es chat se trata de autenticarse en el sistema.

Como para entrar al chat se necesitan credenciales, los usuarios podrán estar en dos estados distintos. El primero “conectándose” cuando su terminal se conecta al puerto del servidor y empieza a mandar las credenciales para poder acceder al chat. Y el segundo “chat” cuando las credenciales del usuario son aceptadas. Una vez se encuentra en el segundo estado ya esta listo para conectarse a salas, crearlas, mandar o recibir mensajes. A continuación se muestra un ejemplo de conexión en la figura 6.2.

La comunicación entre el servidor y el móvil del usuario está encapsulada en JSON y existen una serie de mensajes predefinidos, a continuación se mostrarán los mensajes del servidor (figura 6.3) en y de la aplicación móvil (figura 6.4).

Para el servidor el mensaje de tipo “status” informa al terminal en que estado se encuentra, si esta “conectándose” o esta ya en estado “chat”. El mensaje de tipo “message” lo manda el servidor cuando otro usuario ha mandado un mensaje a una determinada sala en la que el móvil está conectado. El último tipo de mensaje que puede mandar el servidor es de tipo “error” para notificar que se ha producido un error, como por ejemplo credenciales inválidas.

El terminal también puede mandar tres tipos de mensajes diferentes, el primero “login” para mandar las credenciales y así poder conectarse al chat. Con el segundo “message” se mandan mensajes al resto de usuarios de una determinada sala y con “room” se accede a una sala. En caso de que la sala a la que se quiere conectar no exista el servidor la crea.

### 6.1.3. Despliegue

Para facilitar el despliegue del API en el servidor de producción, se desarrollaron unos scripts en Shell Bash para producir una automatización del despliegue, actualizando el código desde repositorio de GitHub<sup>TM</sup> y actualizando el modelo de datos si fuere necesario.

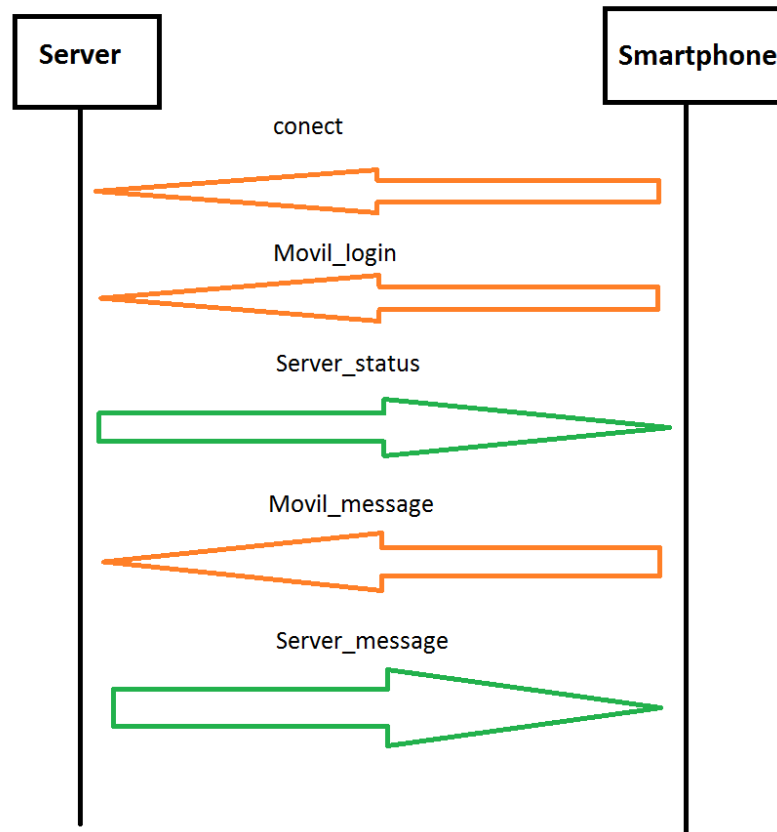


Figura 6.2: Conexión al Chat

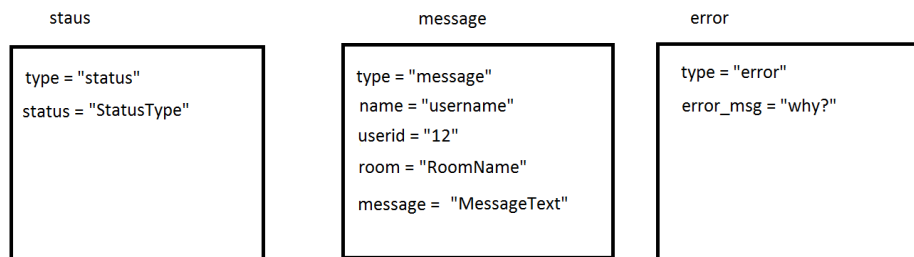


Figura 6.3: Mensajes enviados por el servidor

## 6.2. Aplicaciones móviles

En este apartado se explicará cómo se han afrontado los casos de uso en las tres plataformas sobre las que se ha desarrollado.

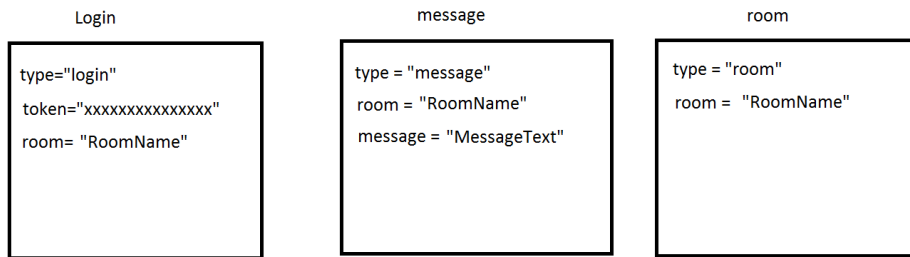


Figura 6.4: Mensajes enviados por los dispositivos

Se debe tener en cuenta que las tres plataformas sobre las que se ha desarrollado la aplicación tienen estilos distintos, por este motivo y aun que las tres cumplen con todos los requisitos del sistema, es bastante probable que no coincidan en la forma de implementarlos. En estos casos se hará una aclaración para cada plataforma si fuere necesario.

### 6.2.1. Primera ejecución

Hay dos casos de uso iniciales, Registrar e Iniciar Sesión. Por ese motivo la aplicación lo primero que hará nada más abrirla será consultar si hay un usuario ya registrado. En caso de que no se exista un usuario registrado hay que dar opción al usuario a elegir entre Iniciar Sesión o Registrarse si ya tiene una cuenta. En la figura 6.5 se muestra lo descrito anteriormente y una captura del registro.

### 6.2.2. UC1: Registrar Usuario

Para empezar se va suponer que el usuario acaba de instalar la aplicación, si este es el caso el siguiente paso será que pulse en el botón registrar y de esta forma la siguiente pantalla que verá será en 6.6. Una vez aquí el usuario deberá elegir su nombre de usuario con el cual se registrara en la aplicación. también tendrá que elegir una contraseña, correo electrónico, provincia, universidad y por último facultad.

Esto parece sencillo, pero tenemos que tener en cuenta que no pueden existir dos usuarios con el mismo nombre de usuario ni con el mismo correo electrónico, a esto hay que sumarle que por motivos de seguridad no son aceptadas las contraseñas cortas. Estos pequeños detalles se controlan y muestran en 6.6.

Para controlar que no se repita nombre de usuario lo que hace el terminal es hacer una llamada al servidor preguntando si ese nombre de usuario está

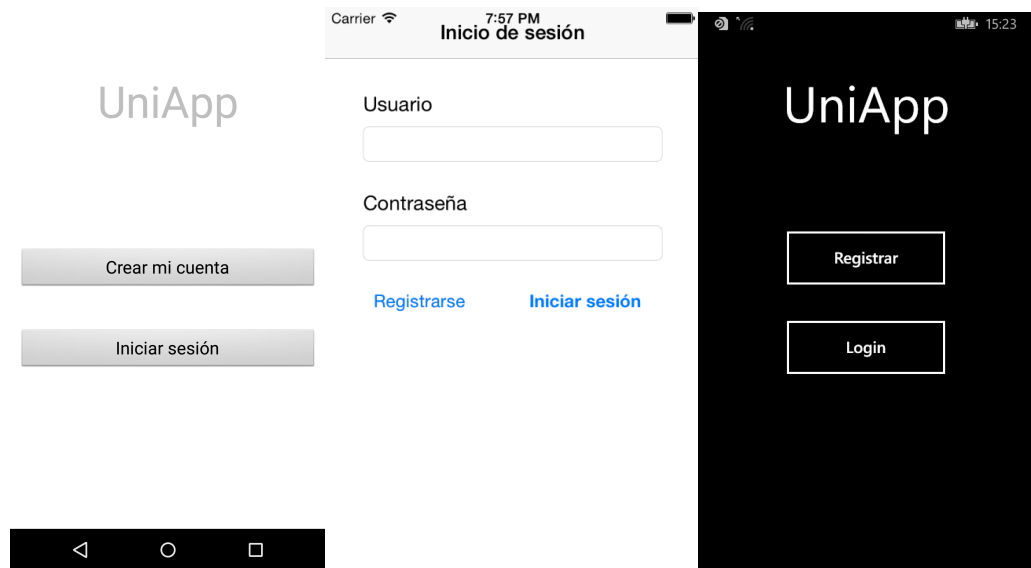


Figura 6.5: Pantalla de inicio Android™, iOS™ y Windows Phone®

disponible, en caso de que el nombre de usuario está en uso se le notifica con un mensaje y se marcará en rojo el campo.

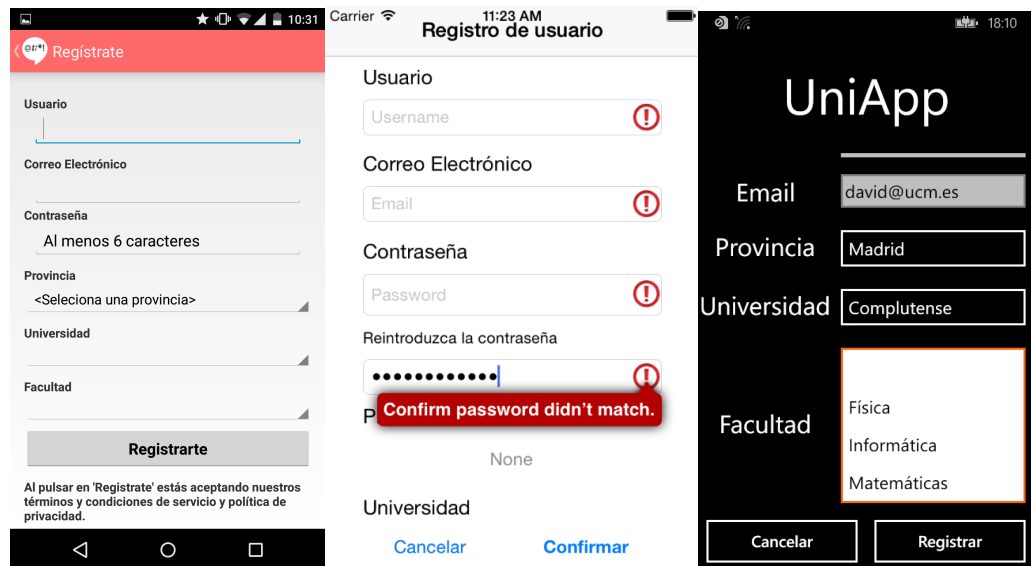


Figura 6.6: Proceso de registro en Android™, iOS™ y Windows Phone®

El control de la contraseña es bastante más sencillo, puesto que no requiere de una llamada al servidor, solamente se cuenta el número de caracteres y en caso de ser menor que seis se le notifica de forma similar al apartado

anterior.

El campo email se controla de forma muy similar al de nombre de usuario, se comprueba con una llamada al servidor si ese email no está siendo ya usado. A esto hay que añadirle que en el terminal se comprueba que lo que este escrito sea “xxx@xxx.xxx” donde las “xxx” son cadenas de caracteres.

Los campos que quedan del formulario son la comunidad, universidad y la facultad. El funcionamiento de estos tres campos es el siguiente: nada más entrar en esta vista el móvil le pregunta al servidor qué comunidades hay, cuando el usuario ha seleccionado una comunidad, el móvil pide al servidor las universidades de esa comunidad y por último cuando se ha seleccionado la universidad se obtienen del servidor las facultades que tiene dicha universidad.

Una vez completado el formulario solo hará falta pulsar el botón “Registrar” y se tendrá acceso al contenido de para usuarios registrados de la aplicación. En caso de que el usuario ya tuviera cuenta solo tendrá que hacer login y no seguir el proceso de registro.

### 6.2.3. UC2: Autenticación de usuario

Para el proceso de autenticación de usuario solo será necesario introducir los campos “Usuario” y “Contraseña” que aparecen en la vista, una vez listo se pulsa sobre el botón “Aceptar/Iniciar Sesión”. Si estas credenciales son correctas se accede a la vista principal de la aplicación donde se verá el tablón.

### 6.2.4. UC3: Obtener mensajes tablón

Una vez ya completado el proceso de registro o login, pasamos a estar en la vista principal, donde se puede ver los mensajes que han publicado los usuarios de la aplicación para el tablón de su facultad. Nada más cargar esta vista desde el móvil lanza una petición al servidor para que este le proporcione los mensajes nuevos. Mientras estos mensajes se cargan se le mostrará al usuario una notificación indicando que se están cargando, esto se muestra en 6.7.

Para volver a obtener los nuevos mensajes del tablón hay un botón con el símbolo “Refrescar”.

Por otro lado y de forma transparente para el usuario, en este proceso se obtienen cambios sobre los mensajes que ya disponemos (número de favoritos) tras obtener los mensajes nuevos.

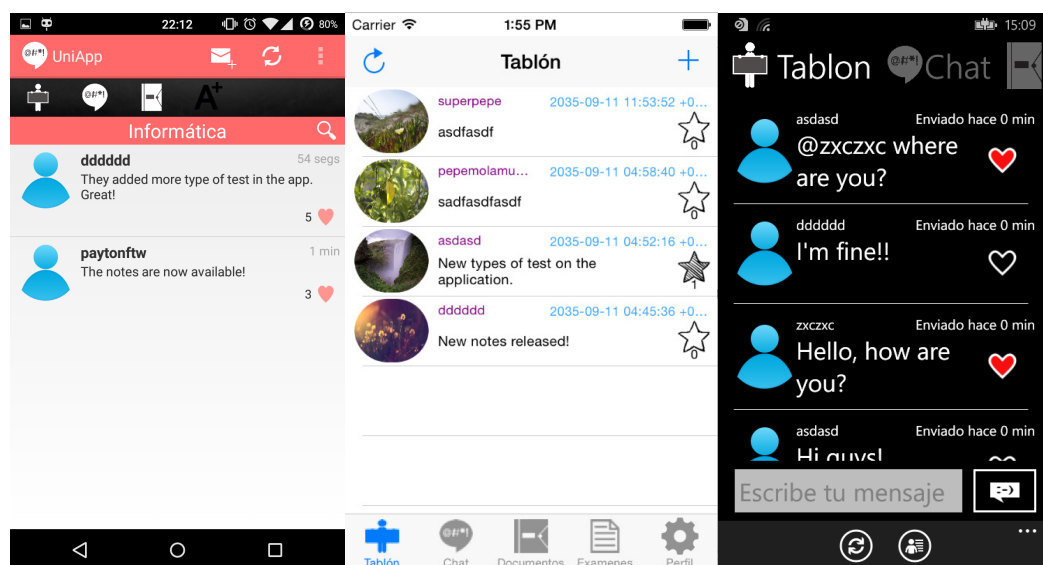


Figura 6.7: Tablón en Android™, iOS™ y Windows Phone®

#### 6.2.5. UC4: Enviar mensaje tablón

En el apartado anterior vimos cómo la aplicación carga los mensajes que los usuarios ponen en el tablón, en este apartado se muestra cómo los usuarios pueden escribir mensajes al tablón. La forma de escribir mensajes varía según la plataforma, Android™ e iOS™ tienen un botón, icono de un sobre con un más para Android e icono con un más para iOS™, que, al pulsarlo, permite escribir un mensaje, en cambio para Windows Phone® puedes hacerlo directamente escribiendo en el cuadro de texto que aparece abajo. Mientras el mensaje se envía al usuario se le mostrara una notificación que le informa de esta situación. Cuando esta operación se haya completado el nuevo mensaje aparecerá en el tablón. Esto se muestra en 6.8.

Además de poder publicar mensajes con el nombre del usuario también se puede enviar de forma anónima. En Android™ e iOS™ esto se pregunta siempre que se quiere mandar un mensaje, en cambio para Windows Phone® se tendrá que acceder al menú de configuración y seleccionar la opción de “Anónimo”. Mientras la opción está habilitada todos los mensajes que se publiquen desde ese terminal serán anónimos.

#### 6.2.6. UC5: Marcar mensaje favorito

Una de interacciones que se permite a los usuarios con los mensajes es agregar un mensaje a favoritos, esto se hace mediante los corazones en Android™ y Windows Phone® o estrellas en iOS™. Cuando un usuario agrega un mensaje a favorito el corazón/estrella se pone rojo/coloreado. Al

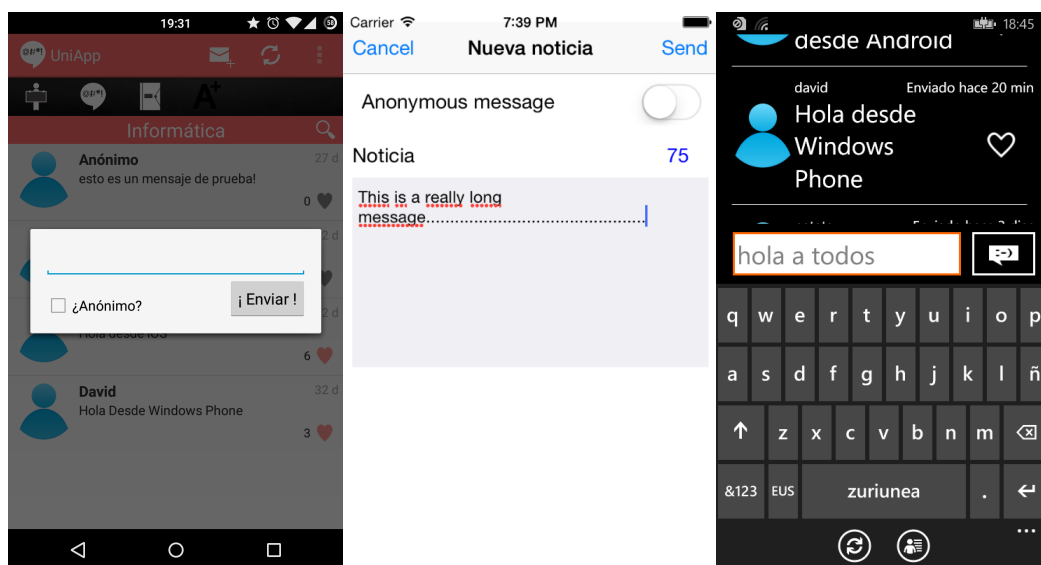


Figura 6.8: Escribir mensaje en Android™, iOS™ y Windows Phone®

pulsar el corazón/estrella el móvil manda una petición al servidor informando que el usuario ha agregado o desagregado un mensaje a favorito. También la aplicación permite ver cuántos usuarios han agregado a favoritos un mensaje. Un ejemplo de esto se ve en 6.7.

### 6.2.7. UC6: Borrar mensaje

Se ofrece a los usuarios la posibilidad de borrar mensajes que ellos hayan publicado. Un usuario solo podrá borrar mensajes que hayan sido escritos por él; además si publicó el mensaje de forma anónima habrá perdido la propiedad del mensaje y tampoco podrá borrarlo. Para borrar un mensaje solo hay que pinchar en él en la vista principal y cuando esté en la vista detalle del mensaje aparecerá una equis en Android™; un botón con el texto “Delete” en iOS™ o papelera en Windows Phone®. Esto se muestra en 6.9.

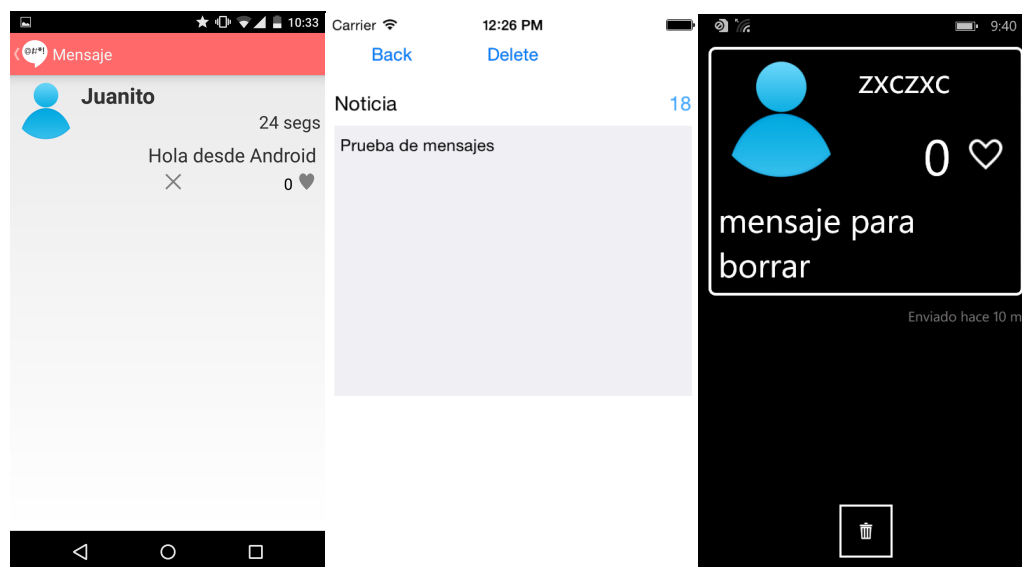


Figura 6.9: Borrar mensaje en Android™, iOS™ y Windows Phone®

### 6.2.8. UC7: Agregar/Borrar asignatura a favoritos

Para acceder a la opción de agregar/borrar asignatura a favoritos desde el menú principal se puede ver una pestaña con el icono de una carpeta en Android™, un folio con el texto “Exámenes” en iOS™ y el icono de una carpeta con el texto “Apuntes” en Windows Phone®, una vez ahí ya se tiene la posibilidad de añadir una asignatura a favoritos.

Primero se preguntará por el curso y posteriormente se muestra información 6.10 de las asignaturas que se disponen para ese curso. Para obtener las asignaturas de un curso, el terminal hace una llamada al servidor y es este el que le notifica que asignaturas dispone cada curso.

Se permite que un usuario elimine de favoritos una asignatura que ya tenía añadido a favoritos. Para ello tenemos la opción de borrarla que es idéntico al paso de agregar.

### 6.2.9. UC8: Realizar examen

Una vez en el apartado de examen se selecciona la asignatura sobre la que se desea realizar el exámen. Esto lleva a otra vista en la que se muestran los temas de dicha asignatura. Una vez en ella, se selecciona el tema sobre el que se quiere hacer examen o la opción de todos los temas 6.11. Una vez decidido el tema sobre el que se realizará el examen, en Android™ y Windows Phone® se dará la opción al usuario de elegir que tipo de examen quiere realizar, mientras que en iOS™ esto se hará de forma aleatoria. Una

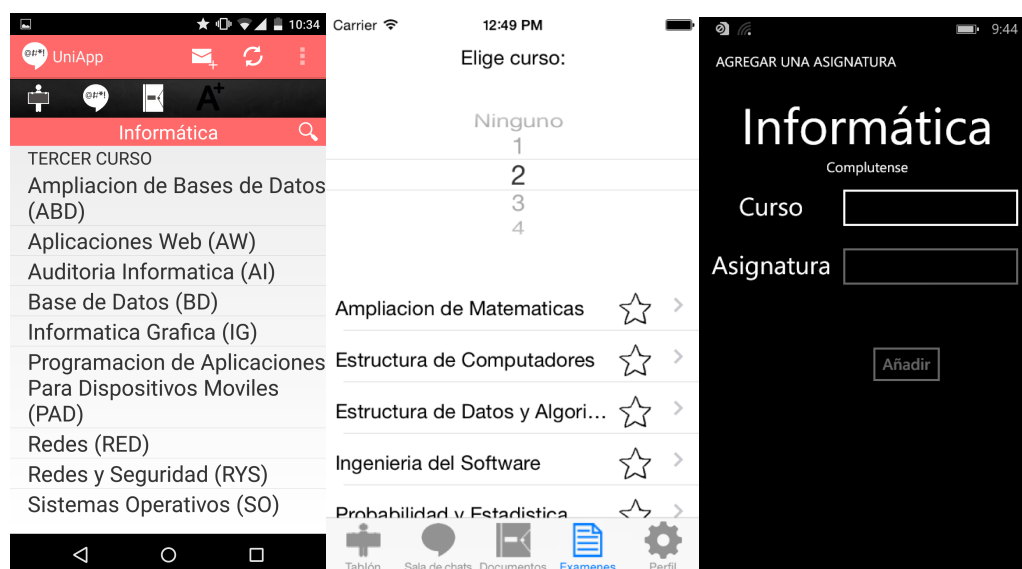


Figura 6.10: Agregar asignatura a favoritos en Android™, iOS™ y Windows Phone®

vez seleccionado el tipo de examen que se desea realizar el terminal solicitará al servidor un examen con esos parámetros y se mostrará la vista de examen.

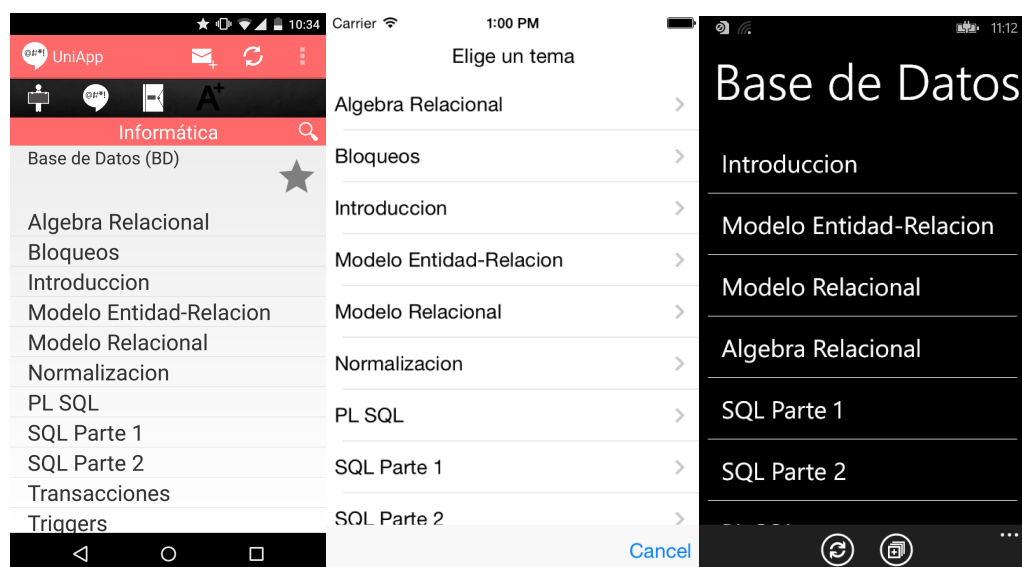


Figura 6.11: Selección de un tema para realizar un examen en Android™, iOS™ y Windows Phone®

Todos los exámenes, da igual su tipo, tendrán un formato unificado para

moverse entre las preguntas. El usuario podrá navegar entre las preguntas tantas veces desee y una vez considere que está finalizado podrá enviar el mismo.

Tras finalizar cada examen se mostrará la nota obtenida y se proporcionará la opción de salir o visualizar la corrección. Para visualizar la corrección, el usuario deberá navegar entre las preguntas y podrá comprobar las respuestas acertadas y falladas en distintos colores. El terminal enviará la información de preguntas acertadas, no respondidas o falladas.

A continuación se expondrán ejemplos sobre los tipos de exámenes que se pueden realizar, las preguntas son de prueba, por eso la información que contienen no es relevante.

### Examen tipo test respuesta única

Consta de preguntas tipo test con una única respuesta correcta. El número de respuestas puede variar desde dos a cinco, pero solo una es la correcta. Esto se muestra en 6.12.

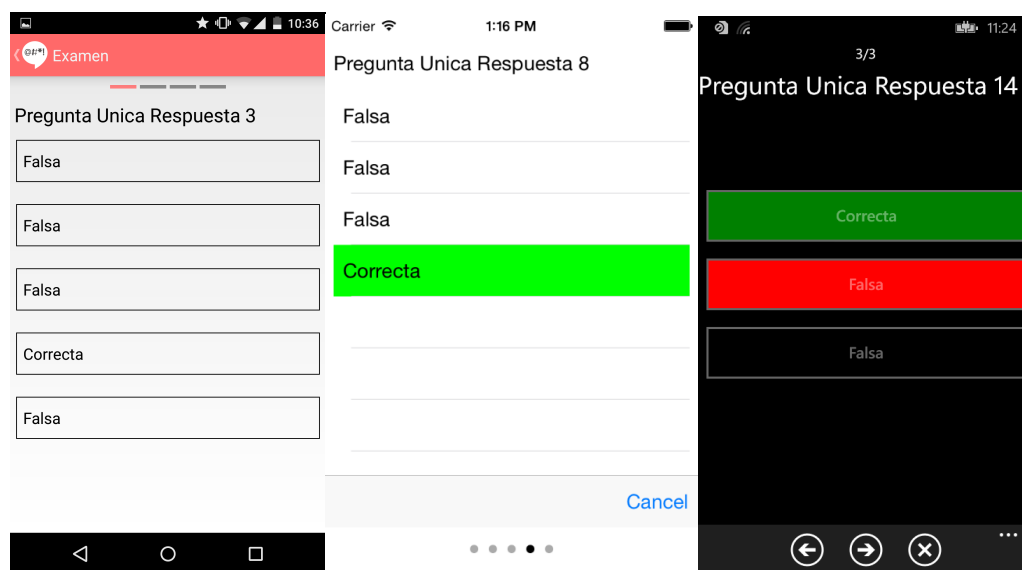


Figura 6.12: Examen tipo text única respuesta Android™, iOS™ y Windows Phone®

### Examen tipo test multirespuesta

Consta de preguntas tipo test con una o más de una respuesta correcta. Igual que en el caso anterior el número de respuestas puede variar de dos a cinco, pero en este caso puede haber más de una correcta, el número de respuestas correctas también puede variar desde una hasta todas las que se

muestran 6.13.

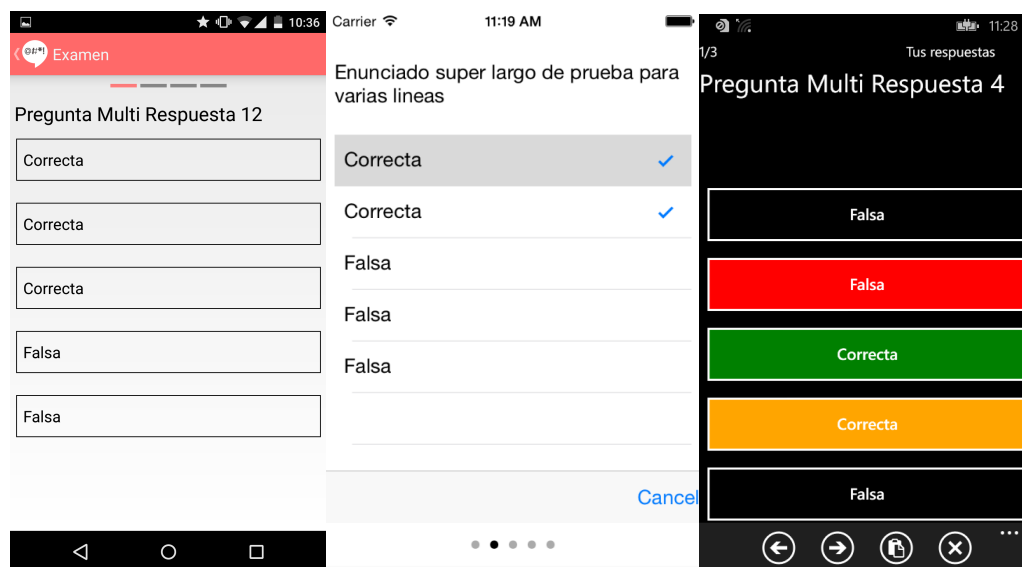


Figura 6.13: Examen tipo text multi-respuesta Android™, iOS™ y Windows Phone®

### Examen respuestas cortas

El tercer tipo de examen es respuestas cortas. En esta ocasión el usuario deberá escribir la respuesta que considere oportuna.

En la corrección de exámenes con respuesta corta se mostrarán en la misma pantalla la respuesta correcta y la que ha dado el usuario. En caso de que las respuestas coinciden se muestra la respuesta del usuario con fondo verde, en caso contrario se muestra con fondo rojo. Se puede visualizar un ejemplo en 6.14.

### Tipo respuestas en parejas

El último tipo de examen son los exámenes tipo respuestas en pareja. En este caso se mostrará dos columnas y el usuario deberá relacionarlas. Cada columna consta de tres respuestas. El usuario deberá relacionar cada una de las columnas con su opuesta.

El estilo en el que se muestra la corrección es bastante similar a los exámenes multi-respuesta, puesto que se dispone del botón que permite cambiar entre respuestas dadas por el usuario o la corrección. Cuando se muestran las respuestas dadas por el usuario se usarán diferentes tonalidades de verde para las que están correctas y diferentes tonalidades de rojo para las incorrectas.

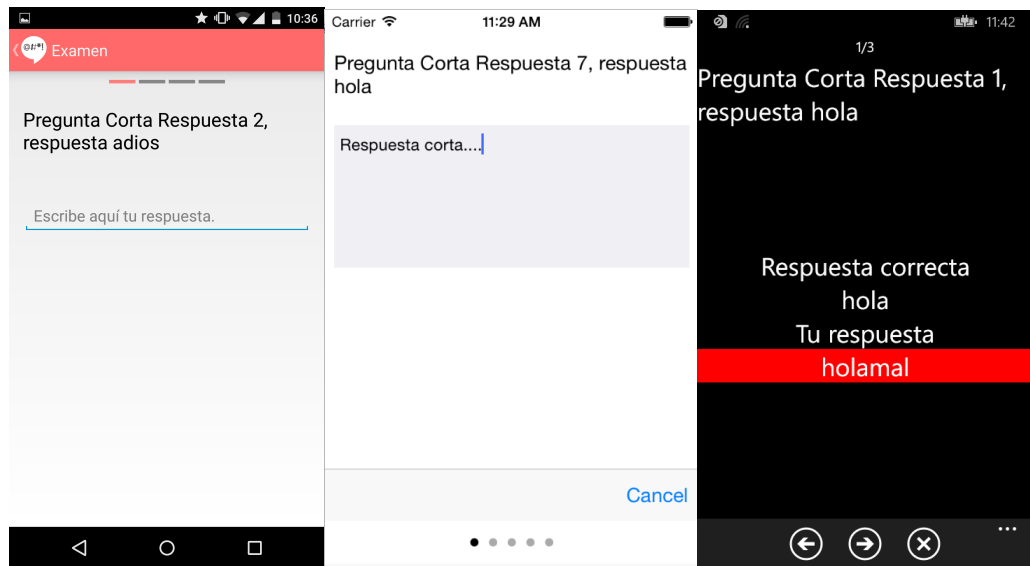


Figura 6.14: Examen tipo respuestas cortas Android™, iOS™ y Windows Phone®

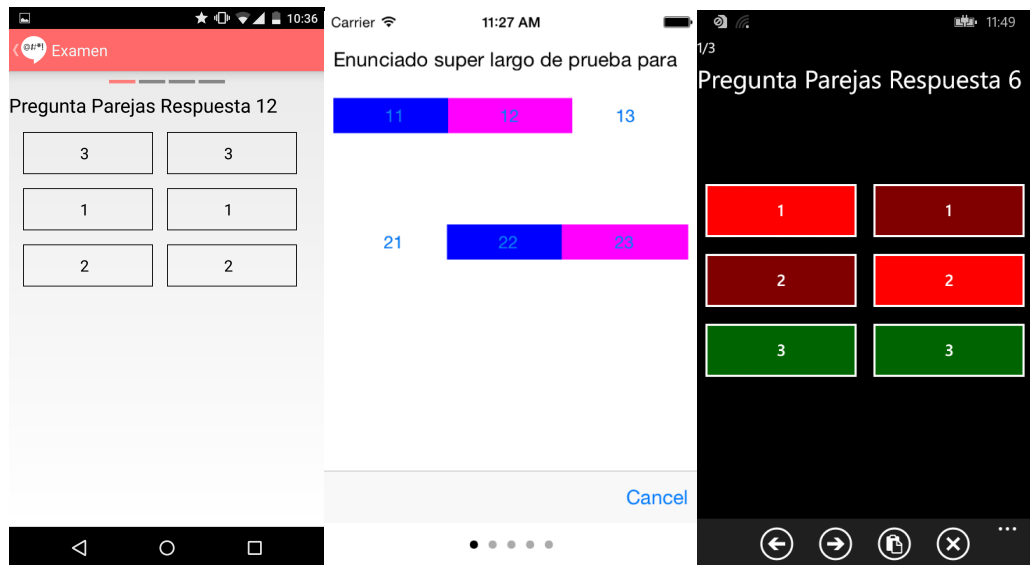


Figura 6.15: Examen tipo parejas Android™, iOS™ y Windows Phone®

### 6.2.10. UC9: Visualizar estadísticas

Tras realizar al menos un examen, se proporciona la posibilidad de obtener estadísticas relacionadas a la realización de los mismos.

La información que se mostrará será la nota media de los exámenes y el tiempo medio de realización agrupada por todos los exámenes realizados hasta la fecha.

Posteriormente se mostrarán (figura 6.16) dos gráficas: un diagrama circular indicando las preguntas acertadas, falladas y no contestadas; una gráfica lineal de los exámenes de forma individualizada que indican las fechas de realización de los mismos y la nota obtenida, preguntas acertadas, falladas y no respondidas.



Figura 6.16: Gráficas en Android™, iOS™ y Windows Phone®

### 6.2.11. UC10: Subir un examen/apuntes

Para poder subir un examen o apuntes se pedirán unos datos básicos para poder identificar el examen o los apuntes que el usuario desea compartir. En ambos casos el tema de la asignatura será algo opcional, puesto que puede ser un examen o apuntes de toda la asignatura y no de un tema específico. Una vez rellenados los datos y seleccionadas las imágenes ya está listo para enviar 6.17.

### 6.2.12. UC11: Consultar examen/apuntes

El apartado anterior no tendría ningún sentido si los usuarios no pudieran ver los exámenes o apuntes que se han subido, en este apartado se mostrará como ver esos apuntes. Al igual que en el apartado anterior, por su similitud

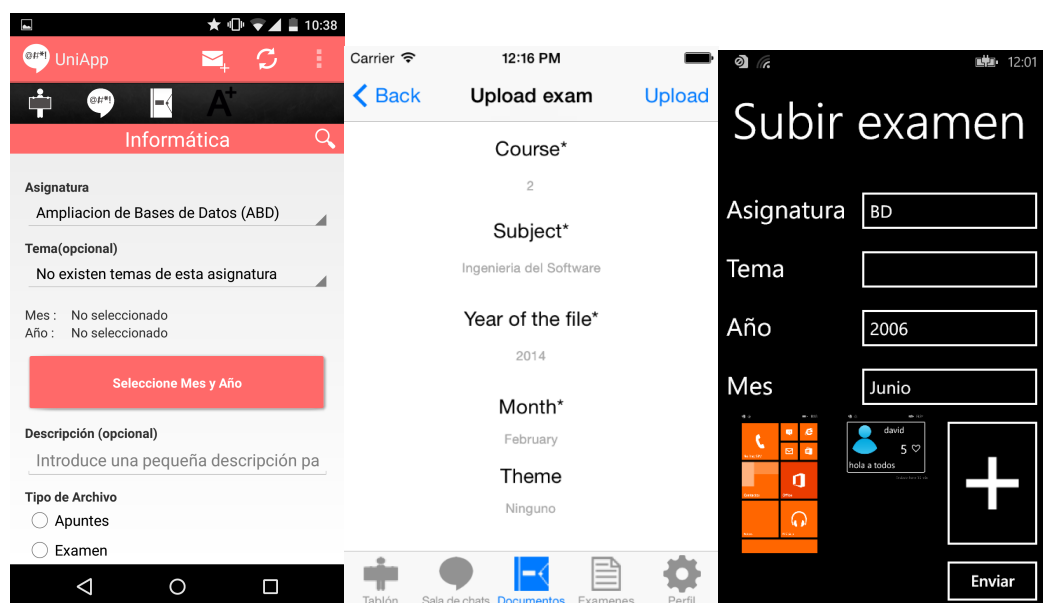


Figura 6.17: Subir exámen Android™, iOS™ y Windows Phone®

se explican juntos.

En esta vista de búsqueda existe un formulario en el cual se debe rellenar lo que quiere buscar. Una vez rellenado esto se le envía al servidor la petición y se abre una vista con todos los exámenes o apuntes que cumplen esos requisitos.

Se selecciona el que se quiere ver y se abre la vista individual del examen o los apuntes y se pueden ver las imágenes que lo forman 6.18.

### 6.2.13. UC12: Modificar perfil usuario

En este apartado se permiten múltiples modificaciones :

Para cambiar la foto de perfil solo hay que acceder al perfil de usuario, esto nos permite poner cualquier foto que se encuentre en el terminal. A partir de ese momento, se usará la nueva foto del usuario para identificarle .

Además, se permite a un usuario cambiar su contraseña y facultad. Por motivos de seguridad antes de realizar ningún cambio se pide la contraseña actual. Todos los parámetros son comprobados finalmente por el servidor, que nos indicará si el cambio ha sido satisfactorio o por el contrario existe algún error en los datos introducidos.

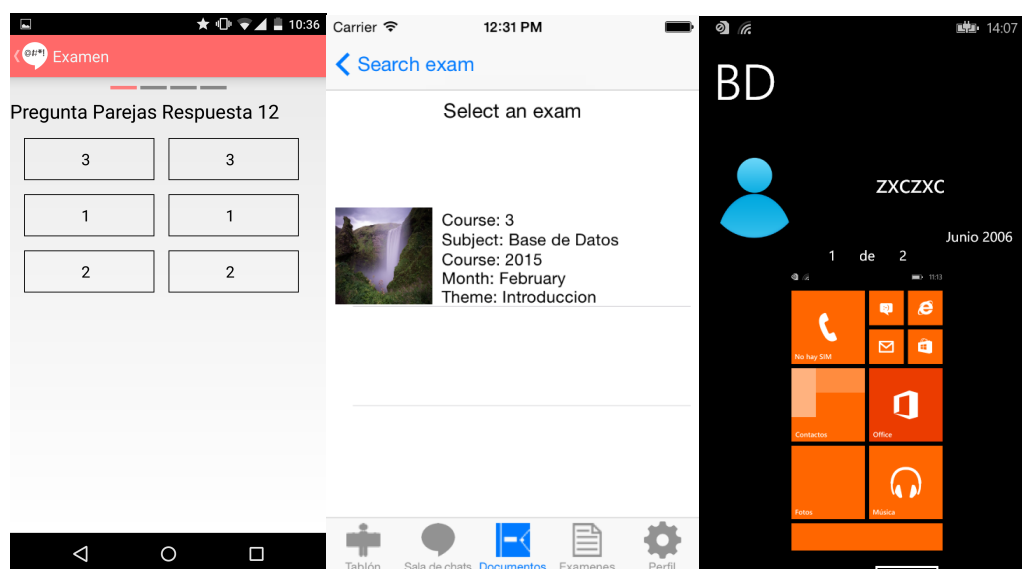


Figura 6.18: Subir exámen Android™, iOS™ y Windows Phone®

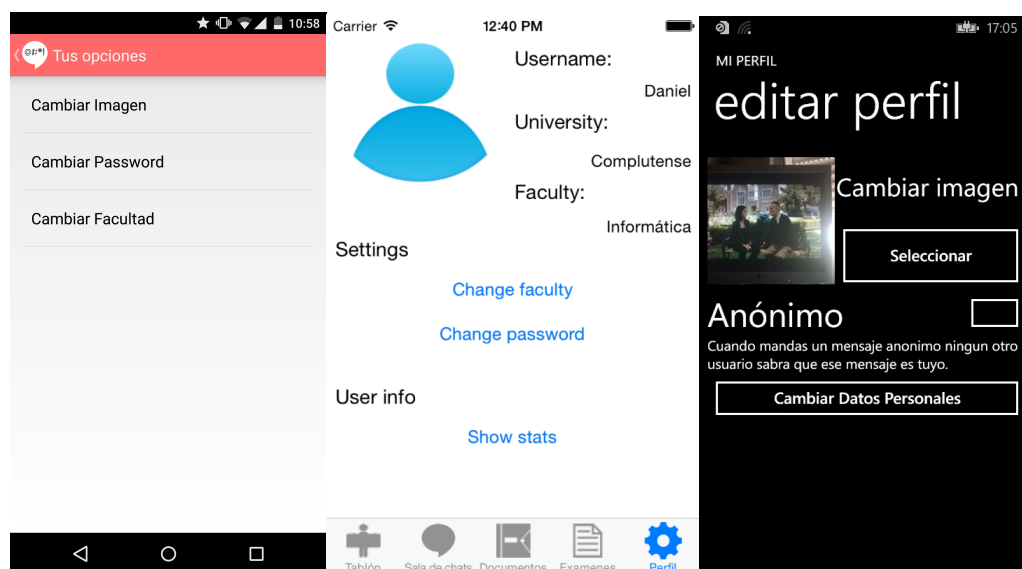


Figura 6.19: Cambiar perfil Android™, iOS™ y Windows Phone®

#### 6.2.14. UC13: Crear sala de chat

En este apartado los usuarios podrán crear salas con el objetivo de chatear con otros usuarios. Los mensajes de estas salas sólo los recibirán los usuarios que están conectados a dicha sala y la información no se almacena, es decir los mensajes no se guarda indefinidamente simplemente se muestran mientras

se permanece en la sala 6.20 .

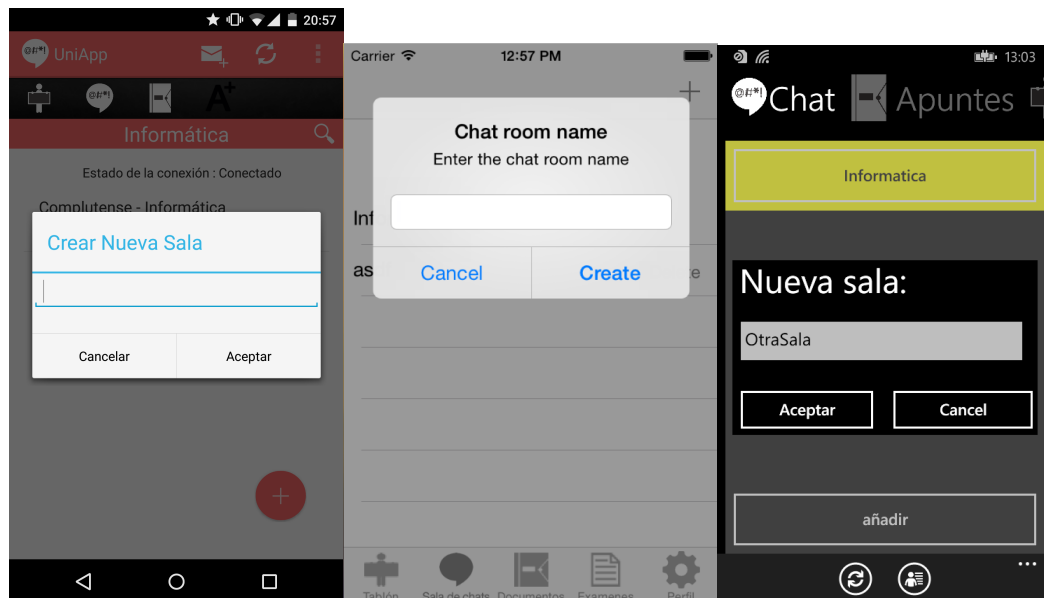


Figura 6.20: Chat en Android™, iOS™ y Windows Phone®

El funcionamiento del chat visto de la perspectiva del móvil consiste en conectarse a un socket TCP al puerto del servidor. Luego seguir los pasos de registro que sean oportunos y ya estará listo para recibir y transmitir.

#### 6.2.15. UC14: Enviar mensajes chat

Cuando el usuario considere oportuno a parte de poder conectarse a una sala podrá escribir en ella, de esta forma tendrá la posibilidad de interactuar con el resto de usuarios de forma inmediata 6.21.

El terminal lo que hará será encapsular la información en un JSON y mandárselo al servidor para que este lo distribuya entre el resto de usuarios.

#### 6.2.16. UC15: Recibir mensaje chat

Cuando el servidor envía los terminales conectados a una determinada sala los mensajes que han escrito otros usuarios, los terminales colocaran esta información en la pantalla de teléfono del usuario para que este pueda leerlo y responder si lo considera necesario.

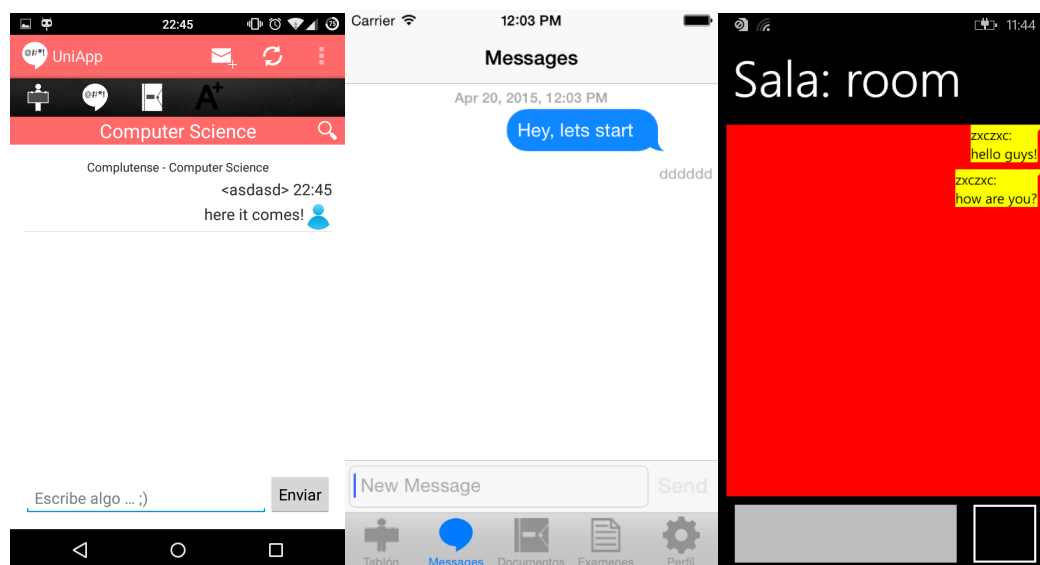


Figura 6.21: Chat en Android™, iOS™ y Windows Phone®

### 6.3. Contribución al proyecto

Como se ha visto en este mismo capítulo el proyecto consta de cuatro aplicaciones principales: servidor, aplicación para Android™, iOS™ y Windows Phone®.

La distribución general ha sido la siguiente:

Cada integrante del equipo del equipo desarrolló por completo cada una de las aplicaciones para dispositivos móviles y el servidor fue desarrollado de forma colaborativa, es decir, para cada problema a resolver, cada integrante propone una forma de resolver el problema y se decide entre los integrantes cual será la solución a implementar.

Es por este motivo, que es complicado identificar tareas realizadas por cada integrante, ya que se ha trabajado en equipo como un único integrante, siendo solo identificable el trabajo individual realizado por cada uno de las aplicaciones para dispositivos móviles.

Se ha realizado un proceso incremental e iterativo de desarrollo para cada uno de las distintas secciones que formaban parte, siendo estas:

- Identificación del caso de uso a resolver.
- Proposición de las distintas soluciones al problema.
- Selección de la solución a implementar.

- Implementación en el servidor de la solución decidida.
- Integración de las llamadas anteriormente implementadas en las distintas plataformas móviles.
- Pruebas de integración y de aplicación, localización de posibles errores y solución.

Por otra parte, se realizaron dos test a dos grupos distintos de alumnos de la Facultad de Informática de la Universidad Complutense de Madrid. El primer grupo, formado por alumnos de la asignatura de Diseño Automático de Sistemas, se les permitió explorar la aplicación, resolver un examen real realizado por José Miguel Montañana Aliaga, y resolver un test de opinión en aplicación.

Otro test se realizó a papel a un grupo de control, simplemente explicando en qué consiste la aplicación, pero sin permitir explorar la aplicación y solamente permitiendo responder a preguntas de opinión.

Para la realización de estos test, todos los integrantes realizaron una investigación leyendo diversos artículos científicos de prestigiosas revistas como *Computers on Education* entre otras, para tomar como ejemplo investigaciones anteriores para realizar preguntas similares o mejorar las ya realizadas y finalmente se procedió a la composición de las mismas, usandolas en los test anteriormente mencionados.

### 6.3.1. Daniel Serrano Torres

Desarrollo completo de la aplicación para la plataforma iOS™ y colaboración en el desarrollo del servidor. Dado que la plataforma de desarrollo de Apple®, iOS™, antes de comenzar el proyecto era completamente desconocida por el desarrollador se ha necesitado un un proceso de investigación en varias áreas o aspectos del proceso de desarrollo enumeradas a continuación:

- Si bien la plataforma de iOS™ puede ser desarrollada con dos lenguajes de programación, Swift y Objective C. Para el desarrollo de esta aplicación se ha optado por Objective C, que ha llevado un proceso de investigación continuo ya que el propio diseño interno del lenguaje dista notablemente en sintáxis e implementación respecto a otros lenguajes como pueden ser Java, Python, C++.
- El lenguaje de programación es una parte muy importante a aprender para el desarrollo de la aplicación, pero también lo es el diseño de la arquitectura y funcionamiento de las aplicaciones de este. Por lo tanto el estudio de la arquitectura de las aplicaciones de iOS™ ha

sido otra tarea de investigación realizada durante el proyecto, donde se han aprendido conceptos fundamentales como por ejemplo: el diseño de interfaces (Autolayout), almacenamiento interno (Core Data) y vinculación de código y elementos visuales (Storyboard).

- De cara a realizar un proceso de desarrollo adecuado también ha sido necesario un proceso de aprendizaje e investigación sobre la herramienta Xcode® (el IDE para iOS™), funcionamiento, atajos de teclado, sistema de depuración de errores, configuración de proyectos, integración de un gestor de dependencias para librerías externas, etc.
- Para realizar pruebas en un dispositivo real, y dado que no se dispone de presupuesto para un dispositivo con iOS™ actual ni para una licencia anual de desarrollador, se ha utilizado un dispositivo antiguo para el cual ha sido necesario un proceso de investigación con el objetivo de conseguir acceso al terminal (JailBreak) sin necesidad de poseer una licencia de desarrollador; y poder desplegar la aplicación de para realizar pruebas. Este proceso se ha realizado en un entorno de carácter educativo.

De cara al desarrollo de partes del servidor también se han realizado procesos de investigación que por ejemplo abarcan el diseño de un modelo basado en herencia con atributos personalizados, en término de estructura de datos a manejar entre el modelo y la base de datos.

### 6.3.2. David García Álvarez

Desarrollador de la aplicación de Windows Phone® y colaboración en el desarrollo del servidor. Sin conocimientos previos de esta plataforma, a parte del desarrollo de la aplicación, hubo un trabajo previo de investigación y aprendizaje del lenguaje C# y .NET.

Como parte a destacar del trabajo en Windows Phone®, a parte de la implementación de los casos de uso, tenemos:

- Obtención de una licencia de Visual Studio® e instalación. Adaptación a este nuevo entorno de desarrollo.
- Obtención de licencia de desarrollo por parte de Microsoft®, para Windows Phone®.
- Lectura y desarrollo de los ejemplos del libro Desarrollo en Windows 8 y Windows Phone 8 con XAML y C# [33].
- Lectura y desarrollo de los ejemplos del The Absolute Beginner [34].

- Investigación sobre la integración de base de datos SQLite en Windows Phone®.
- Instalación y estudio del plugin para Visual Studio® que permite verificar si la app desarrollada cumple los requisitos mínimos para la tienda.
- Investigación y publicación en la tienda de aplicaciones de Microsoft®.

En la parte del servidor ha trabajado con básicamente todo lo correspondiente a la implementación, pero destacar la investigación del funcionamiento de la librería Twisted y la creación del protocolo de comunicación para el chat.

### 6.3.3. Juan Luís Pérez Valbuena

Realización íntegra del diseño e implementación de la aplicación Android™ con aproximadamente 10.000 líneas de código Java y colaboración en el desarrollo del servidor.

Otras tareas que fueron realizadas por este integrante:

- Investigación y pruebas de entorno de desarrollo Android Studio™, no utilizado hasta la fecha. Conocimientos previos de Java™.
- Desarrollo de todos los casos indicados anteriormente.
- Obtención de licencia de desarrollador, investigación y publicación en la tienda de aplicaciones de Google™, Play Store™.
- Instalación y configuración de un servidor privado virtual para producción en CentOS, instalando y configurando los servicios tales como Apache® y PostgreSQL™ además de medidas de seguridad como iptables para evitar accesos no autorizados.
- Obtención y configuración del nombre de dominio "bsodsoftware.me", que es actualmente donde se encuentra situado el servidor de producción.
- Realización de script en Shell Script en bash para automatización para el despliegue de la aplicación en el servidor anteriormente mencionado.

## Capítulo 7

# Discusión y conclusión

### 7.1. Discusión sobre objetivos

En la propuesta del proyecto en el capítulo tres se han descrito los objetivos planteados a abordar por este proyecto. A continuación se va a analizar cómo se han conseguido dichos objetivos.

#### 7.1.1. Objetivos principales

Los objetivos principales eran los siguientes:

- Crear un sistema de noticias denominado tablón.
- Incorporar mensajería instantánea sin almacenamiento de las conversaciones.
- Realizar un módulo para realizar exámenes de distintos tipos.
- Crear una sección donde el usuario pueda ver y gestionar su información personal; un perfil.
- Poner a disposición de los usuarios profesores un panel de administración para crear exámenes.
- Un sistema de compartición de material y exámenes.

Cada uno de estos objetivos ha cumplido completamente como se expone en el capítulo cinco en la sección de trabajo realizado, donde se puede observar el funcionamiento detallado de cada uno de ellos estructurado por casos de uso.

### 7.1.2. Objetivos extraordinarios

Se expuso, como uno de los objetivos extraordinarios, la disminución de las distracciones y la multitarea. Nuestro proyecto puede otorgar a las lecciones un carácter interactivo (proveniente del E-Learning) y que además ayude a disminuir el número de veces o el tiempo invertido en utilizar el dispositivo móvil para tareas ajenas a la lección.

La colaboración, otro aspecto que en la educación puede ser un factor muy útil y necesario pero que puede tornarse complicado de llevar a cabo de manera que resulte de provecho. Mencionado ya anteriormente (capítulo dos) un factor que dificulta la colaboración de grupos de estudio o trabajo es la necesidad de que los miembros de estos requieran reunirse físicamente para compartir material de trabajo, para este punto concreto se han elegido las plataformas móviles que otorgan un carácter ubicuo a la aplicación permitiendo superar esta barrera; y por otro lado también como factor importante es la comunicación de los miembros del grupo de trabajo, lo cual ha sido solventado en el proyecto con un sistema de chat IRC.

Este proyecto pretende aportar al sistema educativo de una herramienta útil y que favorezca la enseñanza E-Learning. Por ello también es interesante la difusión del conocimiento de forma más sencilla, la sección de compartir ficheros es con la cual cumplimos este objetivo y fomentamos la difusión de material educativo. Por otro lado esta funcionalidad también mejora el carácter colaborativo dado que permite compartir los documentos necesarios para realizar un proyecto en grupo.

Y como último objetivo extraordinario se ha planteado disminuir el abandono de los sistemas E-Learning, es aquí donde volvemos sobre la decisión de desarrollar el proyecto sobre dispositivos móviles que nos otorga el factor ubicuo al sistema. Con ello aportamos mayor flexibilidad al usuario para utilizar la aplicación cuando y donde desee, y con ello pueda aprovechar los momentos de motivación u ociosos para utilizarla en vez de tener un horario fijo y un lugar donde debe ir a realizar la tarea. Esto puede ayudar a aumentar la motivación del usuario al utilizar la aplicación, siendo este uno de los factores discutidos en el capítulo dos.

Para finalizar la discusión sobre los objetivos cabe decir que este proyecto ha cumplido con todos los objetivos buscados en su propuesta e incluso con algunos más sin ser estos parte de los requisitos.

## 7.2. E-Learning, futuro

Este método de enseñanza se ha extendido ampliamente y en diferentes formas y ámbitos, y gracias a la tecnología, cada vez las herramientas son mejores.

Por otro lado viendo que el E-Learning viene de hace cerca de veinte años y cada vez ha ido a más no creemos que sea una tendencia a desaparecer, por lo cual consideramos buena idea apostar e invertir en su desarrollo.

Dado que la enseñanza es increíblemente extensa y variada el E-Learning lleva tiempo tratando de complementar la enseñanza de una forma efectiva y apreciable. Existen muchos campos donde el E-Learning ha enfocado su desarrollo: micro aprendizajes, se centra en diseñar actividades con pasos cortos en entornos de medios digitales lo cual implica en el día a día del conocimiento de muchos trabajadores; gamificación, es un campo del E-Learning enfocado a llevar la educación y los conocimientos a través de los videojuegos, desarrollando puzzles y resolviendo problemas; aprendizaje personalizado, trata de personalizar la enseñanza individualmente donde incluso el estudiante puede escoger lo que desea aprender, pudiendo definirse metas a conseguir y tipos de recompensar por hacerlo.

Todo lo mencionado deja ver que el E-Learning tiene una gran proyección por delante de evolución y mejora, lo cual hace una buena decisión plantearse desarrollar sistemas como el de este proyecto.

## 7.3. Trabajo futuro

Como ya se ha expuesto el E-Learning abarca una gran amplitud de campos, temas y formas de mejorar procesos de enseñanza, esto lleva a que nuestro proyecto tiene una amplia capacidad de mejora e incorporación de nuevas características. Pasaremos a exponer algunas mejoras que consideramos se pueden incluir en un futuro al sistema desarrollado.

Una de las funcionalidades de la aplicación más extensible es la sección de exámenes, actualmente se han desarrollado cuatro tipos de controles que se pueden realizar. Es aquí donde se pueden incluir tantos nuevos modelos de preguntas como se puedan ocurrir, por ejemplo completar frases con un conjunto de palabras (extensible a fragmentos de código en el caso de la informática).

En la sección del tablón se puede incorporar un sistema de menciones a las noticias de otros usuarios, para así poder relacionar noticias y contenidos

entre si.

Como todo software con interfaz de usuario siempre hay mejora para la capa de presentación, otorgando mejor usabilidad proporcionada por la retroalimentación de los usuarios.

Para la sección de ficheros se puede incorporar una galería a parte del sistema de filtro ya existente para encontrar los documentos compartidos más fácilmente.

Al chat se le puede incorporar un sistema de invitación a salas creadas por otros usuarios.

Finalmente, se podría desarrollar otra aplicación, como por ejemplo para PC, Mac® o GNU Linux ya que la arquitectura actual permite una completa escalabilidad.

Todas estas son sólo algunas de las ideas de mejora para este proyecto, se puede apreciar la amplitud de mejora de un sistema M-Learning como el que se ha desarrollado.

## 7.4. Conclusión

Este proyecto ha conseguido cumplir con todos los objetivos que buscaba, si bien, a parte de ser una herramienta de soporte a la educación (LMS) con una enumeración de funcionalidades que simplemente asisten, extienden y/o mejoran ciertas tareas en la educación, también ha sido diseñada con la idea de cumplir unos objetivos que van más allá que el mero hecho de desarrollar un software, es decir, varios de los objetivos de este proyecto tratan de aportar soluciones o mejoras sobre problemas y factores implicados en cualquier proceso de E-Learning que dificultan su éxito.

Durante el desarrollo del proyecto se ha realizado una encuesta a dos grupos de alumnos, uno de estos grupos la realizó dentro de la aplicación móvil, lo cual por otro lado demuestra el potencial y el correcto funcionamiento del módulo de exámenes dado que dicha encuesta se realizó a través de este.

Por un lado se ha evaluado el correcto funcionamiento y la utilidad de módulo de exámenes denotando la capacidad polivalente de este, dado que permite realizar encuestas o tipos de cuestionarios similares a parte de exámenes y controles. Otras secciones han sido probadas de forma satisfactoria en los entornos de pruebas pertinentes en todas las plataformas móviles.

# Capítulo 8

## Discussion and conclusion

### 8.1. Discussion on objectives

In the project proposal in chapter three we have described the objectives to be addressed by this project. Then it will be analyzed how these objectives have been achieved.

#### 8.1.1. Main objectives

The main objectives was the following:

- To develop of a news system called time line
- To include a real-time messaging system without storing the conversations
- To develop a module for performing exams of different kind.
- To create a section where the user can be able to manage their personal information; a profile section.
- Make available to teachers an administration panel to create tests.
- A sharing system of education material.

Each of these objectives has completely fulfilled as discussed in chapter five in the work section, where you can see the detail of each structured by use cases.

#### 8.1.2. Extraordinary objectives

It was exposed as one of the extraordinary objectives, to lower distractions and the multitasking. Our project can give an interactive character lessons (from the E-Learning) and also help to decrease the number of times

or the time taken to use the mobile device tasks outside lesson time.

Collaboration, another aspect that education can be a very useful and necessary factor but can become complicated to achieve a way that can be profitable. Already mentioned above (Chapter Two) one factor hindering the cooperation of groups of study or work is the need for the members of these require physically meet to share work material for this particular point are chosen mobile platforms to grant a ubiquity allowing the application to overcome this barrier; and ,on the other hand, also as an important factor, is communication of the members of the working group, which has been solved in the project with a real-time chat system (IRC like).

This project aims to provide to the educational system a useful tool that promotes E-learning. It is therefore also interesting disseminating knowledge more easily, share files section is with which we meet this goal and encourage the dissemination of educational material. On the other hand this functionality also improves the collaborative nature since sharing the necessary documents for a group project.

As last extraordinary objective has been propoused to decrease the abandonment of the E-Learning systems, this is where we go back on the decision to develop the project on mobile devices that gives us the ubiquitous factor system. Thereby we bring greater flexibility to the user to use the application when and where you want, and thus can take advantage of motivation or idle moments to use rather than having a fixed time and place where you should go to perform the task. This can help increase user motivation to use the application, being one of the factors discussed in chapter two.

To conclude the discussion on the objectives can say that this project has fulfilled all the objectives addressed in its proposal and even some more without these being part of the initial requirements.

## 8.2. E-Learning, future

This method of teaching is widespread and in different forms and areas, and thanks to technology, the tools are getting better. On the other hand seeing the E-Learning is about twenty years and each time has gone but not believe it is a tendency to disappear, which we consider good idea to bet and invest in their development.

Since education is incredibly extensive and varied E-Learning takes time trying to supplement the teaching of effective and measurable way. There are many fields where the E-Learning has focused its development: micro

learning, it focuses on design activities with short steps in digital media environments, which means the day of knowledge of many workers; Gamification is a field of E-Learning focused on bringing education and knowledge through games, puzzles and solving development problems; personalized learning, teaching is customized individually where even students can choose what you want to learn and can achieve defined goals and types of reward for doing so.

All of the above reveals that the E-Learning has a great future ahead of evolving and improving, making a good decision to consider developing systems such as this project.

### 8.3. Future work

As already stated the E-Learning covers a wide range of fields, issues and ways to improve teaching processes, this leads to our project has ample capacity for improvement and incorporation of new features. We will expose some improvements that we believe may be included in the developed system.

One of the features of the application is more extensible is the test section, currently has developed four types of controls that can be performed. This is where you can include as many new models of questions as may occur, for example complete sentences with a set of words (extensible code snippets in the case of computers science). Timeline section can incorporate a system of references to news from other users, in order to relate to each news and other content.

As every software with user interface there is always improvement to the presentation layer, providing better usability with the feedback provided by the users.

To file section can incorporate a gallery of the existing filter system to find shared documents more easily

The chat can be incorporated a system of inviting rooms created by other users.

Finally, another application could be developed, such as PC, Mac® or GNU Linux because the current architecture allows full scalability.

These are just some of the ideas for improvement for this project, you can appreciate the extent of improvement of an M-Learning system as it has developed.

## 8.4. Conclusion

This project has managed to achieve all the objectives sought, although, besides being a tool of support to education (LMS) with a list of features that simply attending, extend and/or improve certain tasks in education, also it has been designed with the idea of achieve objectives beyond the mere fact develop a software, that is, more of the objectives of this project are trying to provide solutions or improvements on issues and factors involved in any process of E-learning that difficult their success.

During development of the project has been surveyed two groups of students, one of these groups performed within the mobile application, which on the other hand shows the potential and the proper functioning of the module examinations since the survey was performed through this.

On one side it has assessed the correct functioning and utility of module examinations denoting the capacity of this versatile, since it allows surveys or questionnaires to similar types of examinations and controls. Other sections have been tested successfully in environments relevant on all mobile platforms.

# Glosario

**API** (Application Programming Interface)

Una interfaz definida y conocida por los clientes para realizar peticiones de información.

**Cliente**

Cualquier sistema que realice peticiones de información al sistema en el servidor.

**Backend**

Sistema que se ejecuta en el servidor para proveer información o realizar tareas.

**Branch**

En el contexto de un sistema de control de versiones hace referencia a una rama, es decir, un flujo de trabajo derivado del principal.

**Framework**

Estructura y librería de software.

**HTTP** (HyperText Transfer Protocol)

Es un protocolo de intercambio de información en formato de texto a través de conexiones de red para sistemas web.

**IDE** (Integrated Development Environment)

Un entorno integrado de desarrollo es un conjunto de herramientas que facilitan el proceso de desarrollo de un producto software.

**JSON** (JavaScript Simple Object Notation)

Formato para representar información de forma sencilla y ligera.

**ORM** (Object Relational Mapping)

Sistema de abstracción del modelo y la base de datos, permitiendo objetos del lenguaje de programación en vez de lenguajes como SQL.

**Servidor**

Plataforma conectada a internet que alberga una serie de servicios provistos a los clientes que entablen comunicación con esta.

**SOAP** (Simple Object Access Protocol)

Protocolo que define la comunicación entre dos objetos de diferentes procesos a través de XML.

**Token**

Identificador único para comunicaciones con identificación de cliente.

**XML** (eXtensible Markup Language)

Lenguaje de marcado para representar información.

# Acrónimos

**API:** Application Programming Interface  
**CBT:** Cognitive Behavioral Therapy  
**E-Learning:** Electronic Learning  
**HTTP:** HyperText Transfer Protocol  
**IDE:** Integrated Development Enviroment  
**JSON:** JavaScript Simple Object Notation  
**LMS:** Learning Management System  
**M-Learning:** Mobile Learning  
**SQL:** Structured Query Language  
**T-Learning:** Television Learning  
**U-Learning:** Ubiquitous Learning



## Apéndice A

# ORM API en Django

### A.1. Modelos para el ejemplo

```
from django.db import models

class Blog(models.Model):
    name = models.CharField(max_length=100)
    tagline = models.TextField()

    def __str__(self):
        return self.name

class Author(models.Model):
    name = models.CharField(max_length=50)
    email = models.EmailField()

    def __str__(self):
        return self.name

class Entry(models.Model):
    blog = models.ForeignKey(Blog)
    headline = models.CharField(max_length=255)
    body_text = models.TextField()
    pub_date = models.DateField()
    mod_date = models.DateField()
    authors = models.ManyToManyField(Author)
    n_comments = models.IntegerField()
    n_pingbacks = models.IntegerField()
    rating = models.IntegerField()

    def __str__(self):
```

```
return self.headline
```

## A.2. Creando un nuevo registro en la base de datos

```
from blog.models import Blog
b = Blog(name='Beatles Blog', tagline='All the latest Beatles news.')
b.save()
```

## A.3. Consultando todos los registros de la entidad Entry

```
all_entries = Entry.objects.all()
```

## A.4. Relaciones entre tablas a través de los modelos

```
entry = Entry.objects.get(pk=1)
cheese_blog = Blog.objects.get(name="Cheddar Talk")
entry.blog = cheese_blog
entry.save()
```

## Apéndice B

### Enlaces de interés

#### Aplicaciones móviles

- Aplicación para Android™: <https://play.google.com/store/apps/details?id=com.bsod.uniapp>
- Aplicación para Windows Phone®: <https://www.windowsphone.com/s?appid=8a9f1939-e74b-4d1a-89bd-0a96fa8355a1>

#### Repositorios de código fuente

- Aplicación para Android™: <https://github.com/PaytonZ/TFG-droid>
- Aplicación para Windows Phone®: <https://github.com/own3dh2so4/TFGWP>
- Aplicación para iOS™: <https://github.com/plandevinda/TFG-iOS>
- Aplicación para el servidor: <https://github.com/plandevinda/TFG-Webservice>



# Bibliografía

- [1] “Mobile usage,” 2015. [Online]. Available: <http://www.emarketer.com/Article/Smartphone-Users-Worldwide-Will-Total-175-Billion-2014/1010536>
- [2] “Mobile operating system usage.” [Online]. Available: <http://www.idc.com/getdoc.jsp?containerId=prUS25450615>
- [3] “Iphone 1st generation.” [Online]. Available: [https://support.apple.com/kb/sp2?locale=es\\_ES](https://support.apple.com/kb/sp2?locale=es_ES)
- [4] “Android 2nd release.” [Online]. Available: [http://developer.android.com/sdk/OLD\\_RELEASENOTES.html#m3-rc22a](http://developer.android.com/sdk/OLD_RELEASENOTES.html#m3-rc22a)
- [5] “Epignosis llc,” vol. E-Learning 101, 2014. [Online]. Available: <http://www.talentlms.com/elearning>
- [6] L.-H. Huang, C.-R. dow, and P. H. Yi-Hsung Li, “u-ta: A ubiquitous teaching assistant using knowledge retrieval and adaptative learning techniques,” *Comput Appl Eng Educ*, vol. 21, issue 2, 245-255, 2013.
- [7] C.-R. dow and L.-H. Huang, “Context-aware and lbs learning systems using ubiquitous teaching assistant (u-ta): A case study for service-learning courses,” *Comput Appl Eng Educ*, vol. 22, issue 4, 604-616, 2010.
- [8] D. de Andrés, J.-C. Ruiz, J. M. Montañana, D. del Blanco, and J. Lozano, ““t-learning: oportunidades y limitaciones”, ii jornada de innovación docente ets ingeniería informática,” *JID-INF 2010. Universitat Politècnica de Valencia*, 2010.
- [9] “Facebook info.” [Online]. Available: <http://newsroom.fb.com/company\discretionary\{\}\{\}\info/>
- [10] “Social networks statistics.” [Online]. Available: <https://www.globalwebindex.net/>

- 
- [11] I. Hamburg, “elearning 2.0 and social, practice-oriented communities to improve knowledge in companies,” *IEEE Computer Society*, pp. 411–416, DOI:10.1109/ICIW.2010.68, 2010.
- [12] M. E. Edrees, “elearning 2.0: Learning management systems readiness. e-learning best practices in management, design and development of e-courses: Standards of excellence and creativity?” *Fourth International Conference on*, pp. 90–96, DOI:10.1109/ECONF.2013.57, 2013.
- [13] D. Xanthidis and A. S. Alali, “Effects of social media on elearning development in the gcc. computer applications & research (wscar),” *World Symposium on*, pp. 1–6, DOI:10.1109/WSCAR.2014.6916787, 2014.
- [14] C. Jones, R. Ramanau, S. Cross, and G. Healing, “Net generation or digital natives: Is there a distinct new generation entering university?” *Computers & Education (April 2010)*, vol. 54, Issue 3, 722-732, 2010.
- [15] M.-C. Lee, “Explaining and predicting users continuance intention toward e-learning: An extension of the expectation-confirmation model,” *Computers & Education*, vol. 54, Issue 2, 506-516 (Feb), 2010.
- [16] C. Calderwood and E. M. C. Phillip L. Ackerman, “What else do college students “do” while studying? an investigation of multitasking,” *Computers & Education*, vol. 75, 19-29. (Jun), 2014.
- [17] P.-C. Sun, R. J. Tsai, G. Finger, Y.-Y. Chen, and D. Teh, “What drives a successful e-learning? an empirical investigation of the critical factors influencing learner satisfaction,” *Computers & Education*, vol. 50, Issue 4, 1183-1202 (May), 2008.
- [18] F. Sana, T. Weston, and N. J. Cepeda, “Laptop multitasking hinders classroom learning for both users and nearby peers,” *Computers & Education*, vol. 62, 24-31 (Mar), 2013.
- [19] J. L. V. Barbosa, R. Hahn, D. N. F. Barbosa, and W. Segatto, “Intensive use of mobile technologies in a computer engineering course,” *Comput Appl Eng Educ*, vol. 22, issue 2, 329-339 (Jun), 2014.
- [20] J. L. V. Barbosa, R. Hahn, D. N. F. Barbosa, and C. F. R. Geyer, “Learning in small and large ubiquitous computing environments, proceedings of the ieee/ifip international conference on embedded and ubiquitous computing,” *IEEE*, pp. 17–20, 2008.
- [21] H. Ogata and Y. Yano, “Supporting awareness in ubiquitous learning,” *Int J Mob Blend Learn 1*, pp. 1–11, 2009.
- [22] B.-Y. Shih, C.-Y. Chen, and C. E. Li, “The exploration of the mobile mandarin learning system by the application of triz theory,” *Comput Appl Eng Educ*, 2010.

- 
- [23] “Iniciativa de movilidad en la red académica.” [Online]. Available: <http://www.eduroam.es/>
- [24] N. Hoic-Bozic, V. Mornar, and I. Boticki, “A blended learning approach to course design and implementation,” *Education, IEEE Transactions on*, vol. 52, issue 1, 19-30, DOI:10.1109/TE.2007.914945, 2009.
- [25] V. Bevanda, J. Azenovié, and D. Muic, “Privacy preserving in e-learning environment (case of modeling hippocratic database structure),” *P. Informatics*, pp. 47–52, BCI '09. Fourth Balkan Conference in, DOI:10.1109/BCI.2009.20, 2009.
- [26] “Internet relay chat rfc 2812,” <https://tools.ietf.org/html/rfc2812>.
- [27] “Most popular desktop ides & code editors.” [Online]. Available: <https://blog.codeanywhere.com/most-popular-ides-code-editors/>
- [28] “Windows phone sdk 8.0, + requisitos del sistema.” [Online]. Available: <https://www.microsoft.com/es-es/download/details.aspx?id=35471>
- [29] “Getting started developing apps for windows phone 8 and windows 8.” [Online]. Available: [https://msdn.microsoft.com/en-us/library/windows/apps/jj714071\(v=vs.105\).aspx](https://msdn.microsoft.com/en-us/library/windows/apps/jj714071(v=vs.105).aspx)
- [30] “Ventajas de utilizar patrones de diseño.” [Online]. Available: [http://www.ciberaula.com/articulo/ventajas\\_patrones/](http://www.ciberaula.com/articulo/ventajas_patrones/)
- [31] “Python decorators.” [Online]. Available: <https://wiki.python.org/moin/PythonDecorators>
- [32] “Twisted matrix labs, what is twisted?” [Online]. Available: <https://twistedmatrix.com/trac/>
- [33] “Desarrollo en windows 8 y windows phone 8 con xaml y c#.” [Online]. Available: [http://www.campusmvp.es/catalogo/Product-Desarrollo-en-Windows-8-y-Windows-Phone-8-con-XAML-y-C\\_163.aspx](http://www.campusmvp.es/catalogo/Product-Desarrollo-en-Windows-8-y-Windows-Phone-8-con-XAML-y-C_163.aspx)
- [34] “The absolute beginner.” [Online]. Available: <https://absolutebeginner.codeplex.com/releases/view/108480>