

**UNIVERSIDAD AUTÓNOMA DE MADRID**

**ESCUELA POLITÉCNICA SUPERIOR**



**PROYECTO FIN DE CARRERA**

**GESTOR DE EXÁMENES Y CALIFICACIONES EN  
ANDROID**

**Ing. Telecomunicación**

**Diego Miguel Arribas Marcos**

**Enero 2016**



# **GESTOR DE EXÁMENES Y CALIFICACIONES EN ANDROID**

**AUTOR: Diego Miguel Arribas Marcos**  
**TUTOR: Carlos Minchola Guardia**  
**PONENTE: Eduardo Boemo Scalvinoni**

Digital System Laboratory  
Dpto. Tecnología Electrónica y de Comunicaciones  
Escuela Politécnica Superior  
Universidad Autónoma de Madrid  
Enero 2016



## **Resumen**

Este proyecto consiste en la realización de una aplicación para dispositivos Android. Dicha aplicación permite crear y gestionar exámenes (o preguntas de autoevaluación) de respuestas múltiples en Android, genéricos para cualquier asignatura.

El proyecto consta de dos aplicaciones, una web donde acceden los profesores y administración para gestionar los exámenes y una aplicación Android a la que acceden los alumnos desde sus smartphones o tablets para llevar a cabo los exámenes.

El acceso como usuario al Módulo de Administración permite añadir y eliminar alumnos, añadir y eliminar profesores, añadir y eliminar asignaturas y asignar profesores y alumnos a cada asignatura. En cambio, el acceso al Módulo de Profesores solo permite acceder a las asignaturas que les asignó Administración. Asimismo, éstos pueden crear exámenes, programar cuándo activarlos y desactivarlos, comunicarse con los alumnos y consultar las notas de los estudiantes tras el examen. Por su parte, el acceso al Módulo de Alumnos no se lleva a cabo vía web, sino a través de dispositivos Android, desde los cuales se permite a los estudiantes llevar a cabo los exámenes, consultar sus calificaciones y comunicarse con los profesores.

La aplicación web se encuentra en una web (Hostinger) que provee de alojamiento web y hosting de forma gratuita. La parte Android del proyecto se encuentra publicada en Google Play Store, donde existe la posibilidad de descargársela para su uso.

## **Palabras Clave**

Examen, Teléfono Móvil, Smartphone, Tablet, Nota, Calificación, Android, Administración, Profesores, Alumnos, Web, Contraseña.

## **Abstract**

This project involves the accomplishment of an application for Android devices. This application lets you create and manage multiple choices exams (or self-assessment questions) in Android, generics to any subject.

The Project consists of two applications, website where teachers and administrators access to manage exams and an Android application that students access from their smartphones or tablets to conduct the tests.

The user access to the Administration Module allows to add and remove students, add and delete teachers, add and remove subjects and assign teachers and students to each subject. On the other hand, log in to the Professors Module allows to access the subjects assigned to them by Administration. Also, teachers can create examinations, schedule when to activate and deactivate the exams, communicate with the students and review the students' notes after the exam. Meanwhile, access to the Student module is not carried out via the web, but through Android devices, from which students are allowed to carry out the tests, check their marks and communicate with the teachers.

The web application is a web (Hostinger) which provides free hosting. The Android part of the project is published in Google Play Store, where there is the possibility download it.

## **Key Words**

Exam, Mobile Phone, Smartphone, Tablet, Mark, Scores, Android, Administration, Teachers, Students, Web, Password.

## *Agradecimientos*

A la mujer de mi vida  
Por todos maltratada  
Cara algo arrugada  
De los cielos caída

Querer es su virtud  
Esa es la verdad  
Tormenta de bondad  
De gran magnitud

Jamás una promesa incumplida  
Madre de Plaza de Mayo  
Si hace frío, a mí el sayo  
Paciencia más que desmedida

Salida de emergencia  
Única y fiel seguidora  
Consejera, gestora  
Me aterra tu ausencia

Por el destino embestida  
Del destino es candela  
Si no duermo, se desvela  
Antes muerta que rendida

Bendito tu nombre  
El sol que hace falta  
La cabeza bien alta  
Quien me hizo hombre

Tu hijo se confiesa:  
Gracias, Teresa.

# ÍNDICE DE CONTENIDOS

<b>1 Introducción</b> .....	<b>1</b>
1.1 Motivación.....	1
1.2 Objetivos.....	2
1.3 Organización de la memoria.....	2
<b>2 Estado del arte</b> .....	<b>5</b>
2.1 Historia de los terminales móviles.....	5
2.2 Internet.....	8
2.2.1 Historia de Internet .....	8
2.2.2 Otros elementos empleados en la parte web.....	11
2.3 Sistemas Operativos de dispositivos móviles.....	13
2.3.1 Android.....	13
2.3.2 Otros Sistemas Operativos.....	19
2.4 Lenguajes de programación web empleados .....	23
2.5 Aplicaciones similares .....	25
<b>3 Diseño</b> .....	<b>35</b>
3.1 Aplicación Web .....	35
3.1.1 Finalidad de la aplicación web .....	35
3.1.2 Elección de requisitos: parte web .....	36
3.1.3 Módulos: parte web .....	38
3.1.4 Limitaciones: parte web .....	40
3.2 Aplicación Android .....	41
3.2.1 Finalidad de la aplicación Android.....	41
3.2.2 Elección de requisitos: parte Android .....	41
3.2.3 Módulos: parte Android .....	43
3.2.4 Limitaciones: parte Android.....	44
<b>4 Desarrollo</b> .....	<b>45</b>
4.1 Aplicación web .....	45
4.1.1 Primeros pasos.....	45
4.1.2 Tipos de archivos.....	46
4.1.3 Desarrollo web.....	51
4.2 Aplicación Android .....	55
4.2.1 Primeros pasos.....	57
4.2.2 Conceptos Clave .....	61
4.2.3 Tipos de archivos.....	66
4.2.4 Desarrollo Android.....	70
<b>5 Integración, pruebas y resultados</b> .....	<b>74</b>
5.1 Aplicación web .....	74
5.2 Aplicación Android .....	91
<b>6 Conclusiones y trabajo futuro</b> .....	<b>99</b>
6.1 Conclusiones.....	99
6.2 Trabajo futuro .....	101
<b>Referencias</b> .....	<b>103</b>
<b>Glosario</b> .....	<b>105</b>
<b>Anexos</b> .....	<b>6-I</b>
A.    Manual de programas .....	6-I
B.    Presupuesto.....	V
C.    Pliego de condiciones .....	VII

# ÍNDICE DE FIGURAS

FIGURA 1 – LOGO EPS.....	0
FIGURA 2 – LOGO UAM.....	0
FIGURA 3 – HANDIE TALKIE H12-16.....	5
FIGURA 4 – MOTOROLA DYNATAC 8000X.....	5
FIGURA 5 – IBM SIMON PERSONAL COMMUNICATOR.....	6
FIGURA 6 – IPHONE.....	6
FIGURA 7 – IDIOMAS USADOS EN INTERNET.....	9
FIGURA 8 – USUARIOS DE INTERNET POR CONTINENTE.....	9
FIGURA 9 – ARQUITECTURA DE ANDROID.....	15
FIGURA 10 – VERSIONES DE ANDROID POR USUARIOS.....	18
FIGURA 11 – DISTRIBUCIÓN SO MÓVILES EN ESPAÑA.....	19
FIGURA 12 – DISTRIBUCIÓN SO MÓVILES ALEMANIA.....	20
FIGURA 13 – DISTRIBUCIÓN SO MÓVILES: EEUU vs JAPON.....	20
FIGURA 14 – SO MÓVILES EN EL MUNDO.....	21
FIGURA 15 – VENDEDORES DE SMARTPHONES EN EL MUNDO.....	22
FIGURA 16 – ICONO SEQUENTIAL CIRCUITS.....	25
FIGURA 17 – ICONO COMBINATIONAL CIRCUITS.....	26
FIGURA 18 – ICONO KARN MAP.....	27
FIGURA 19 – PÁGINA INICIO SIGMA UAM.....	28
FIGURA 20 – EXPEDIENTE SIGMA UAM.....	29
FIGURA 21 – CAMBIO CONTRASEÑA SIGMA UAM.....	30
FIGURA 22 – ENCUESTAS SIGMA UAM.....	30
FIGURA 23 – SOPORTE SIGMA UAM.....	31
FIGURA 24 – INICIO APLICACIÓN CUSTOM EXAM 1.....	32
FIGURA 25 – INICIO APLICACIÓN CUSTOM EXAM 2.....	32
FIGURA 26 – INICIO APLICACIÓN CUSTOM EXAM 3.....	32
FIGURA 27 – AJUSTES APLICACIÓN CUSTOM EXAM.....	33
FIGURA 28 – INICIO WEB CUSTOM EXAM.....	34
FIGURA 29 – CREAR EXAMEN WEB CUSTOM EXAM 1.....	35
FIGURA 30 – CREAR EXAMEN WEB CUSTOM EXAM 2.....	35
FIGURA 31 – EXÁMENES WEB CUSTOM EXAM 1.....	36
FIGURA 32 – EXÁMENES WEB CUSTOM EXAM 2.....	36
FIGURA 33 – RESULTADO EXAMEN CUSTOM EXAM.....	37
FIGURA 34 – DIAGRAMA FLUJO APLICACIÓN WEB.....	35
FIGURA 35 – DIAGRAMA FLUJO APLICACIÓN ANDROID.....	40
FIGURA 36 – ESTILO CSS.....	43
FIGURA 37 – CASOS DE USO MÓDULO ADMINISTRACIÓN.....	49
FIGURA 38 – CASOS DE USO MÓDULO PROFESORADO.....	49
FIGURA 39 – MÓDULO ENTIDAD RELACIÓN APLICACIÓN WEB.....	51
FIGURA 40 – MODELO RELACIONAL APLICACIÓN WEB.....	52
FIGURA 41 – PÁGINA INICIO ANDROID STUDIO.....	55
FIGURA 42 – ANDROID STUDIO SDK MANAGER.....	56
FIGURA 43 – ANDROID VIRTUAL DEVICE MANAGER.....	57
FIGURA 44 – ANDROID VIRTUAL DEVICE.....	58
FIGURA 45 – ANDROID RESOURCES.....	61
FIGURA 46 – DIAGRAMA CLASE ACTIVITY.....	62
FIGURA 47 – ESTRUCTURA PROYECTO ANDROID STUDIO.....	66
FIGURA 48 – CARPETA APP ANDROID STUDIO.....	66
FIGURA 49 – CARPETA JAVA ANDROID STUDIO.....	67
FIGURA 50 – CARPETA RES ANDROID STUDIO.....	68
FIGURA 51 – CARPETA DRAWABLE ANDROID STUDIO.....	68
FIGURA 52 – DENSIDADES PANTALLA ANDROID.....	69
FIGURA 53 – CASOS DE USO DEL MÓDULO DE ALUMNOS.....	70
FIGURA 54 – DIAGRAMA DE CLASES APLICACIÓN ANDROID.....	71
FIGURA 55 – PARSEO MEDIANTE JSON.....	72
FIGURA 56 – PÁGINA LOGIN APLICACIÓN WEB.....	75

FIGURA 57 – PÁGINA ERROR APLICACIÓN WEB.....	75
FIGURA 58 – PÁGINA LOGIN CON CHECK DE ADMIN APLICACIÓN WEB.....	76
FIGURA 59 – PÁGINA PRINCIPAL ADMINISTRADOR APLICACIÓN WEB.....	77
FIGURA 60 – CUADRO SESIÓN APLICACIÓN WEB.....	77
FIGURA 61 – MENÚ PROFESORES MÓDULO ADMINISTRACIÓN APLICACIÓN WEB.....	78
FIGURA 62 – ELIMINAR PROFESOR DEL MÓDULO DE ADMINISTRACIÓN APLICACIÓN WEB.....	78
FIGURA 63 – MENÚ PROFESORES MÓDULO ADMINISTRACIÓN APLICACIÓN WEB.....	79
FIGURA 64 – CONTROL ERRORES ELIMINAR PROFESOR APLICACIÓN WEB.....	79
FIGURA 65 – FORMULARIO INSERTAR PROFESOR APLICACIÓN WEB.....	80
FIGURA 66 – CONTROL ERRORES FORMULARIO INSERTAR PROFESOR APLICACIÓN WEB.....	81
FIGURA 67 – REGISTRO PROFESOR APLICACIÓN WEB.....	81
FIGURA 68 – NUEVO PROFESOR APLICACIÓN WEB.....	82
FIGURA 69 – ASIGNATURAS APLICACIÓN WEB.....	83
FIGURA 70 – INSERTAR ASIGNATURA APLICACIÓN WEB.....	83
FIGURA 71 – SELECCIONAR PROFESOR NUEVA ASIGNATURA APLICACIÓN WEB.....	84
FIGURA 72 – ASIGNATURA REGISTRADA WEB.....	84
FIGURA 73 – LISTADO ASIGNATURAS APLICACIÓN WEB.....	85
FIGURA 74 – ALUMNOS APLICACIÓN WEB.....	85
FIGURA 75 – ACCESO MÓDULO PROFESORES APLICACIÓN WEB.....	86
FIGURA 76 – ASIGNATURAS POR PROFESOR APLICACIÓN WEB.....	86
FIGURA 77 – MENSAJES APLICACIÓN WEB.....	87
FIGURA 78 – CONSULTAR NOTAS APLICACIÓN WEB.....	88
FIGURA 79 – FORMULARIO CREAR EXAMEN APLICACIÓN WEB.....	88
FIGURA 80 – FORMULARIO CREAR EXAMEN RELLENO APLICACIÓN WEB.....	89
FIGURA 81 – PREGUNTAS Y RESPUESTAS EXAMEN APLICACIÓN WEB.....	90
FIGURA 82 – ACCESO APLICACIÓN ANDROID 1.....	91
FIGURA 83 – ACCESO APLICACIÓN ANDROID 2.....	91
FIGURA 84 – MENÚ PRINCIPAL APLICACIÓN ANDROID.....	92
FIGURA 85 – ASIGNATURAS MATRICULADO ANDROID.....	92
FIGURA 86 – EXÁMENES DIE ANDROID.....	93
FIGURA 87 – SIN CALIFICACIÓN DIE ANDROID.....	93
FIGURA 88 – EXÁMENES BBDD ANDROID.....	94
FIGURA 89 – CALIFICACIÓN EXAMEN BBDD ANDROID.....	94
FIGURA 90 – MENÚ MENSAJES ANDROID.....	95
FIGURA 91 – PROFESORES MENSAJES ANDROID.....	95
FIGURA 92 – ENVIAR MENSAJE ANDROID.....	96
FIGURA 93 – MENSAJES ENVIADOS ANDROID.....	96
FIGURA 94 – EXAMEN NO ACTIVADO ANDROID.....	97
FIGURA 95 – PREGUNTA EXAMEN ANDROID.....	97
FIGURA 96 – PREGUNTA SELECCIONADA EXAMEN ANDROID.....	98
FIGURA 97 – CALIFICACIÓN EXAMEN ANDROID.....	98
FIGURA 98 – EJEMPLO DE NOTEPAD++.....	I
FIGURA 99 – MENÚ DE WAMPSEVER.....	II
FIGURA 100 – EJEMPLO DE FILEZILLA.....	III
FIGURA 101 – MENÚ CUENTA SERVIDOR WEB HOSTINGER.....	IV
FIGURA 102 – EJEMPLO DE ANDROID STUDIO.....	V

## ÍNDICE DE TABLAS

TABLA 1 – USUARIOS DE INTERNET POR PAÍS.....	10
TABLA 2 – VERSIONES DE ANDROID.....	17
TABLA 3 – SO MÓVILES EN EL MUNDO.....	21
TABLA 4 – VENDEDORES DE SMARTPHONES EN EL MUNDO.....	22

# 1 Introducción

---

## 1.1 Motivación

A lo largo de los últimos años, los teléfonos móviles se han convertido en unos dispositivos imprescindibles, superando incluso en uso a los ordenadores. Hoy en día, estos dispositivos se han extendido tanto que, prácticamente todo el mundo dispone de uno de ellos. Por ello, dado el auge que han experimentado los teléfonos inteligentes y las tabletas en toda la sociedad, desde el principio se consideró muy interesante explotar este ámbito tecnológico.

Este proyecto surge con el objetivo de aprovechar las ventajas que ofrecen estos dispositivos inteligentes en los estudios universitarios. Asimismo, con la idea de otorgar un valor añadido a la Universidad mediante la realización del PFC, se identificó la posibilidad de desarrollar una funcionalidad todavía no presente en los servicios ofrecidos por Sigma.

Por lo tanto, para evitar a los profesores emplear tiempo corrigiendo exámenes y aprovechando la existencia y proliferación de exámenes de respuesta múltiple, se decidió implementar un Gestor de Exámenes y Calificaciones. De esta forma, se desarrollará este proyecto como piloto para una posible implantación de dicha aplicación en la Universidad.

Por otro lado, y no menos importante, para la elección de este proyecto ha sido fundamental la posibilidad de completar los conocimientos académicos con respecto a materias informáticas de lenguajes como Android, HTML, CSS, PHP, etc.

A continuación, se destacarán algunas ventajas que ofrecen los dispositivos portables tales como smartphones y tablets, para los cuales se contempla este proyecto:

- Cálculo instantáneo de la nota automáticamente al finalizar el examen.
- Comodidad de uso.
- Eliminación del uso de papel y bolígrafo.
- Facilidad de introducir datos a través de la pantalla táctil, otorgando sencillez también a la edición de respuestas.
- Mayor protección intelectual frente a plagios.
- Portabilidad. Los terminales móviles y tabletas ocupan bastante menos espacio que un ordenador.
- Posibilidad de modificación y redistribución automática a través de Google Play.

## **1.2 Objetivos**

Este proyecto surge con la idea de aprovechar las ventajas que ofrecen los dispositivos inteligentes en los estudios universitarios. El objetivo principal de este proyecto es desarrollar una aplicación gratuita que permita la gestión y realización de exámenes de tipo test por parte de los alumnos en dispositivos Android. Esta aplicación nace como proyecto piloto de cara a una posible implantación en esta u otras universidades o centros educativos de un gestor de exámenes de tipo test.

La aplicación ofrece la funcionalidad de que los profesores creen exámenes y consulten las calificaciones que obtuvieron los alumnos en los exámenes de tipo test que realizaron desde un dispositivo Android. Todo ello, controlado por Administración, encargada de asignar profesores y alumnos a cada asignatura. Además, también está contemplada la comunicación entre alumnos y profesores.

Desde el punto de vista del estudiante, el objetivo del presente PFC es, aparte de generar valor añadido a la Universidad con las funcionalidades que ofrece la aplicación desarrollada, completar la formación académica en temas de programación en Android, aspectos de BBDD y programación web. Ninguna de las materias anteriores están cubiertas por asignaturas obligatorias de la carrera, por lo que constituyen una formación adicional.

## **1.3 Organización de la memoria**

La memoria consta de los siguientes capítulos:

- **Capítulo 1:** Introducción: Motivación, Objetivos del Proyecto y Organización de la Memoria.
- **Capítulo 2:** Estado del Arte (Evolución de los dispositivos móviles, Sistemas Operativos móviles y Aplicaciones similares).
- **Capítulo 3:** Diseño de la aplicación (parte web y parte Android).
- **Capítulo 4:** Desarrollo (parte web y parte Android).
- **Capítulo 5:** Integración, pruebas y resultados.
- **Capítulo 6:** Conclusiones y trabajo futuro.

1.



## 2 Estado del arte

---

En este apartado se llevará a cabo una panorámica de los dispositivos móviles, abarcando desde su evolución a lo largo de la historia hasta los principales sistemas operativos móviles existentes en el mercado. Por supuesto, el sistema operativo Android se tratará con mayor detalle puesto que es sobre el que se fundamenta la aplicación. Asimismo, se analizarán aplicaciones similares a la que se desarrolla en este PFC.

### 2.1 Historia de los terminales móviles

En primer lugar, cabe señalar que entendemos como telefonía móvil a la comunicación a distancia entre terminales no conectados físicamente. Al no existir cables entre un terminal y otro, el medio de transmisión es el aire, transmitiéndose la información a través de ondas electromagnéticas. En principio, surgieron para facilitar la transmisión de voz, sin embargo, actualmente han ido adquiriendo cada vez mayor valor añadido y también soportan la demanda de imágenes, vídeos y acceso a internet.

Como se ha estudiado en la asignatura de Comunicaciones Móviles, el servicio de telefonía móvil se basa en una red de celdas (teóricamente hexagonales) donde, en el centro de ellas, se encuentra una antena repetidora de señal llamada estación base. Es por este motivo que, en países latinoamericanos, se denomina “celular” al teléfono móvil.

El primer teléfono móvil tal y como lo conocemos hoy, fue puesto a la venta en el año 1984 por la compañía Motorola. El denominado *DynaTAC 8000x* pesaba 800 gramos, costaba 4.000 USD y su batería contaba con autonomía para una hora de conversación. No obstante, ya en los inicios de la Segunda Guerra Mundial, Motorola desarrolló un equipo que permitía la comunicación a distancia sin cables, tan necesaria en aquella época. El llamado *Handie Talkie H12-16* permitía el contacto a distancia mediante ondas de radio.



Figura 3: Handie Talkie H12-16



Figura 4: DynaTAC 8000x

En cuanto al primer Smartphone de la historia, aunque por todos es conocido que Apple revolucionó el concepto en 2007 con la comercialización del iPhone, el primer teléfono inteligente se remonta a años atrás. En 1994, IBM puso a la venta Simon Personal Communicator, capaz de combinar las funcionalidades de un teléfono móvil con una PDA a través de su pantalla táctil.



*Figura 5: Primer Smartphone*



*Figura 6: Primer iPhone*

Por tanto, la aparición en los años 80 de los primeros terminales móviles dio comienzo a la denominada Primera Generación de Teléfonos Móviles (1G), donde los dispositivos operaban mediante ondas de radio ofreciendo una calidad de voz relativa y capacidad de transmisión de datos baja (del orden de kbps). Esta generación se caracterizaba por ser analógica y estar enfocada únicamente para la transmisión de voz.

Ya en los años 90 surge la denominada Segunda Generación (2G), motivada por la aparición del protocolo GSM (Global System for Mobile Communications), lo que supuso el paso de la telefonía analógica hacia la digital. Este protocolo permitía interconectar redes y conectarse a ellas con un mismo terminal, apareciendo el primer concepto de roaming. Con la llegada del 2G los servicios de telefonía móviles se volvieron populares. Asimismo, se consiguió mejorar la calidad de voz, aumentar la velocidad de transmisión de datos (a 56 kbps) y se crearon los famosos SMS.

Acto seguido, gracias a la implantación de la tecnología GPRS (General Packet Radio Service), se siguieron mejorando los servicios de la telefonía móvil, permitiendo incluso el intercambio de imágenes (MMS) y la posibilidad de navegar por Internet en páginas a color. Este avance se conoce como la Generación 2.5G.

La siguiente generación fue la 3G. Surgió debido a la aparición del UMTS (Universal Mobile Telecommunication System), sistema de acceso múltiple por división de código de banda ancha. Esta generación mejora la potencia de las antenas, permitiendo más conexiones, así como una mayor calidad de voz y mayor velocidad para la transferencia de datos, llegando a velocidades de Mbps. El 3G motivó la aparición de aplicaciones de audio, imágenes y comunicaciones con vídeo en tiempo real.

La evolución ha continuado, dando paso al 4G actual, donde se sigue mejorando la calidad de las antenas en cuanto a su capacidad, cobertura y calidad de señal. Todo ello radica en una mejora de la velocidad de transferencia de datos, capaz de alcanzar los 20 Mbps, permitiendo así la reproducción de vídeo en tiempo real con buena calidad. Esta generación se basa en el protocolo IP.

A pesar de que todavía se está implantando la tecnología 4G, ya se ha empezado a trabajar en la nueva generación, 5G. Esta tecnología, aún sin estandarizar, permitirá alcanzar velocidades de Gbps.

Por lo tanto, como se ha podido comprobar estudiando la historia de los terminales móviles, su evolución ha sido muy rápida. Tanto es así, que, a finales de 2014, existían en el mundo casi tantos teléfonos móviles como personas, lo que supone una penetración de 95,5 líneas por cada 100 habitantes. El avance de la demanda de servicio TIC sigue al alza, no en vano según el informe de 2013 “La Sociedad en Red”, elaborado por el Observatorio Nacional de las Telecomunicaciones y de la Sociedad de la Información (ONTSI), la penetración de la telefonía móvil ha llegado al 96% en España. En lo que respecta a penetración de smartphones, España se encuentra muy por encima de la media europea, contando con un 81% de teléfonos inteligentes con respecto al total de dispositivos móviles.

Los síntomas de este progreso tecnológico muestran también que más del 80% de los usuarios de Internet en España acceden a través de un dispositivo móvil. Asimismo, esta tendencia alcista en cuanto al uso de dispositivos móviles en la sociedad, también se corrobora en el auge de las tablets, las cuales han experimentado un crecimiento de un 68% en cuanto a su venta con respecto a 2013.

*- Los datos anteriores han sido extraídos del informe de “Sociedad de la Información” elaborado por Telefónica para el ejercicio 2014 -*

## **2.2 Internet**

### **2.2.1 Historia de Internet**

Ya que la aplicación que se desarrolla en este PFC cuenta con una parte de acceso web, se llevará a cabo una breve introducción sobre la historia de Internet, que ha revolucionado la informática, las comunicaciones y en definitiva, nuestra vida.

Tal y como lo conocemos hoy, Internet es el conjunto de las redes de comunicación interconectadas que emplean la familia de protocolos TCP/IP, permitiendo a las redes físicas que la componen dar lugar a una única red lógica de alcance global.

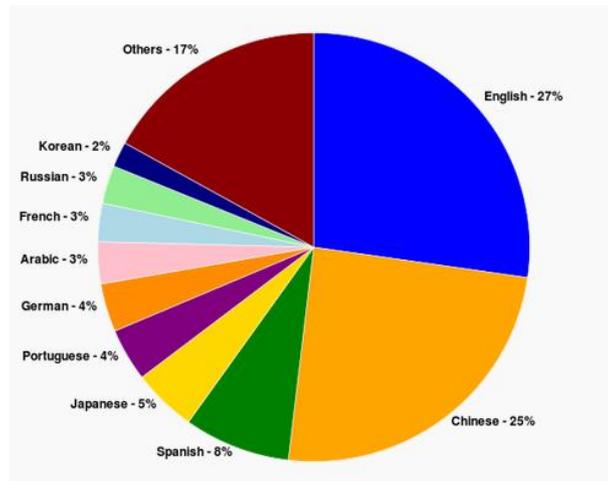
Los orígenes de Internet se remontan a los años 60. Surgieron en ARPA, hoy DARPA (Defense Advanced Research Projects Agency) como respuesta a la necesidad de esta organización de buscar formas para potenciar y maximizar el uso de los computadores de aquella época. En sus inicios, se conoció como ARPANet (Advanced Research Projects Agency Network), que era una red inicial de comunicaciones de alta velocidad a la cual fueron integrándose poco a poco otras instituciones gubernamentales y académicas durante los años posteriores, beneficiándose de la posibilidad de comunicarse entre ellas. La primera red interconectada surgió a finales de 1969, creándose un enlace entre las universidades de UCLA y Stanford empleando la línea telefónica conmutada.

Fue en 1972 cuando se llevó a cabo la primera demostración pública de ARPANET. Una vez mostrado que esta interconexión de redes que operaban sobre la red telefónica conmutada funcionaba, se comenzaron a desarrollar protocolos e comunicaciones que permitieran un intercambio de información de forma “transparente” para las computadoras conectadas. No obstante, no será hasta 1983 cuando se cambie el protocolo NCP por el TCP/IP actual y hasta 1989 cuando se integren los protocolos OSI en la arquitectura de Internet.

Ya en 1990, científicos del CERN (Consejo Europeo para la Investigación Nuclear) crearon lo que se conoció como WorldWideWeb, que supuso la creación del código HTML y con él, del primer cliente web. Por tanto, WWW fue el primer navegador web y editor de páginas web. En 1993, el CERN liberó el código fuente de WorldWideWeb bajo el dominio público, convirtiéndolo en software libre.

No es trivial establecer el tamaño exacto de Internet, dado que está en continuo crecimiento y no existe una manera fiable de acceder a su contenido. No obstante, a continuación se incluirán algunas estadísticas y datos interesantes sobre Internet.

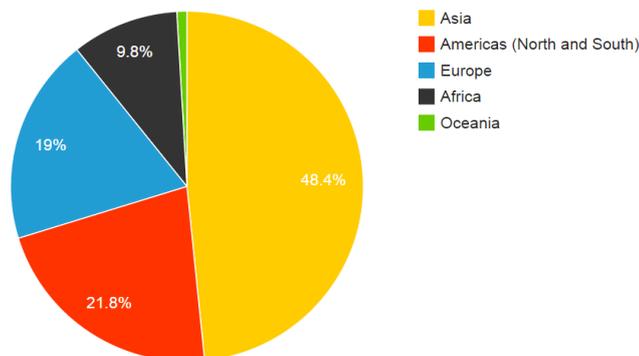
La distribución de idiomas empleados en Internet es la siguiente:



*Figura 7: Idiomas usados en Internet*

A pesar de que el castellano solo sea el tercer idioma más empleado en Internet, la aplicación se desarrollará en este idioma, dando lugar a una primera versión. Para versiones posteriores, se contempla la inclusión de más idiomas con el objetivo de dotar a la aplicación de un enfoque global, pudiendo ser utilizada por más usuarios.

Alrededor del 41% de la población mundial, es decir, más de 3.000 millones de personas, dispone de acceso a Internet. La distribución de usuarios de Internet por continente es acorde al número de población de cada uno:



*Figura 8: Usuarios de Internet por continente*

La distribución de usuarios con acceso a Internet por países es la siguiente:

Rank	Country	Internet Users	1 Year Growth %	1 Year User Growth	Total Country Population	1 Yr Population Change (%)	Penetration (% of Pop. with Internet)	Country's share of World Population	Country's share of World Internet Users
1	<a href="#">China</a>	641,601,070	4%	24,021,070	1,393,783,836	0.59%	46.03%	19.24%	21.97%
2	<a href="#">United States</a>	279,834,232	7%	17,754,869	322,583,006	0.79%	86.75%	4.45%	9.58%
3	<a href="#">India</a>	243,198,922	14%	29,859,598	1,267,401,849	1.22%	19.19%	17.50%	8.33%
4	<a href="#">Japan</a>	109,252,912	8%	7,668,535	126,999,808	-0.11%	86.03%	1.75%	3.74%
5	<a href="#">Brazil</a>	107,822,831	7%	6,884,333	202,033,670	0.83%	53.37%	2.79%	3.69%
6	<a href="#">Russia</a>	84,437,793	10%	7,494,536	142,467,651	-0.26%	59.27%	1.97%	2.89%
7	<a href="#">Germany</a>	71,727,551	2%	1,525,829	82,652,256	-0.09%	86.78%	1.14%	2.46%
8	<a href="#">Nigeria</a>	67,101,452	16%	9,365,590	178,516,904	2.82%	37.59%	2.46%	2.30%
9	<a href="#">United Kingdom</a>	57,075,826	3%	1,574,653	63,489,234	0.56%	89.90%	0.88%	1.95%
10	<a href="#">France</a>	55,429,382	3%	1,521,369	64,641,279	0.54%	85.75%	0.89%	1.90%
11	<a href="#">Mexico</a>	50,923,060	7%	3,423,153	123,799,215	1.20%	41.13%	1.71%	1.74%
12	South Korea	45,314,248	8%	3,440,213	49,512,026	0.51%	91.52%	0.68%	1.55%
13	<a href="#">Indonesia</a>	42,258,824	9%	3,468,057	252,812,245	1.18%	16.72%	3.49%	1.45%
14	<a href="#">Egypt</a>	40,311,562	10%	3,748,271	83,386,739	1.62%	48.34%	1.15%	1.38%
15	<a href="#">Viet Nam</a>	39,772,424	9%	3,180,007	92,547,959	0.95%	42.97%	1.28%	1.36%
16	<a href="#">Philippines</a>	39,470,845	10%	3,435,654	100,096,496	1.73%	39.43%	1.38%	1.35%
17	<a href="#">Italy</a>	36,593,969	2%	857,489	61,070,224	0.13%	59.92%	0.84%	1.25%
18	<a href="#">Turkey</a>	35,358,888	3%	1,195,610	75,837,020	1.21%	46.62%	1.05%	1.21%
19	<a href="#">Spain</a>	35,010,273	3%	876,986	47,066,402	0.30%	74.38%	0.65%	1.20%

*Tabla 1: Usuarios de Internet por país*

Según informe de “Sociedad de la Información” elaborado por Telefónica para el ejercicio 2014, más de 26 millones de españoles acceden regularmente a Internet.

- Los datos anteriores han sido extraídos [www.internetlivestats.com](http://www.internetlivestats.com) -

## 2.2.2 Otros elementos empleados en la parte web

### Página Web

Una página web es un documento electrónico capaz de contener texto, sonido, vídeos, programas, enlaces, etc. Se accede a ellas mediante un navegador. La información de las páginas webs se encuentra normalmente en formato HTML o XHTML. Las páginas web, pueden estar almacenadas en un equipo local o en un servidor web remoto (se tratará más adelante). El acceso a las páginas web se lleva a cabo mediante una transferencia desde servidores, empleando el protocolo de transferencia de hipertexto (HTTP).

- Estáticas: el funcionamiento es el siguiente: al introducir una url en el navegador escogido, el navegador realiza un pedido al servidor web. Entonces, el servidor busca el archivo solicitado (normalmente archivos HTML) y si lo encuentra, lo devuelve al navegador.

- Dinámicas: la mayoría de sitios web son dinámicos. En ellos, el servidor debe llevar a cabo funciones más allá de las de una web estática.

### Servidores Web

Son programas informáticos que procesan las peticiones del cliente. El código recibido por el cliente suele ser compilado y ejecutado en un navegador web. Para la transmisión de los datos, se suele emplear un protocolo (normalmente HTTP).

El servidor web se ejecuta en un ordenador, manteniéndose a la espera de peticiones por parte del cliente (navegador web), respondiendo a estas peticiones mediante una página web que se exhibirá en el navegador o mostrando un error si sucedió. El cliente será el encargado de interpretar el código HTML que recibe, es decir, de mostrar las fuentes, colores, disposición de textos y objetos, etc. El servidor se limita a transferir el código de la página sin llevar a cabo ninguna interpretación de la misma. Aparte de código HTML, los servidores web también pueden entregar aplicaciones web.

### Bases de Datos

Se denomina Base de Datos a los bancos de información que contienen un conjunto de datos pertenecientes a un mismo contexto, almacenados para su uso o consulta.

Las BBDD se pueden clasificar según su contenido. Asimismo, se pueden dividir de la siguiente manera:

Estáticas: Bases de datos únicamente de lectura, utilizadas para almacenar datos históricos.

Dinámicas: la información almacenada en estas BBDD se modifica en el tiempo, permitiendo operaciones como el borrado y la consulta y edición de datos.

Además, también se pueden clasificar de acuerdo a su modelo de administración de datos. Algunos de los modelos de BBDD más frecuentes son los siguientes:

**Bases de datos jerárquicas**: en este modelo los datos se organizan en forma de árbol invertido, donde un nodo padre puede tener varios hijos.

**Bases de datos de red**: su diferencia principal con respecto al modelo jerárquico es permitir que un mismo nodo tenga varios padres.

**Bases de datos transaccionales**: BBDD cuyo fin es el envío y recepción de datos a grandes velocidades.

**Bases de datos relacionales**: en este modelo, el lugar y la forma en que se almacenan los datos no tiene relevancia. La información puede ser recuperada o almacenada mediante queries (consultas) que ofrecen una amplia flexibilidad para administrar la información. El lenguaje más habitual para generar las consultas a estas bases de datos es SQL.

**Bases de datos orientadas a objetos**: modelo relativamente reciente propio de los modelos informáticos orientados a objetos. Tratan de almacenar en la base de datos los objetos completos (estado y comportamiento). Incorporan conceptos del paradigma de objetos tales como la encapsulación, la herencia o el polimorfismo.

## **Entorno de Desarrollo Integrado (IDE)**

Se trata de una aplicación software que incluye servicios para facilitar el desarrollo de software al programador. Normalmente, un IDE consiste en un editor de código fuente, herramientas de construcción automáticas y un depurador de código.

Algunos IDEs poseen un compilador, un intérprete o ambos, como tienen IDEs muy extendidos como son NetBeans y Eclipse. Algunos IDEs están dedicados específicamente a un lenguaje de programación, sin embargo, la mayoría abarcan la posibilidad de programar en múltiples lenguaje bajo el mismo programa.

## 2.3 Sistemas Operativos de dispositivos móviles

### 2.3.1 Android

#### Introducción

Sistema operativo móvil de código abierto basado en el kernel de Linux presentado en 2007. El hecho de que sea de código abierto implica que cualquier desarrollador puede crear aplicaciones y compilarlas a código nativo de ARM (API de Android); asimismo, cualquiera pueda acceder al código fuente de Android y modificarlo a su gusto. Se trata del sistema operativo más utilizado en smartphones y tablets. A pesar de que fue desarrollado por la compañía Android Inc., ahora es propiedad de Google, ya que esta última adquirió Android Inc. en 2005.

El primer smartphone lanzado al mercado con sistema operativo Android fue el HTC Dream, en 2008. Al ser de código abierto, este sistema operativo cuenta con una gran comunidad de desarrolladores. Los programas están escritos en el lenguaje de programación Java. La tienda oficial para la descarga y venta de aplicaciones de Android es Google Play, no obstante existen otras como Samsung Apps o F-Droid. Dada la facilidad para programar y los menores gastos para publicar aplicaciones, existen más de un millón de aplicaciones Android.

#### Características

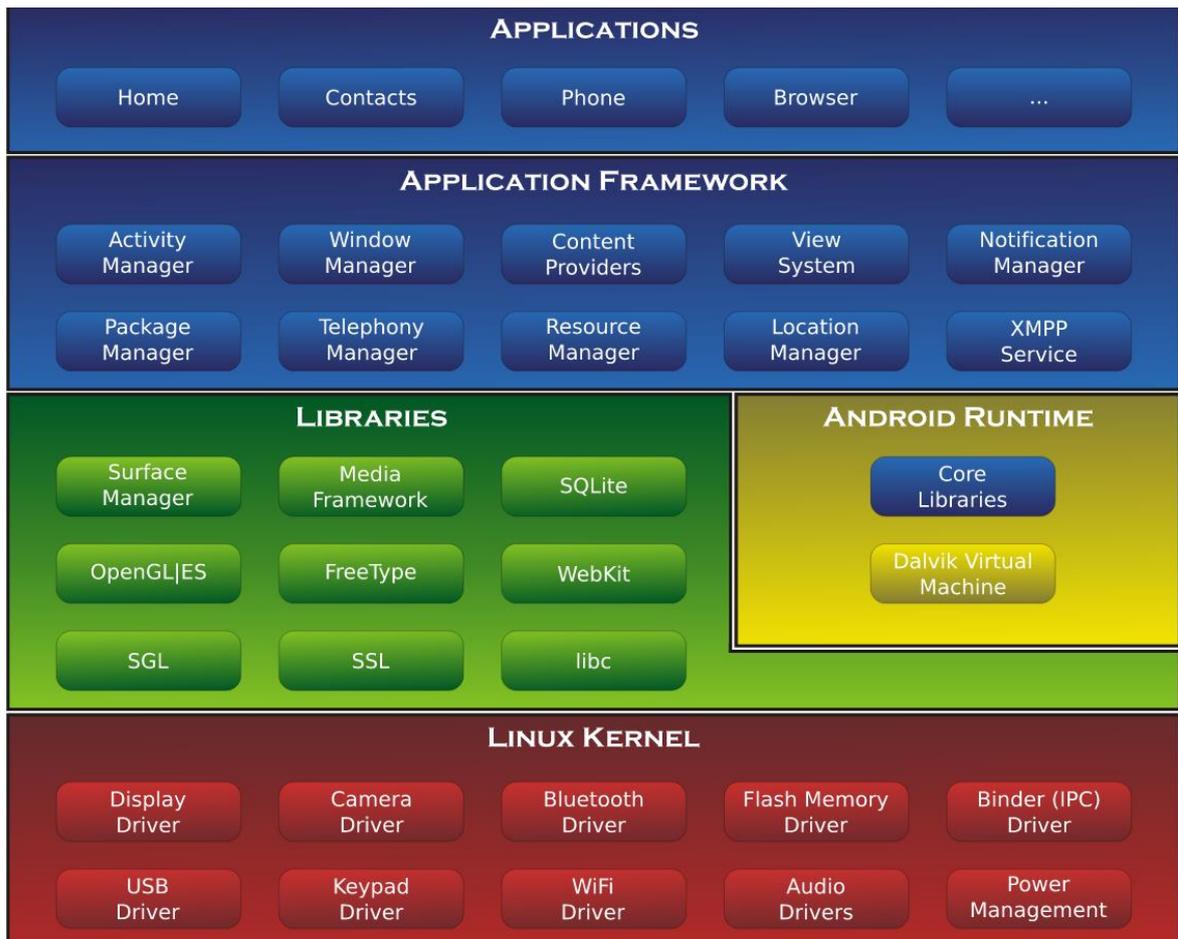
En este apartado se enumeran algunas de las principales características con las que cuentan actualmente los dispositivos Android:

- **Almacenamiento:** Para el almacenamiento de datos, se emplea la base de datos SQLite.
- **Conectividad:** Android soporta múltiples tecnologías de conectividad como GSM/EDGE, Bluetooth, CDMA, UMTS, Wi-Fi, LTE, WIMAX, GPRS, HSDPA, etc.
- **Mensajería:** soporta SMS y MMS.
- **Navegador web:** el navegador está basado en el motor WebKit.
- **Soporte de Java:** a pesar de que la mayoría de aplicaciones están escritas en Java, no existe una máquina virtual de Java. Para ello, primero se compila en un ejecutable Dalvik y corre en la Máquina Virtual Dalvik, desarrollada por Google.
- **Soporte multimedia:** Android soporta múltiples formatos multimedia como MP4, MP3, MIDI, JPEG, PNG, GIF, BMP, etc. También tiene soporte para streaming.

- **Soporte para hardware adicional:** Android soporta accesorios como cámaras de fotos, de vídeo, pantallas táctiles, GPS, sensores de luz o proximidad, termómetros, etc.
- **Entorno de desarrollo:** incluye un emulador de dispositivos, herramientas para la depuración de memoria y análisis del rendimiento del software. En los inicios, se empleaba Eclipse como IDE (Entorno de Desarrollo Integrado) con el plugin de ADT. Actualmente, se considera el software Android Studio (empleado en este proyecto) como entorno oficial.
- **Multi-táctil:** Android dispone de soporte nativo para pantallas capacitivas con soporte multi-táctil.
- **Bluetooth:** cuenta con soporte bluetooth desde la versión 1.5.
- **Multitarea:** las aplicaciones que no se ejecutan en primer plano, reciben ciclos de reloj.
- **Videollamada:** Android soporta videollamada a través de Hangouts (antes Google Talk).
- **Tethering:** se permite al dispositivo ser usado como “hot spot” (centro de acceso inalámbrico).

## Arquitectura

La arquitectura interna de Android se forma básicamente por cuatro capas. Al igual que en el modelo OSI, cada una de ellas emplea servicios de las capas anteriores y ofrece servicios a las capas siguientes. Abajo, se muestra un diagrama explicativo de las distintas capas de la plataforma Android:



*Figura 9: Arquitectura de Android*

A continuación, se explicará la funcionalidad de cada una de las capas más en detalle:

**Aplicaciones:** este nivel está formado por el conjunto de aplicaciones instaladas en el dispositivo. Todas ellas, han de ejecutarse en la máquina virtual Dalvik para garantizar la seguridad del sistema. Normalmente, las aplicaciones Android están escritas en Java, para lo cual se suele utilizar Android SDK (Software Development Kit). Si se quieren desarrollar aplicaciones utilizando C o C++, se suele emplear el Android NDK (Native Development Kit). Todas las aplicaciones usan los servicios, API y librerías de las capas anteriores.

**Entorno de Aplicación:** proporciona una plataforma de desarrollo libre de aplicaciones. Esta capa se ha diseñado para simplificar la reutilización de componentes. Los servicios más importantes que incluye son los siguientes:

- Views: extenso conjunto de vistas (parte visual de los componentes).
- Resource Manager: proporciona acceso a recursos que no son código.
- Activity Manager: maneja el ciclo de vida de las aplicaciones y proporciona un sistema de navegación entre ellas.
- Notification Manager: permite a las aplicaciones mostrar alertas personalizadas.
- Content Providers: mecanismo para acceder a datos de otras aplicaciones.

**Bibliotecas:** conjunto de librerías en C/C++ usadas en varios componentes de Android. Están compiladas en código nativo del procesador. Algunas de estas librerías son:

System C library: derivación de la librería BSD de C estándar (libc) adaptada para dispositivos embebidos basados en Linux.

Media Framework: librería que soporta códecs de reproducción y grabación de multitud de formatos de audio, vídeo e imágenes.

Surface Manager: maneja el acceso al subsistema de representación gráfica en 2D y 3D.

WebKit: soporta un moderno navegador web utilizado en el navegador Android. Se trata de la misma librería que utiliza Google Chrome y Safari de Apple.

SGL: motor de gráficos 2D.

Librerías 3D

FreeType: fuentes en bitmap y renderizado vectorial.

SQLite: potente y ligero motor de bases de datos relacionales.

SSL: proporciona servicios de encriptación Secure Socker Layer.

**Runtime de Android:** basado en el concepto de máquina virtual utilizado en Java. Dadas las limitaciones de los dispositivos donde se ejecuta Android (poca memoria y procesador limitado), no fue posible utilizar una máquina virtual Java estándar, para lo que se creó la máquina virtual Dalvik. No obstante, a partir de la versión 5.0 de Android, se reemplaza Dalvik por ART, que consigue reducir el tiempo de ejecución del código Java hasta en un 33%. También se incluye en el Runtime de Android el “core de librerías” con la mayoría de librerías disponibles en el lenguaje Java.

**Kernel de Linux:** Android emplea el núcleo 2.6 de Linux como capa de abstracción entre el hardware y el resto de la pila, por lo tanto, es la única que depende del hardware. Como se observa en la figura anterior, esta capa posee los drivers necesarios para que se puedan emplear múltiples componentes hardware en conjunción con el Smartphone o Tablet. Asimismo, esta capa proporciona otros servicios como la seguridad, multiproceso, pila de protocolos o maneja de la memoria.

## Versiones

Es curioso el hecho de que las versiones de Android reciban el nombre de diferentes dulces (en inglés). De hecho, en cada versión, el dulce elegido empieza por una letra, distinta, conforme a un orden alfabético:

A: Apple Pie (v1.0): Tarta de manzana

B: Banana Bread (v1.1): Pan de plátano

C: Cupcake (v1.5): Cupcake

D: Donut (v1.6): Donut

E: Éclair (v2.0/v2.1): Pepito

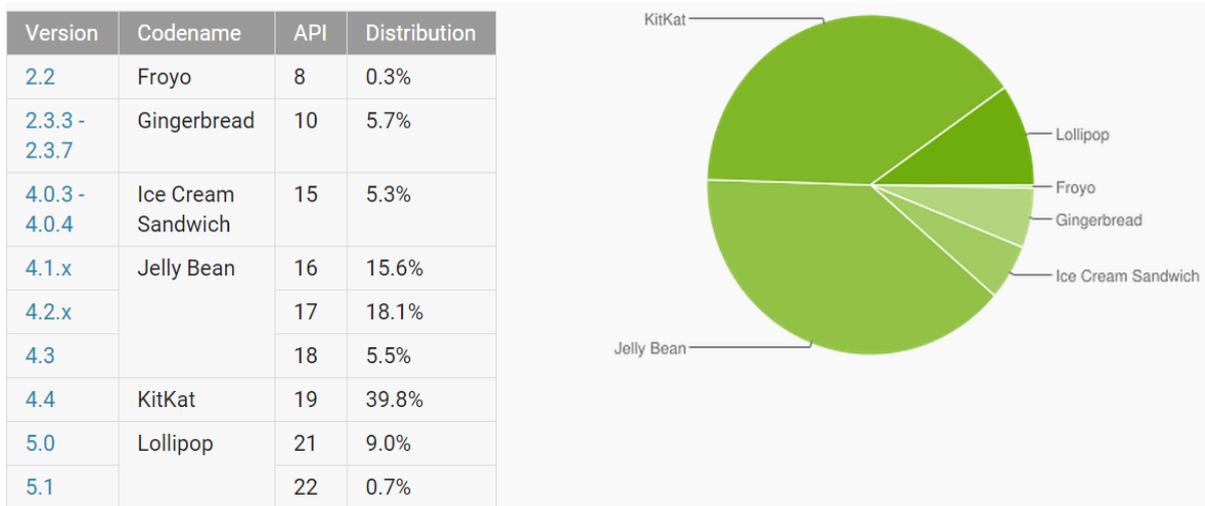
F: Froyo (v2.2): Yogur helado  
 G: Gingerbread (v2.3): Pan de jengibre  
 H: Honeycomb (v3.0/v3.1/v3.2): Panal de miel  
 I: Ice Cream Sandwich (v4.0): Sandwich de helado  
 J: Jelly Bean (v4.1/v4.2/v4.2): Gragea  
 K: KitKat (v4.4): Kit Kat  
 L: Lollipop (v5.0/v5.1): Piruleta

Debido a la eliminación de errores e inclusión de nuevas funcionalidades, han existido múltiples versiones de Android a lo largo de los años. La lista completa de versiones con nombre, número y nivel de API es la siguiente:

Code name	Version	API level
Lollipop	5.1	API level 22
Lollipop	5.0	API level 21
KitKat	4.4 - 4.4.4	API level 19
Jelly Bean	4.3.x	API level 18
Jelly Bean	4.2.x	API level 17
Jelly Bean	4.1.x	API level 16
Ice Cream Sandwich	4.0.3 - 4.0.4	API level 15, NDK 8
Ice Cream Sandwich	4.0.1 - 4.0.2	API level 14, NDK 7
Honeycomb	3.2.x	API level 13
Honeycomb	3.1	API level 12, NDK 6
Honeycomb	3.0	API level 11
Gingerbread	2.3.3 - 2.3.7	API level 10
Gingerbread	2.3 - 2.3.2	API level 9, NDK 5
Froyo	2.2.x	API level 8, NDK 4
Eclair	2.1	API level 7, NDK 3
Eclair	2.0.1	API level 6
Eclair	2.0	API level 5
Donut	1.6	API level 4, NDK 2
Cupcake	1.5	API level 3, NDK 1
(no code name)	1.1	API level 2
(no code name)	1.0	API level 1

*Tabla 2: Versiones de Android*

A continuación, se incluyen estadísticas del primer cuatrimestre de 2015 que muestran la distribución de los usuarios de Android con respecto a las versiones actuales.



*Figura 10: Versiones de Android por usuarios*

Se puede observar que, dada su reciente aparición, la última versión (Lollipop) todavía no ha conseguido el mayor porcentaje de usuarios, siguiendo a la cabeza las versiones 4.

*- Los datos anteriores han sido extraídos de la web [developer.android.com](http://developer.android.com)”-*

## 2.3.2 Otros Sistemas Operativos

### iOS

Es el sistema operativo de código cerrado desarrollado por Apple Inc. optimizado para los dispositivos móviles de dicha compañía, como son los iPhones, iPods y iPads. Está basado en una variante del kernel de Mac. Se trata del segundo sistema operativo móvil más extendido en todo el mundo.

### Windows Phone

Sistema operativo móvil compacto desarrollado por Microsoft como sucesor de Windows Mobile. A diferencia de su predecesor, está enfocado en el mercado de consumo en lugar de en el mercado empresarial. La interfaz de usuario integra varios de los servicios propios de Windows como OneDrive, Skype o Xbox Live. Entre los fabricantes que han lanzado móviles con Windows Phone se encuentran Nokia, HTC, Hacer, Alcatel o Toshiba.

### Blackberry

Sistema operativo móvil de código cerrado desarrollado por Blackberry, antigua Research in Motion (RIM) para los smartphones Blackberry. Lideraron el mercado entre 2008 y 2011.

### Otros

**Firefox OS** es un sistema operativo móvil relativamente nuevo basado en HTML5 con núcleo Linux, de código abierto. Ha sido desarrollado por Mozilla Corporation. Está enfocado sobre todo a dispositivos móviles de los llamados de gama baja.

**Symbian** surgió de una alianza de empresas como fueron Nokia, Sony Ericsson o Siemens, entre otras. Tuvo bastante auge durante sus inicios, pero actualmente ha quedado en desuso.

**Ubuntu Touch** es un sistema operativo móvil desarrollado por Canonical Ltd. basado en Linux.

**Tizen OS** es un Sistema Operativo móvil de código abierto basado en Linux. Tiene mucho en común con Android. Fue desarrollado por Samsung en asociación con Intel. Al ser soportar servicios que no son de Google, es utilizado sobre todo por habitantes chinos que no tienen acceso a servicios de Google.

**Jolla Sailfish OS** está basado en Linux. Fue creado por antiguos empleados de Nokia. Es compatible con Android.

## Estadísticas

El último estudio de Kantar World Panel sobre Sistemas Operativos muestra la siguiente situación en España (primer trimestre de 2015):

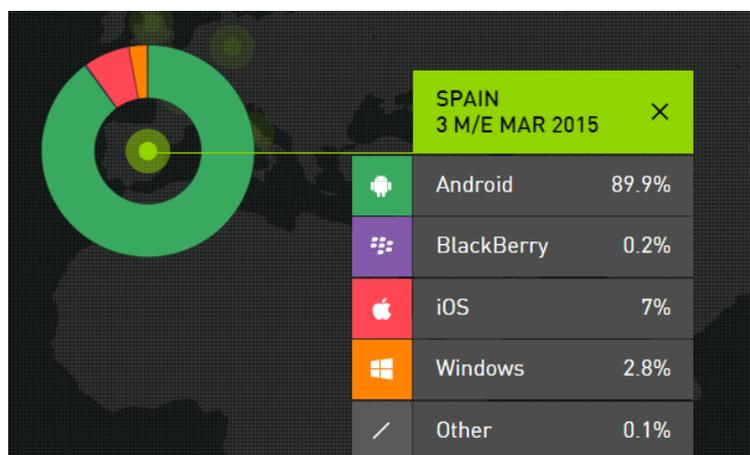


Figura 11: Distribución SO móviles España

En el resto de Europa, la situación con respecto a la segregación de los Sistemas Operativos Móviles difiere ligeramente. A continuación, se muestra la situación de Alemania, cuyos porcentajes de cuota de mercado de Sistemas Operativos Móviles se encuentra alineada con la media europea:

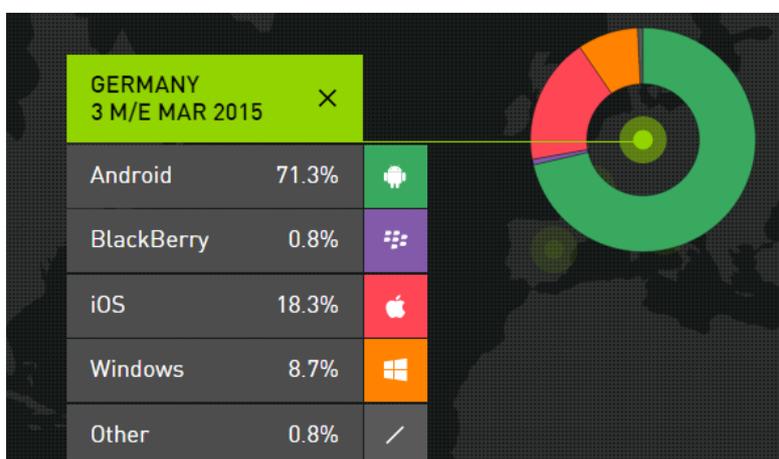


Figura 12: Distribución SO móviles Alemania

Por su parte, en EEUU, Apple se acerca bastante más a Android, mostrando un 36% de cuota de mercado frente al 57% de Android. Japón era el único país donde iOS (57%) superaba a Android (41%) en 2014, habiéndose revertido la situación en favor de Android a principios de 2015.

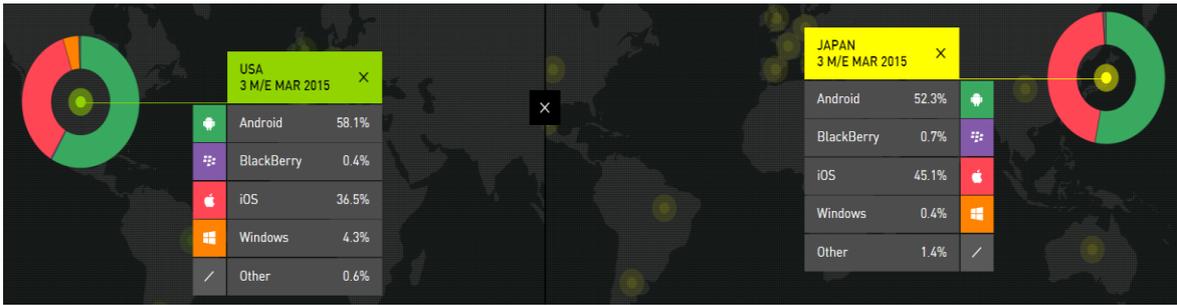


Figura 13: SO móviles: EEUU vs Japón

Finalmente, con el objetivo de ofrecer una panorámica más global, se incluyen gráficas que muestran la cuota de mercado a nivel mundial, tanto a nivel de Sistemas Operativos móviles como de compañías vendedoras de terminales móviles.

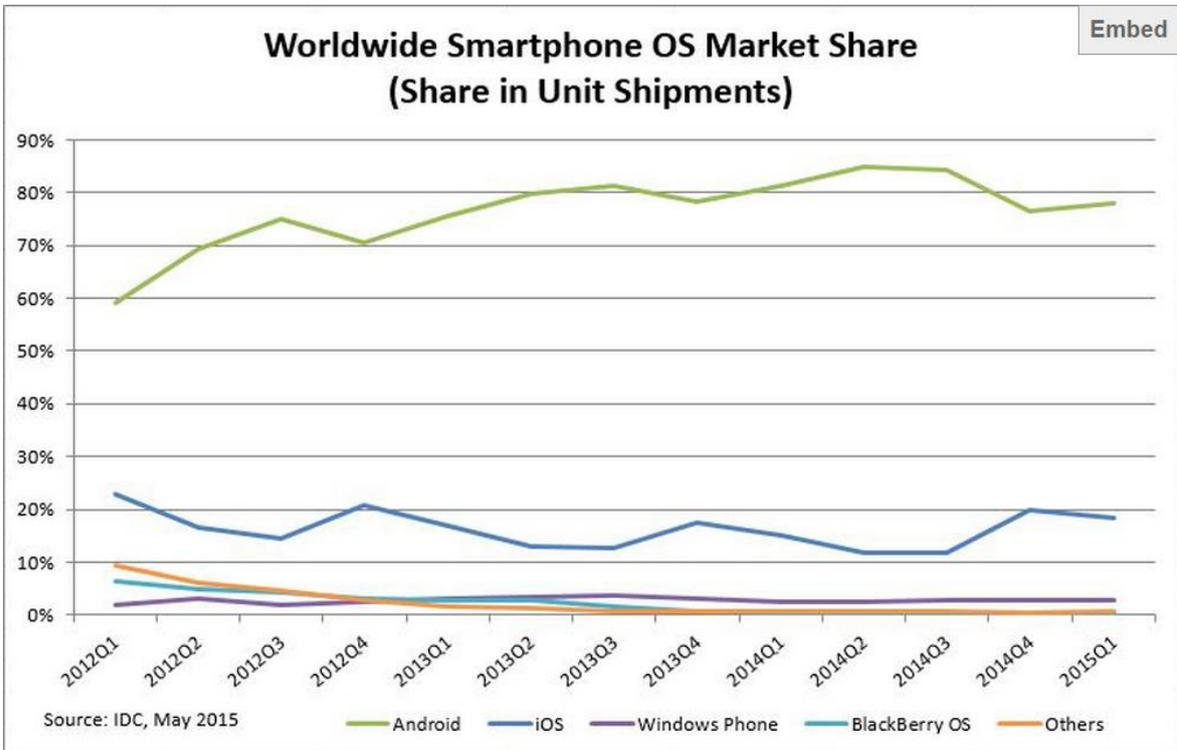


Figura 14: SO móviles en el mundo

Period	Android	iOS	Windows Phone	BlackBerry OS	Others
Q1 2015	78.0%	18.3%	2.7%	0.3%	0.7%
Q1 2014	81.2%	15.2%	2.5%	0.5%	0.7%
Q1 2013	75.5%	16.9%	3.2%	2.9%	1.5%
Q1 2012	59.2%	22.9%	2.0%	6.3%	9.5%

Tabla 3: SO móviles en el mundo

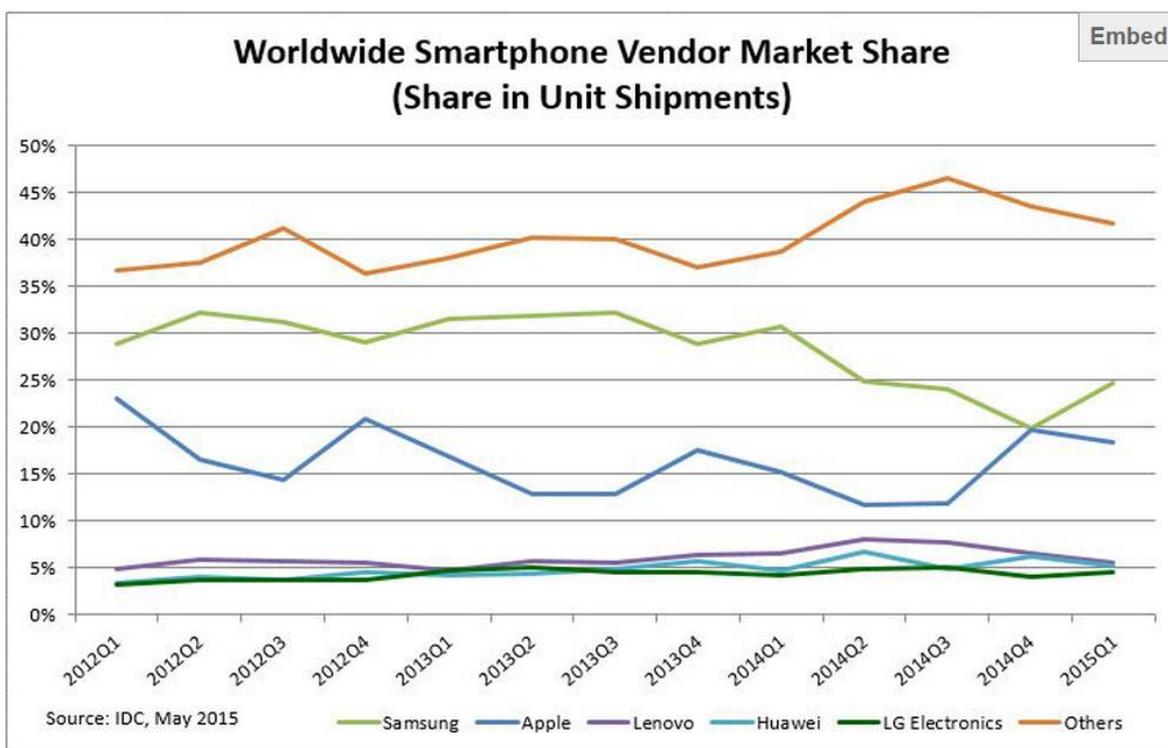


Figura 15: Vendedores de smartphones en el mundo

Period	Samsung	Apple	Lenovo*	Huawei	LG Electronics	Others
Q1 2015	24.6%	18.3%	5.6%	5.2%	4.6%	41.7%
Q1 2014	30.7%	15.2%	6.6%	4.7%	4.3%	38.6%
Q1 2013	31.5%	16.9%	4.7%	4.2%	4.7%	38.1%
Q1 2012	28.9%	22.9%	5.0%	3.4%	3.2%	36.6%

Tabla 4: Vendedores de smartphones en el mundo

No obstante, todos estos datos se encuentran en constante volatilidad. De hecho, el panorama ha cambiado a pasos agigantados desde 2012, donde BlackBerry y otros Sistemas Operativos se mostraban muy superiores a Android e iOS. Por otro lado, es razonable que Android supere a Apple, ya que Android cuenta con una gran variedad de smartphones, mientras que Apple solo dispone de los iPhone para iOS. No obstante, tras el lanzamiento del iPhone 6, los números de 2015 mostrarán mejores resultados que los de 2014 para Apple. Asimismo, tras los esfuerzos de Microsoft, Windows Phone experimentó en 2014 un repunte en detrimento de Android, no corroborando esta tendencia y perdiendo cuota de mercado en lo que llevamos de 2015.

- Los datos anteriores han sido extraídos de las webs “Kantar Media Panel” e “IDC (International Data Corporation)” -

## **2.4 Lenguajes de programación web empleados**

Para poder interactuar con la máquina, es necesario desarrollar la aplicación web en lenguajes entendibles por dicho equipo. Para ello, se empleará texto en lenguaje formal empleando distintos lenguajes mediante un editor de texto plano (Notepad++ en nuestro caso).

En este apartado se llevará a cabo un estudio de los lenguajes de programación web empleados para desarrollar la parte web de la aplicación de este PFC, desde la cual accede Administración y los Profesores para la gestión de exámenes y calificaciones.

### **HTML**

Se trata de un lenguaje de marcado que permite definir el contenido de la página web. En su sintaxis, HTML dispone de elementos o etiquetas (tags) delimitados por los siguientes signos < >. Estas etiquetas son los elementos que los navegadores leen e interpretan para montar y dar forma a las páginas web. Por ello, es posible obtener visualizaciones distintas para cada navegador, ya que son éstos los que interpretan los datos obtenidos.

### **CSS**

Es un lenguaje utilizado para la presentación de documentos HTML. Permite definir el estilo de la página web, es decir, permite indicar el formato, posición, tamaño o color de un elemento de una página. Normalmente, el código CSS se encuentra embebido (mezclado) con el código HTML mediante delimitadores. No obstante, también es posible crear archivos .css e importarlos.

### **JQUERY**

JavaScript es un lenguaje cliente utilizado para dotar de efectos y procesos dinámicos a documentos HTML. Es empleado principalmente por los programadores web para dar respuestas rápidas a las acciones del usuario sin necesidad de enviar la información de lo que ha hecho el usuario al servidor y esperar respuesta de éste. El código JavaScript se carga al mismo tiempo que el código HTML en el navegador y reside en el cliente, por lo que JavaScript continúa funcionando incluso aunque se produzca un corte en la conexión a Internet.

JQUERY es un framework de JavaScript, es decir, se trata de una biblioteca de JavaScript de código abierto. Permite establecer cómo se va a comportar la aplicación antes las interacciones del usuario. Su característica principal es que permite cambiar el contenido de una página web sin necesidad de recargarla. Asimismo, el lenguaje empleado es algo más simple que el utilizado en JavaScript.

Muy útil para gestionar distintos componentes como menús desplegados o calendarios.

## **PHP**

PHP es un potente lenguaje servidor de alto nivel de código abierto que puede ser incrustado en páginas web HTML. Permite conectar una aplicación a un servicio web. PHP se ejecuta en el servidor (no es posible ejecutarlo en nuestro ordenador a no ser que lo hagamos funcionar como servidor). PHP, una vez es interpretado por el servidor, genera una salida HTML que permite visualizar los resultados en los navegadores.

Se emplea en páginas web dinámicas. Dado que en nuestro caso la información de la aplicación se encuentra almacenada en una base de datos, es necesario emplear PHP para acceder a ella. El código PHP está comprendido entre etiquetas de comienzo y final especiales: `<?php` y `?>`. La ventaja principal de PHP es que es extremadamente simple para el principiante, pero a su vez, ofrece muchas características avanzadas para los programadores profesionales y más avanzados.

Lo que distingue a PHP de JavaScript es que el código es ejecutado en el servidor, generando HTML y enviándolo al cliente como si fuera una página web estática. El cliente recibirá los resultados que el servidor devuelve después de interpretar el código PHP, sin ninguna posibilidad de determinar qué código ha producido el resultado recibido. Es decir, a través de nuestro navegador podríamos ver el código HTML, pero nunca el código PHP que dio lugar al resultado HTML.

## **SQL**

Lenguaje estándar de programación para el acceso a bases de datos. Permite trabajar con cualquier tipo de lenguaje (ASP o PHP) en combinación con cualquier tipo de base de datos (MS Access, Oracle, SQL Server, MySQL...).

## 2.5 Aplicaciones similares

La aplicación aquí desarrollada, por definición, guarda el estilo de la serie de aplicaciones anteriores llevadas a cabo en el laboratorio DSLab de la Escuela Politécnica Superior de la UAM. Estas restricciones incluyen aspectos como:

- Tamaño de la letra
- Colores
- Formato del menú inicial
- Icono: Resolución de gráficos
- Relación de aspecto de gráficos
- Formato de botones
- Menú de ayuda
- Página “about”
- Funcionamiento del tutorial

Bajo el mecenazgo del laboratorio DSLab de la UAM, ya se han implementado tres aplicaciones Android gratuitas con el objetivo de hacer más dinámica la realización de las guías de ejercicios así como para hacer más fácil el estudio a los alumnos de las asignaturas desde su dispositivo Android. Dichas aplicaciones se desarrollaron con el fin de constituir un apoyo adicional a los alumnos de las asignaturas Circuitos Electrónicos Digitales y Dispositivos Integrados Especializados del Grado en Ingeniería de Tecnologías y Servicios de Telecomunicación.

Como se ha explicado anteriormente, la aplicación contemplada en este PFC sigue la línea de las realizadas por DSLab y supone un paso más allá del mero aprendizaje de los alumnos, como pudieran ser las anteriores. A continuación, se procede a explicar las diferentes aplicaciones educativas desarrolladas por el laboratorio:

### Sequential Circuits



*Figura 16: Icono Sequential Circuits*

Esta aplicación fue publicada en diciembre de 2013. Se caracteriza por ofrecer una serie de ejercicios de electrónica secuencial (máquinas de estados), incluye consejos para ayudar a resolver estos problemas, un tutorial centrado en máquinas de estados tipo Moore y un manual con el manejo básico de la aplicación. Los contenidos de esta aplicación abarcan temas como: Mealy, Moore, Flip Flops, memoria ROM, puertas lógicas, detector de secuencias, aritmética serie, solapamiento y control de

motores. Un año después de la publicación de la primera versión, la aplicación cuenta con 1.800 descargas y una nota media de 4,14 sobre 5 en las valoraciones.

## Combinational Circuits



*Figura 17: Icono Combinational Circuits*

Centrada en ejercicios de electrónica combinacional. Al igual que la anterior incluye consejos para ayudar a resolver los problemas. También dispone de un manual que explica cómo utilizar la aplicación y un tutorial sobre circuitos combinacionales. Esta aplicación se centra en: memoria ROM, multiplexores, puertas lógicas, PALs, BCD 7-segmentos, Codificador y decodificador, código Gray y Karnaugh. Su primera versión se publicó el 16 de diciembre de 2013. En la actualidad se han publicado un total de 7 versiones. La aplicación se ha descargado aproximadamente 3.200 veces y la nota media de sus valoraciones es un 4 sobre 5.

## Karn Map



*Figura 18: Icono Karn Map*

Ofrece una serie de herramientas para la simplificación de funciones lógicas. Si la función tiene de 3 a 5 variables tiene la opción de la simplificación automática y gráfica con mapas de Veitch-Karnaugh. De 3 a 7 variables simplifica funciones introducidas por expresión lógica, marcando una tabla de verdad, eligiendo minitérminos o maxitérminos. Además dibuja el esquemático de dicha función. Los contenidos tratados son: simplificación por mapas Veitch-Karnaugh, simplificación de hasta 7 variables, tabla de verdad, sumatorio de minitérminos, producto de maxitérminos, expresiones lógicas, representación esquemática, conversor, binario, octal y hexadecimal. Esta aplicación es la más reciente, no obstante en los dos primeros meses tras su publicación, consiguió instalarse en más de 1.000 dispositivos. Y al igual que las anteriores, la media obtenida de las valoraciones realizadas por los usuarios es muy alta, en este caso 4,78.

Previo a la concepción de la aplicación, es importante estudiar si el mercado ofrece algo similar, puesto que si lo que se va a desarrollar ya existe, no tiene mucho sentido llevarlo a cabo. Para ello, el análisis se llevará a cabo en los principales sistemas operativos, que, como se ha visto anteriormente, son Android, iOS y Windows Phone. Cada uno de ellos dispone de su propia tienda de aplicaciones: Google Play Store en Android, App Store de iOS y Windows Phone Store la de Windows Phone.

Por otro lado, a continuación se va a llevar a cabo un análisis de aplicaciones similares existentes en el mercado. Para ello, se han buscado aplicaciones cuya funcionalidad es parecida a la de la aplicación que se desarrolla en este PFC. En nuestro caso, la principal función de la aplicación es la creación y gestión de exámenes de respuesta múltiple.

Tras el estudio de aplicaciones similares en los diferentes mercados, se ha encontrado que no existe ninguna aplicación parecida disponible en el mercado. De hecho, como se explica en el apartado de Motivación, uno de los motivos que condujeron a la elección de este PFC fue la ausencia de una aplicación que permitiera la realización de exámenes de respuesta múltiple gestionada por personal universitario (Profesorado y Administración).

No obstante, se va a llevar a cabo un estudio diferenciado, analizando tanto algunas aplicaciones de realización de exámenes de tipo test como aplicaciones de gestión de calificaciones.

## SIGMA

En lo que respecta a aplicaciones web de gestión de calificaciones, cabe señalar que existe múltiples en Internet. Sin embargo, teniendo en cuenta el propósito de este PFC, la aplicación que mejor encaja con la finalidad de lo que aquí se desarrolla es la aplicación web para la gestión de calificaciones que dispone la UAM. Por tanto, se va a evaluar la aplicación web SIGMA, la cual se emplea en la Universidad Autónoma de Madrid así como en múltiples universidades españolas para la gestión de expedientes. Como vemos, en este caso el diseño está adaptado a los colores corporativos de la Universidad.



Figura 19: Página Inicio Sigma UAM

**Accesos:** este menú permite la inscripción en Másteres, Doctorados, Dobles Titulaciones y otros cursos especiales.

**Encuestas Web:** al acceder a este menú, se permite la posibilidad de rellenar encuestas.

**Expedientes:** menú que posee toda la información asociada al expediente concreto del alumno con la sesión abierta.

**Gestión de Becas:** menú para la gestión de becas relacionadas con la Universidad.

**Matrículas:** menú que permite realizar la matrícula respectiva a cada curso lectivo así como consultar la información relativa a matrículas de cursos pasados.

**Tasas:** menú para la gestión de los pagos concernientes a las tasas universitarias.

**Tercer Ciclo:** menú para la gestión y consulta de información asociada a terceros ciclos como puedan ser másters o doctorados.

**Trabajos Fin de Estudios y Prácticas en Empresa:** menú que permite solicitar plazas para optar a prácticas en empresas externas a la Universidad.

Como vemos, aunque dicha aplicación web contenga más funcionalidades de las que abarca este proyecto, la idea para la que fue concebida es la misma. Como se observa, aunque no tengamos acceso, los administradores cargan la información necesaria para los distintos profesores y alumnos, haciendo posible que éstos solo accedan a la información que les concierne (p.ej.: solo pueden rellenar encuestas de las asignaturas en las que están matriculados o solo pueden acceder a su propio expediente). Asimismo, los profesores también cargan información, pues son los encargados de subir las notas de los alumnos antes del cierre de actas. Por tanto, al igual que la aplicación que se desarrolla en este proyecto, el acceso es restringido a los usuarios que se han habilitado desde el módulo de Administración, gestor de la aplicación.

	Año académ. ↓	Código ↓	Descripción ↓	Cr. ↓	Gr. ↓	Convocatoria ↓	Calificación ↓	C.N. ↓	Cvl. ↓	Tipo Asig. ↓	Cic. ↓	Cur. ↓	Observaciones ↓
	2009/10-0	15296	FÍSICA I	7.5	11	Febrero	Notable	7.5	N	Troncal	1	1	
	2009/10-0	15297	ANÁLISIS MATEMÁTICO I	6.0	11	Febrero	Sobresaliente	9.0	N	Troncal	1	1	
	2009/10-0	15298	ÁLGEBRA Y ECUACIONES DIFERENCIALES	6.0	11	Febrero	Aprobado	5.0	N	Troncal	1	1	
	2009/10-0	15299	FUNDAMENTOS DE PROGRAMACIÓN	7.5	11	Febrero	Notable	8.4	N	Troncal	1	1	
	2009/10-0	15300	INTRODUCCIÓN A LA INGENIERÍA	3.0	11	Febrero	Sobresaliente	9.6	N	Obligatoria	1	1	
	2009/10-0	15301	FÍSICA II	6.0	11	Junio	Notable	7.1	N	Troncal	1	1	

Figura 20: Expediente Sigma UAM

Por otro lado, SIGMA permite a los usuarios cambiar la contraseña. Asimismo, tiene implementadas políticas de buenas prácticas en lo que respecta a restricciones de contraseña.



Figura 21: Cambio Contraseña Sigma UAM

SIGMA no dispone de un módulo de realización de exámenes. Sin embargo, como ya se ha comentado previamente, sí posee de un menú que permite rellenar encuestas. Por ello, no debería suponer mucha dificultad adaptar SIGMA para la realización de exámenes de respuesta múltiple.

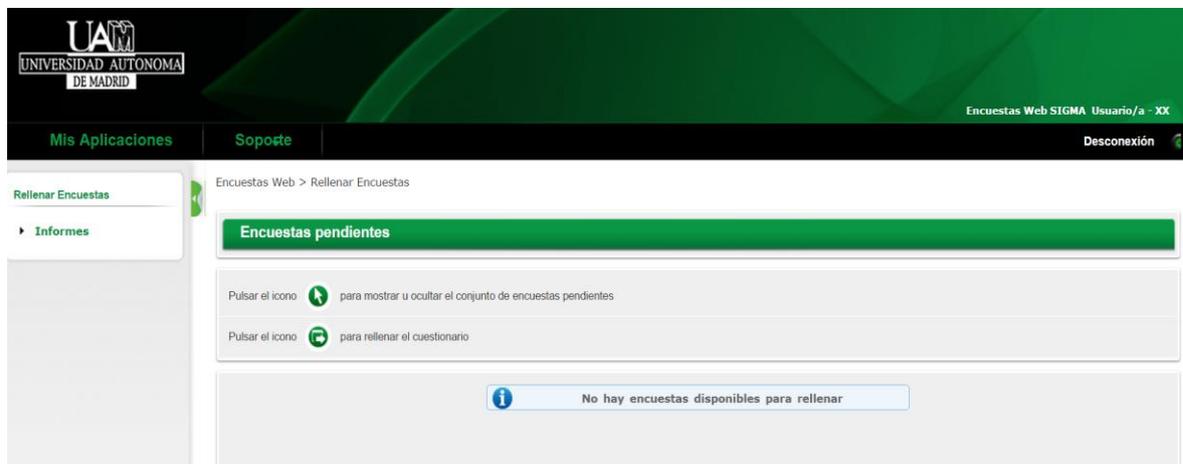


Figura 22: Encuestas Sigma UAM

Por otro lado, en el módulo de Soporte, dispone de opciones de Ayuda (no operativa en el momento del análisis), posibilidad de cambio de contraseña, contacto vía e-mail o información de la aplicación. Además, cuenta con posibilidad de cerrar la sesión iniciada haciendo click en el botón de Desconexión.



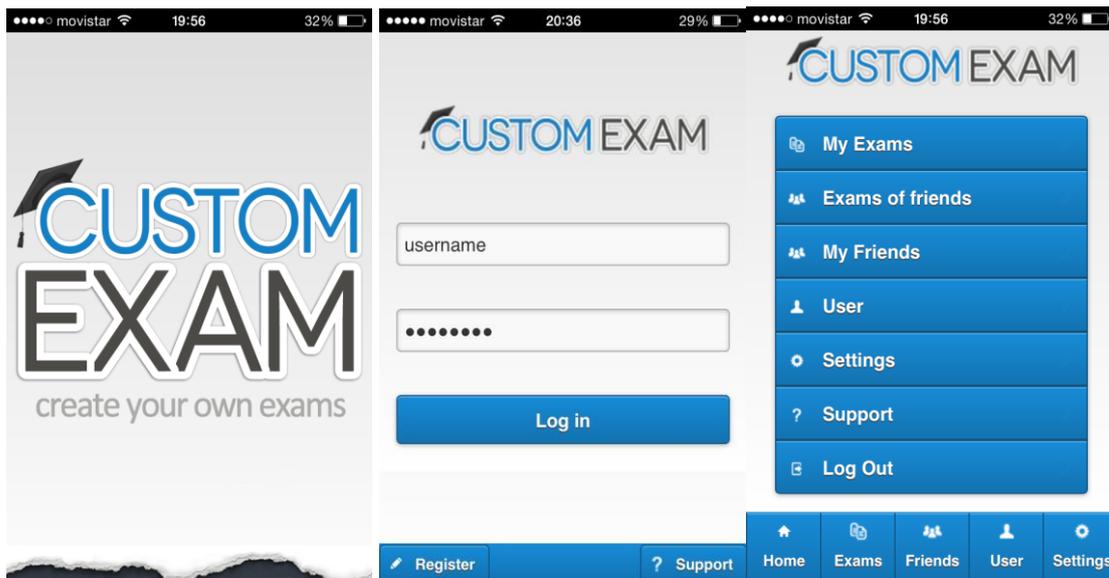
*Figura 23: Soporte Sigma UAM*

Además, SIGMA cuenta con acceso a dicha aplicación web desde Smartphones y Tablets tanto Android como Apple, siendo la funcionalidad prácticamente la misma que en la web.

## CUSTOM EXAM

La única aplicación que se ha encontrado con la funcionalidad de crear y gestionar exámenes de respuesta múltiple a través tanto de web como de dispositivos móviles es la aplicación Custom Exam.

La aplicación Android (también disponible en iOS) controla el acceso mediante nombre de usuario y contraseña. Al contrario que la aplicación que desarrolla este PFC, el acceso está abierto a cualquier usuario que desee registrarse.



Figuras 24, 25 y 26: Aplicación Móvil Custom Exam

Como se observa, el menú principal permite acceder a los exámenes creados por el usuario, exámenes creados por amigos, ver el listado de amigos, ver la información del perfil de usuario, menú de ajustes, ayuda y cierre de sesión. En el menú de ajustes, se permite configurar el idioma, contactar por email con los administradores de la aplicación y visitar la web.



Figura 27: Ajustes aplicación Custom Exam

A continuación, se analizará el acceso web, el cual permite la opción de recuperar contraseña.

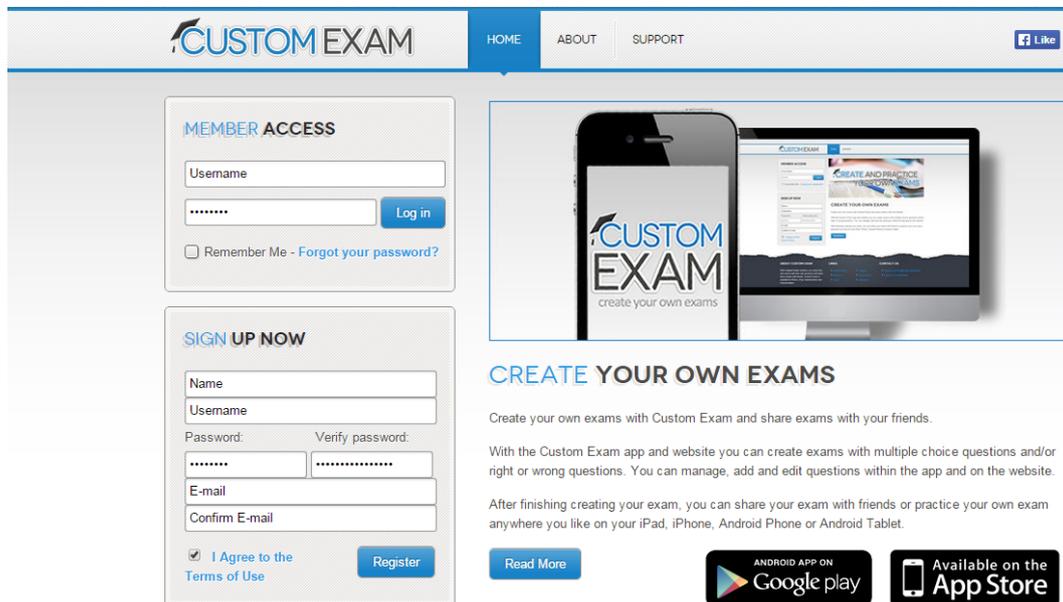
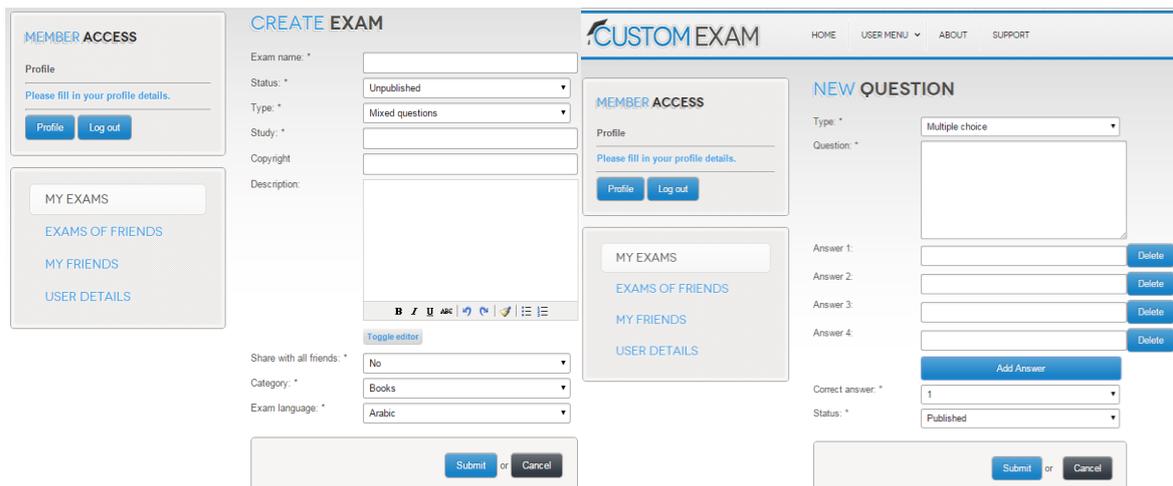


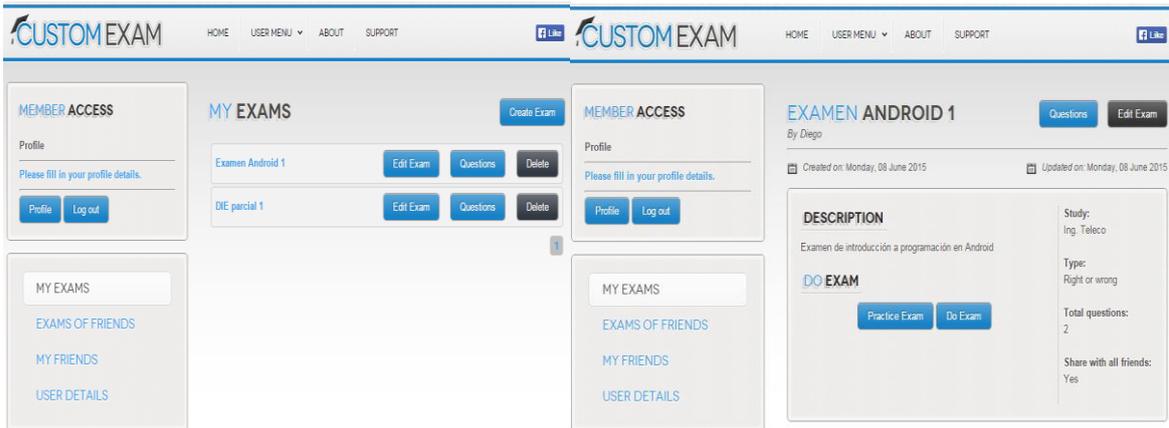
Figura 28: Inicio web Custom Exam

Una vez se ha accedido con un usuario creado previamente desde la web o el dispositivo móvil, pasaremos a analizar la creación de exámenes. Vemos que existen campos obligatorios como el nombre del examen, status, tipo, asignatura, compartición con amigos, categoría e idioma y campos opcionales como copyright o descripción.



Figuras 29 y 30: Crear examen web Custom Exam

Existe opción de añadir el número de respuestas que uno desee. Al igual que en la aplicación que desarrolla este proyecto, solo puede existir una solución correcta. Sin embargo, para añadir preguntas, es preciso hacerlo de una en una, lo cual resulta tedioso. Una vez se ha creado el examen, aparecerá en el menú “My Exams”, existiendo la posibilidad de realizar dicho examen.



Figuras 31 y 32: Exámenes Web Custom Exam

Se ha llevado a cabo el examen, observando que una vez realizado, se generan estadísticas del tiempo empleado, total de preguntas, total de respuestas correctas e incorrectas y calificación en %.

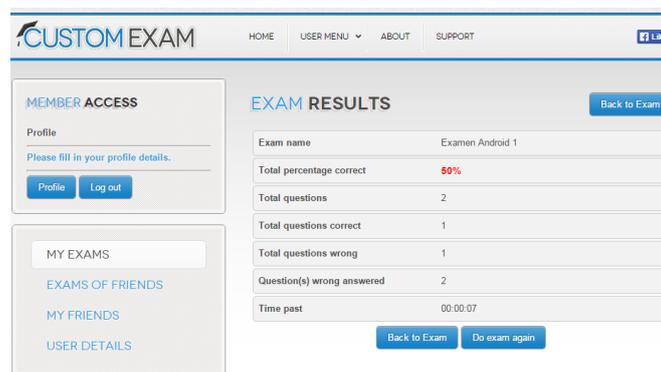


Figura 33: Resultado Examen Custom Exam

Se comprueba por tanto que no hay opción de fijar un tiempo máximo para la realización del examen.

Asimismo, se ha observado que no existe una correcta sincronización entre los exámenes creados desde la web y los de la aplicación móvil. Asimismo, el acceso no está restringido, por lo que cualquiera puede hacer los exámenes creados por un usuario. Además, la funcionalidad de la web y de la aplicación es la misma.

No obstante, como se ha indicado anteriormente, no se han encontrado aplicaciones en el mercado con posibilidad de gestionar vía web el manejo y creación de los exámenes de tipo test que se realizan desde el Smartphone o Tablet de la manera que propone este PFC.

## 3 Diseño

---

### 3.1 Aplicación Web

#### 3.1.1 Finalidad de la aplicación web

Como se ha comentado previamente, el objetivo principal de este PFC es permitir que los alumnos lleven a cabo exámenes de respuesta múltiple mediante sus dispositivos móviles (smartphones y tabletas) Android. Asimismo, la finalidad fundamental de este proyecto, la cual subyace en todo momento sobre los objetivos descritos, es la de adquirir competencias sobre programación tanto web como Android, que permitan desarrollar e incrementar mis aptitudes con respecto al conocimiento logrado durante la carrera.

En concreto, para llevar a cabo el objetivo principal de realizar exámenes de tipo test en dispositivos móviles, se contempló dotar a la aplicación de una parte web para facilitar la creación y gestión de exámenes por parte de los profesores. Sin duda, crear exámenes incluyendo preguntas y respuestas a través de un dispositivo móvil no es tan accesible como desde un equipo portátil o de sobremesa. No obstante, se ha permitido en todo momento que los exámenes se puedan crear desde smartphones o tablets de cualquier modelo y sistema operativo, existiendo la posibilidad de acceder a la parte web a través del navegador del dispositivo móvil.

Por tanto, la finalidad de dotar a la aplicación de acceso web ha sido en todo momento para facilitar la accesibilidad y propiciar un mejor manejo de la aplicación, siendo más intuitivo crear exámenes mediante un teclado estándar que a través del de un dispositivo móvil.

Una vez quedó clara la necesidad de incluir esta parte web, se comenzó a delinear su alcance. Rápidamente y en colaboración con el ponente de este proyecto se convino lo siguiente: esta parte permitirá el acceso de profesores para crear y gestionar exámenes así como de Administración para la creación de asignaturas y asignación de profesores y alumnos a ellos.

### **3.1.2 Elección de requisitos: parte web**

Tras analizar el conjunto de aplicaciones existentes en el mercado, resumido de forma breve en el epígrafe de estado del arte de esta memoria, se llegó a la conclusión de que actualmente no existe ninguna aplicación con la misma funcionalidad que la que aquí se desarrolla. Como se ha explicado, existen aplicaciones para llevar a cabo exámenes de tipo test en el móvil, así como aplicaciones web de gestión de calificaciones. Sin embargo, no se ha encontrado ninguna aplicación tan enfocada al ámbito académico y que segregue el acceso de tal forma que Administración controle qué asignaturas hay, quién las imparte, qué alumnos hay en cada asignatura y que solo los profesores puedan crear exámenes.

Una vez puestos en contexto, a continuación se argumentarán las decisiones tomadas para la definición de requisitos de la parte web del proyecto.

En primer lugar, el módulo de Administración debe ser el gestor principal, es decir, el único usuario autorizado para llevar a cabo cambios y para introducir las asignaturas vigentes así como asignar los profesores y alumnos a dichas asignaturas una vez han sido creadas.

Antes de crear una asignatura, es necesario insertar profesores en el sistema. Para ello, se solicita introducir nombre, apellidos email y una password para que accedan como usuarios. Asimismo, es posible borrar profesores. La web muestra un menú con el listado de profesores en la aplicación y la información de sus campos.

Para crear una asignatura, es preciso introducir el nombre, una breve descripción y asociar un profesor (de los ya creados en el sistema). Asimismo, es posible borrar asignaturas. La web muestra un menú con el listado de asignaturas en la aplicación y la información de sus campos.

Una vez creada la asignatura, asociada en su creación a un profesor en concreto, otra de las funcionalidades del módulo de Administración es la posibilidad de generar listados de alumnos para informar de la relación de estudiantes que cursarán cada una de las asignaturas. En este caso, dado que el número de estudiantes que cursan una asignatura es elevado y sería costoso añadirlos manualmente, se ofrece la posibilidad de incluirlos en masa a través de un fichero de texto plano. Asimismo, es posible borrar alumnos. Al igual que en los casos anteriores, la web muestra un menú con el listado de alumnos dados de alta en la aplicación y la información de sus campos.

Como se ha explicado anteriormente, dada la mayor accesibilidad desde el ordenador para gestionar exámenes que desde un dispositivo móvil, los profesores también accederán desde la web a la aplicación. Allí, tendrán únicamente permisos sobre las asignaturas que previamente se le asignaron por parte de Administración.

Por un lado, los profesores podrán consultar las calificaciones de los alumnos en los exámenes. Asimismo, existirá la posibilidad de filtrar por email del alumno, de manera que se facilite la búsqueda en el caso de existir muchas entradas en la tabla de notas de alumnos.

Por supuesto, la aplicación dispondrá de una opción para crear un examen. Tras seleccionar una asignatura, introduciendo la información pertinente en un formulario (password, duración, hora de comienzo, descripción, etc.) se podrá crear un examen. Una vez creado el examen, los alumnos que dispongan de la aplicación Android en sus dispositivos móviles, podrán visualizar el examen.

Además, tras seleccionar una asignatura, existirá otra opción que permitirá al profesor activar o desactivar los exámenes ya creados. De esta forma, tras activar el profesor un examen desde la parte web, si se les proporciona a los alumnos la contraseña y siempre dentro del horario prefijado al crear el examen, será posible llevar a cabo el examen en los dispositivos móviles de los alumnos que cuenten con la aplicación.

En cuanto al equipo en el que se desarrolla esta parte del proyecto, el único requisito es disponer de Windows 7, ya que la última versión de WampServer no funciona en versiones anteriores de Windows. Ya que el ordenador donde se ha realizado el PFC posee SO Windows, no se contemplan requisitos mínimos para otros SO como Mac o Linux.

### 3.1.3 Módulos: parte web

Como se ha explicado en el apartado anterior, la parte web dispondrá de dos módulos, cuyos requisitos se han explicado previamente:

- Administración
- Profesores

A continuación, se incluye un diagrama de flujo que resume la funcionalidad definida e implementada para la parte web. No obstante, en primer lugar es preciso explicar qué información contiene y cómo la expresa este tipo de representación.

Un diagrama de flujo refleja la representación gráfica de un proceso. Generalmente, presenta un único punto de inicio y un único punto de cierre, aunque puede tener más, siempre que cumpla con la lógica requerida.

Las siguientes, son acciones previas a la realización del diagrama de flujo:

- Identificar las ideas principales al ser incluidas en el diagrama de flujo.
- Definir qué se espera obtener del diagrama de flujo.
- Identificar quién lo empleará y cómo.
- Establecer el nivel de detalle requerido.
- Determinar los límites del proceso a describir.

Los pasos a seguir para construir el diagrama de flujo son:

- Establecer el alcance del proceso a describir. De esta manera quedará fijado el comienzo y el final del diagrama. Frecuentemente el comienzo es la salida del proceso previo y el final la entrada al proceso siguiente.
- Identificar y listar las principales actividades/subprocesos que están incluidos en el proceso a describir y su orden cronológico.
- Si el nivel de detalle definido incluye actividades menores, listarlas también.
- Identificar y listar los puntos de decisión.
- Construir el diagrama respetando la secuencia cronológica y asignando los correspondientes símbolos.

- Asignar un título al diagrama y verificar que esté completo y describa con exactitud el proceso elegido.

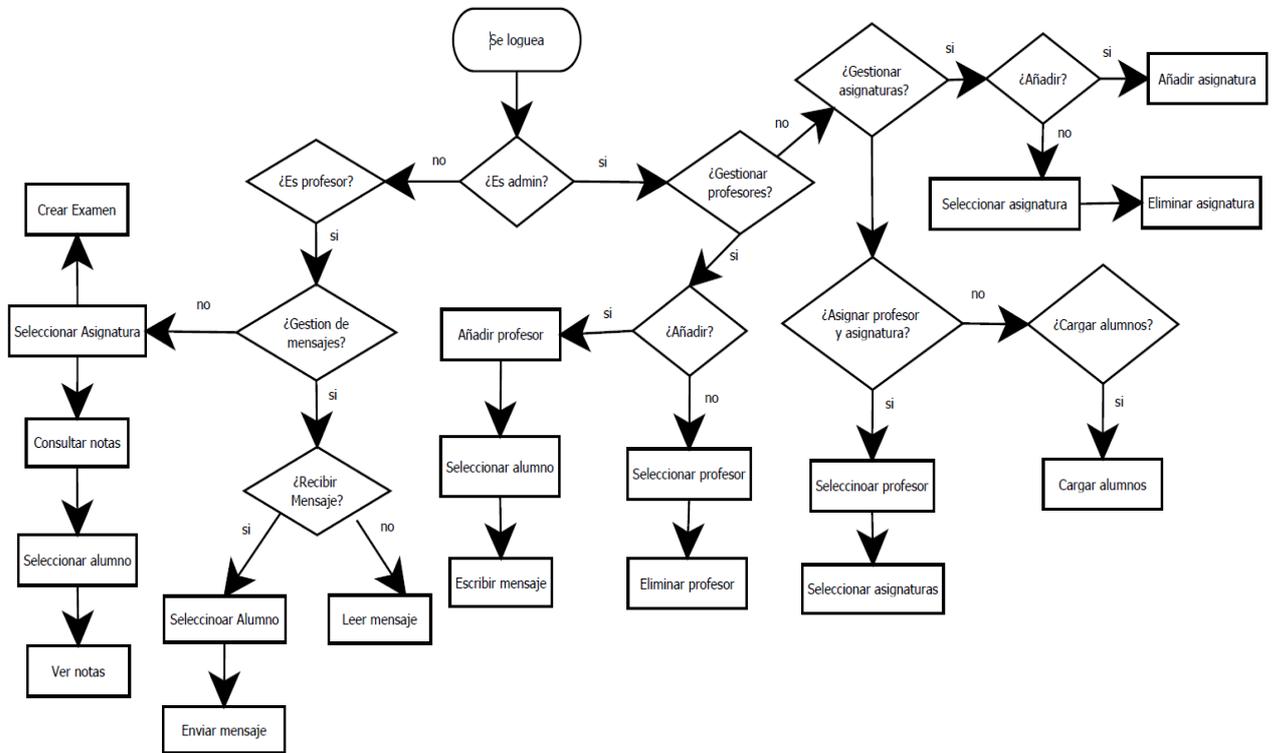


Figura 34: Diagrama Flujo Aplicación Web

Como se puede observar en el diagrama, el primer paso consiste en iniciar sesión. Para ello, se discrimina según el nivel de acceso: Administrador o Profesor.

Tras acceder como Profesor, es posible entrar en la opción de mensajería para comunicarse con los alumnos y leer los mensajes recibidos o enviar nuevos. Por otro lado, podría acceder a la parte de gestión de asignaturas, pudiendo crear exámenes o consultar las notas de los alumnos.

Si se accede como Administrador, es posible gestionar profesor, asignaturas o alumnos. Entrando a la parte de gestión de profesores, es posible introducir nuevos profesores o eliminar alguno de los ya existentes. Por otro lado, en gestión de asignaturas es posible crear una asignatura (asociándole un profesor que ya esté en la aplicación) o eliminar alguna de las ya existentes. Si por el contrario se accede a gestionar alumnos, se permite añadir alumnos a las asignaturas ya creadas (a través de un fichero de texto plano).

Por tanto, como se puede ver, la definición del diagrama concuerda con lo definido en el establecimiento de requisitos.

### 3.1.4 Limitaciones: parte web

Debido a que el PFC tiene un tamaño delimitado desde el principio por los requisitos definidos, existen limitaciones en cuanto al diseño, ya que dado que el tiempo es limitado, no es posible satisfacer en esta primera versión todas las funcionalidades que uno desearía. No obstante, en este apartado se enumeran una serie de limitaciones de la aplicación web con el objetivo de contemplarlas en futuras versiones de la aplicación. Asimismo, en el apartado “6.2: Trabajo Futuro” también se valoran posibles modificaciones y mejoras para otorgar mayor funcionalidad y calidad a la aplicación. Abajo se enumeran algunas limitaciones de la aplicación web:

- Las passwords pueden ser tan simples como el usuario decida. No existe complejidad de contraseñas.
- No existe opción de modificar la contraseña.
- Las preguntas y las respuestas de los exámenes solo admiten texto. No es posible añadir imágenes a las preguntas y/o a las respuestas.
- El idioma por defecto es el castellano. No hay opción de acceder a la aplicación web en otro idioma.
- No se ha habilitado opción de incluir los profesores en masa a través de un fichero de texto plano.
- Por limitación de los requisitos acordados, solo existen 4 respuestas posibles y una correcta. Por ello, no se ha creado opción para personalizar el número de respuestas posibles y el de soluciones correctas.
- No se ha implementado posibilidad de consultar calificaciones referentes a cursos pasados.

## **3.2 Aplicación Android**

### **3.2.1 Finalidad de la aplicación Android**

Como se ha comentado previamente, el objetivo principal de este PFC es permitir que los alumnos lleven a cabo exámenes de respuesta múltiple mediante sus dispositivos móviles (smartphones y tabletas) Android. Asimismo, la finalidad fundamental de este proyecto, la cual subyace en todo momento sobre los objetivos descritos, es la de adquirir competencias sobre programación tanto web como Android, que permitan desarrollar e incrementar mis aptitudes con respecto al conocimiento logrado durante la carrera.

### **3.2.2 Elección de requisitos: parte Android**

Tras analizar el conjunto de aplicaciones existentes en el mercado, resumido de forma breve en el epígrafe de estado del arte de esta memoria, se llegó a la conclusión de que actualmente no existe ninguna aplicación con la misma funcionalidad que la que aquí se desarrolla. Como se ha explicado, existen aplicaciones para llevar a cabo exámenes de tipo test en el móvil, así como aplicaciones web de gestión de calificaciones. Sin embargo, no se ha encontrado ninguna aplicación tan enfocada al ámbito académico y que segregue el acceso de tal forma que Administración controle qué asignaturas hay, quién las imparte, qué alumnos hay en cada asignatura y que solo los profesores puedan crear exámenes.

Por lo explicado anteriormente, ya sabemos que la parte Android contiene el módulo de Alumnos de la aplicación. Según lo acordado con el ponente del proyecto, los alumnos podrán instalar su aplicación en sus dispositivos móviles. Para entrar en la aplicación, solo podrán hacerlo con usuarios creados previamente por Administración desde la app web. Una vez dentro de la aplicación, los estudiantes podrán hacer uso de 3 opciones:

- Consultar notas de exámenes ya realizados.
- Enviar mensajes a profesores o leer los ya recibidos.
- Realizar exámenes. Para poder completarlos, los exámenes deben haber sido activados, encontrarse dentro de su franja horaria establecida y el estudiante debe haber introducido la contraseña proporcionada por el profesor de manera correcta.

Los requisitos que se han acordado para llevar a cabo en el desarrollo mediante el programa Android Studio son los siguientes:

- **Sistema Operativo:**

- Android

- **Versiones del Sistema Operativo:**

- Desde la versión 4 Ice Cream Sandwich hasta la 5.1 Lollipop

- **Tipos de dispositivos:**

- Móviles
- Tabletas

- **Tipos de pantalla:** Los diferenciaremos según:

- Tamaño: “Small”, “Normal”, “Large” y “Xlarge”
- Densidad: “ldpi”, “mdpi”, “hdpi”, “xhdpi” y “xxhdpi”

- **Idioma:**

- Castellano

Asimismo, el ordenador donde se llevará a cabo el desarrollo del PFC deberá cumplir unos mínimos. Para poder ejecutar Android Studio correctamente, es preciso que el equipo cumpla los siguientes requisitos:

- Microsoft® Windows® 8/7/Vista/2003 (32 o 64-bits)
- 2 GB RAM mínimo, 4 GB RAM recomendados
- 400 MB de espacio en el disco duro
- Como mínimo, 1 GB para Android SDK, el emulador y cachés
- 1280 x 800 de resolución mínima de pantalla
- Java Development Kit (JDK) 7

### 3.2.3 Módulos: parte Android

En este apartado se incluye un diagrama de bloques que muestra la disposición en cuanto a menús y opciones que tendrá el módulo de la parte Android, el de Alumnos:

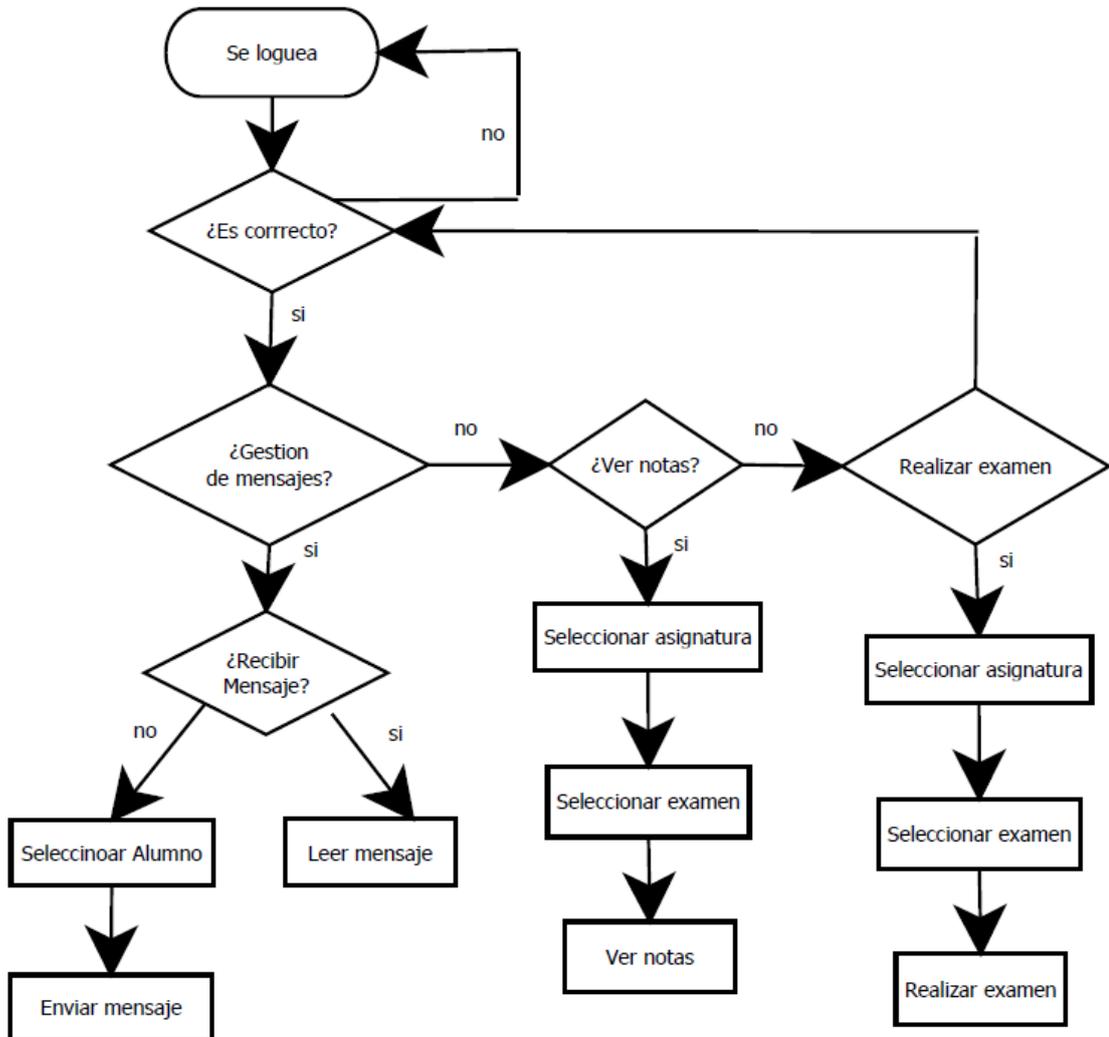


Figura 35: Diagrama Flujo Aplicación Android

### 3.2.4 Limitaciones: parte Android

Debido a que el PFC tiene un tamaño delimitado desde el principio por los requisitos definidos, existen limitaciones en cuanto al diseño, ya que dado que el tiempo es limitado, no es posible satisfacer en esta primera versión todas las funcionalidades que uno desearía. No obstante, en este apartado se enumeran una serie de limitaciones de la aplicación Android con el objetivo de contemplarlas en futuras versiones de la aplicación. Asimismo, en el apartado “6.2: Trabajo Futuro” también se valoran posibles modificaciones y mejoras para otorgar mayor funcionalidad y calidad a la aplicación.

- No existe complejidad de contraseñas. Las passwords pueden ser tan simples como el usuario decida.
- No existe opción de modificar la contraseña.
- El idioma por defecto es el castellano. No hay opción de acceder a la aplicación web en otro idioma.
- La aplicación Android no se encuentra disponible en otros sistemas operativos móviles como iOS o Windows Phone.
- No se ha implementado posibilidad de consultar calificaciones referentes a cursos pasados.
- Los alumnos visualizan desde su dispositivo móvil todos los exámenes creados, a pesar de no estar matriculados en las asignaturas. No obstante, si no disponen de la contraseña, no pueden acceder.

# 4 Desarrollo

---

## 4.1 Aplicación web

Como se ha comentado anteriormente, el alumno no cuenta con conocimiento previo a la realización de este PFC sobre programación web, ya que no se ha cursado en la carrera ninguna asignatura que trate estos aspectos. Por ello, es necesario valorar el esfuerzo que se ha llevado a cabo aprendiendo de forma autónoma conceptos y lenguajes nunca vistos.

### 4.1.1 Primeros pasos

Las aplicaciones web funcionan en cualquier navegador moderno (Mozilla Firefox, Google Chrome, Safari, Internet Explorer, Opera, etc.) y son independientes del Sistema Operativo instalado en el cliente. Para desarrollarlas, es posible instalar un entorno de desarrollo (IDE) cualquiera o simplemente un editor de texto (caso de este proyecto), puesto que se puede probar la funcionalidad del código escrito ejecutándolo en un navegador sin necesidad de conexión a Internet.

Por tanto, empleando los lenguajes descritos en el apartado 2.4 de esta memoria u otros similares, es posible desarrollar una aplicación web. No obstante, cabe señalar la dura tarea que ha sido el aprendizaje de todos los lenguajes de programación web empleados durante el desarrollo de la aplicación.

Para ser accesible desde la web, la aplicación web debe estar alojada en un servidor.

Durante el desarrollo en local de la aplicación, simplemente basta con generar código en un editor de texto y ejecutarlo mediante un navegador. Para facilitar esta tarea y gestionar las BBDD, se ha instalado el programa WampServer. Este programa supone un espacio de desarrollo web en el que se van a poder crear aplicaciones con Apache, bases de datos MySQL y que estén en lenguaje PHP. También incluye, y esta es la parte que más nos interesa, PHPMyAdmin para manejar las bases de datos de forma muy sencilla. Dentro de la carpeta www/, es donde se guardarán los ficheros, scripts y la base de datos de la aplicación web.

Para transferir todos los ficheros empleados desde nuestro ordenador al servidor, se ha empleado el cliente FTP gratuito Filezilla. Para ello, se deben indicar la IP de servidor, usuario FTP y contraseña de usuario y puerto a emplear (21). En la ventana principal de Filezilla, se observa nuestro disco duro con sus carpetas correspondientes en una parte y el sistema de archivos del servidor en otra; para mover los archivos de un lugar a otro simplemente basta con arrastrarlos desde el disco duro al servidor.

El servidor web elegido ha sido Hostinger. Para utilizarlo y subir a la web el trabajo realizado, es preciso crear una cuenta, escogiendo el nombre de dominio y la contraseña para acceder. Existen muchos menús que permiten gestionar la web de muchas maneras:

- Detalles de Cuenta
- Email
- Dominios
- Sitio Web
- Archivos
- Bases de Datos
- Avanzado
- Otros

En Dominios se informa de la url que se deberá emplear para acceder a la web. Por otro lado, en Archivos se encontrarán todos los scripts que se hayan transferido desde local. Por su parte, el menú Bases de Datos permite acceder a phpMyAdmin para la gestión de las BBDD e informa de las BBDD asociadas al sitio web.

#### 4.1.2 Tipos de archivos

En este apartado se van a explicar brevemente el objetivo de lo codificado en cada uno de los ficheros que se han creado para la aplicación web.

**Style.css:** fija el estilo de todas las páginas, menús y botones de la aplicación web.

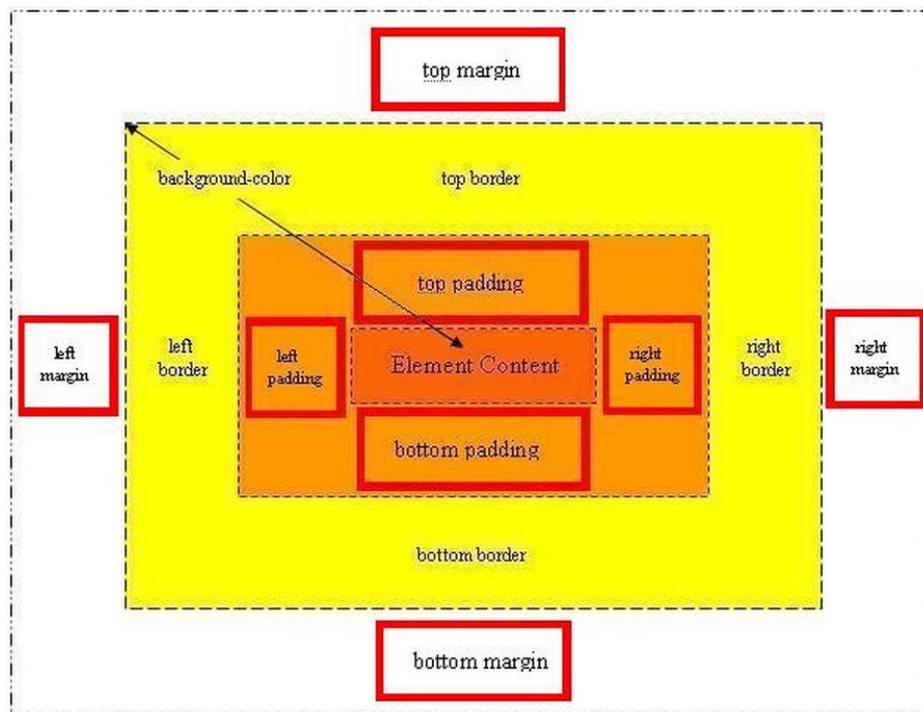


Figura 36: Estilo CSS

**Alumnos.txt:** fichero de texto plano que se emplea para carga alumnos de forma masiva a la asignatura deseada. El formato es el siguiente: Nombre Apellido Email Teléfono Contraseña.

**Examen número.txt:** una vez creados, los exámenes generan un fichero .txt cuyo formato es el siguiente:

----- Numero: 1 -----

Pregunta: Que es Android

Respuesta 1: Un SO | Respuesta 2: Un Lenguaje | Respuesta 3: Un Movil |

Respuesta 4: Una variable

Respuesta Correcta:Un SO

----- Numero: 2 -----

Pregunta: Que es un int

Respuesta 1: Una variable | Respuesta 2: otras | Respuesta 3: otras | Respuesta 4: otras

Respuesta Correcta: Una variable

**Imgs:** carpeta que contiene las imágenes e iconos empleados en la aplicación web. Los iconos se han obtenido de forma gratuita a través de la web [www.flaticon.com](http://www.flaticon.com).

**Script.js:** script que contiene todas las definiciones de las funciones que llevan a cabo partes de la funcionalidad del código. Es decir, agrupa todas las funciones relativas al registro y eliminación de profesores, asignaturas, alumnos, exámenes y mensajes. Asimismo, también contiene las funciones asociadas al control de errores, gestión de login y sesión, activación y desactivación de examen y gestión de clicks.

**verNotas.php:** codificación llevada a cabo para visualizar las notas de los exámenes ya realizados por los alumnos desde el módulo de profesores.

**Profesores.php:** script que codifica la página principal del menú de profesores dentro del módulo de Administración.

**preguntasExamen.php:** script que genera el formulario para la inclusión de preguntas en un examen.

**Mensajes.php:** script que codifica la página principal de gestión de mensajes entre profesores y alumnos. Se accede pinchando en el sobre de la parte superior derecha.

**Index.php:** codificación de la página de acceso a la aplicación web, donde se muestran cajas de texto para introducir nick de usuario y contraseña.

**Home.php:** codificación de la página principal de la aplicación web una vez se ha accedido desde index a través de una cuenta de Administrador.

**gestionExamen.php:** página que muestra los exámenes creados para una cierta asignatura junto a su información asociada, con posibilidad de activar o desactivarlos.

**Error.php:** tras intentar acceder introduciendo un nombre de usuario o contraseña incorrecto, se redirecciona al usuario a una página de error.

**crearExamen.php:** tras seleccionar una asignatura, se puede acceder a esta página en caso de querer crear un examen. Para ello, se deberá introducir la información solicitada a través de un formulario.

**asignaturasAsignadas.php:** una vez se ha accedido al módulo de profesores mediante una cuenta de profesor autorizada, la página principal de este módulo se genera mediante este script. Existe un menú desplegable con todas las asignaturas asociadas al profesor en concreto, así como botones para consultar notas, crear exámenes y activar o desactivar los exámenes ya creados.

**Asignaturas.php:** script que codifica la página principal del menú de asignaturas dentro del módulo de Administración.

**Alumnos.php:** script que codifica la página principal del menú de alumnos dentro del módulo de Administración.

**panelUsuario.php:** caja contenedora del menú desplegable de la información de usuario.

**Login.php:** script que recoge los valores del formulario de logeo y maneja si se debe iniciar sesión accediendo al módulo de Administración o al de Profesorado.

**Head.php:** cabecera de la aplicación web.

**Funciones.php:** script que contiene la definición de dos funciones. Una de ellas es la que se encarga de validar si la dirección de email que se ha introducido en el formulario es una dirección de email con formato correcto. La otra función se encarga de leer cada línea del fichero de carga masiva de alumnos.

**conexionBD.php:** script que se encarga de llevar a cabo la conexión con el servidor de Hostinger así como con la base de datos.

**cerrarLogin.php:** script que simplemente cierra la sesión. Se accede al tras pinchar en el botón de cerrar sesión.

**Cabecera.php:** otro script de la cabecera de la web.

**cargarProfesores.php:** se conecta a la base de datos y se realiza una consulta, la cual devolverá todos los profesores de la base de datos, expresando en un menú el total de profesores que contiene la base de datos en ese momento.

**comprobarPassword.php:** tras conectarnos a la base de datos, verifica la contraseña y si el email ya se asignó a otro profesor.

**eliminarProfesor.php:** se encarga de eliminar de la BD el profesor/es seleccionado/s.

**registrarProfesor.php:** se encarga de registrar un nuevo profesor en la BD tras rellenar el formulario correspondiente.

**cargarMensajes.php:** se conecta a la base de datos y se realiza una consulta, la cual devolverá todos los mensajes de la base de datos, expresando en un menú el total de mensajes que contiene la base de datos en ese momento para el profesor seleccionado.

**formEnviarMensaje.php:** script que genera el formulario de envío de mensajes por parte de profesor a alumnos.

**eliminarMensaje.php:** se encarga de eliminar de la BD el mensaje/s seleccionado/s.

**enviarMensaje.php:** se encarga de enviar mensajes a los alumnos, registrándolos en la base de datos tras haber rellenado todos los campos del formulario de mensaje.

**activarDesactivarExamen.php:** script que lleva a cabo la activación o desactivación de los exámenes seleccionados.

**cargarExámenes.php:** tras conectarse a la BD, muestra en un menú todos los mensajes que contiene la base de datos en ese momento creados por el profesor con la sesión abierta.

**generarPreguntas.php:** función que genera el formulario de inserción de preguntar cuando se está creando un examen.

**insertarExamen.php:** función que recoge todas las variables que se reciben del formulario de examen.

**insertarPreguntasExamen.php:** en función del número de preguntas introducido en el formulario de examen, se genera dicho número de preguntas con sus respectivas respuestas en el examen.

**cargarAsignaturas.php:** tras acceder a la lista de asignaturas, devuelve todas las asignaturas existentes en la base de datos.

**cargarListaAsignaturasPorProfesor.php:** script para la codificación del menú desplegable del módulo de Profesorado donde aparecen todas las asignaturas asignadas al profesor con sesión abierta.

**cargarListaProfesores.php:** tras acceder a la lista de profesores, devuelve todos los profesores existentes en la base de datos.

**cargarNombreAsigPorID.php:** tras acceder a la base de datos, carga en la web todas las filas con asignaturas.

**eliminarAsignatura.php:** script creado para llevar a cabo la eliminación de asignaturas desde el módulo de Administración.

**opcionAsignatura.php:** script para la codificación del menú desplegable del módulo de Administración donde aparecen todas las asignaturas existentes.

**registrarAsignatura.php:** recoge y procesa las variables que se reciben del formulario de inserción de nuevas asignaturas desde el módulo de Administración.

**cargarAlumnos.php:** código que devuelve todos los alumnos de la base de datos, colocando todas las entradas por filas.

**cargarTablaAlumnosPorAsignatura.php:** código que coloca en una tabla los alumnos que posee una asignatura en concreto.

**eliminarAlumnos.php:** código encargado de eliminar alumnos del listado total dentro del módulo de Administración.

**insertarAlumnosFichero.php:** tras conectarnos a la base de datos llamando al script conexionBD.php, recoge las variables que se reciben de la lectura del fichero de alumnos, comprobando que no falte ningún campo.

### 4.1.3 Desarrollo web

Se decidió comenzar con el desarrollo web antes que con el de la parte Android ya que se consideraba más intuitivo y beneficioso para, puesto que se trata de la parte más grande al abarcar dos módulos y la parte Android se nutre de más información de la parte web que al contrario.

En este apartado se van a mostrar algunos diagramas para facilitar el entendimiento de lo que se ha desarrollado en la codificación de software. Para ello, en primer lugar se explicará qué son los casos de uso.

En el contexto de ingeniería del software, un caso de uso es una secuencia de interacciones que se desarrollarán entre un sistema y sus actores en respuesta a un evento que inicia un actor principal sobre el propio sistema. Los diagramas de casos de uso sirven para especificar la comunicación y el comportamiento de un sistema mediante su interacción con los usuarios y/u otros sistemas. O lo que es igual, un diagrama que muestra la relación entre los actores y los casos de uso en un sistema. Una relación es una conexión entre los elementos del modelo, por ejemplo la especialización y la generalización son relaciones. Los diagramas de casos de uso se utilizan para ilustrar los requerimientos del sistema al mostrar cómo reacciona a eventos que se producen en su ámbito o en él mismo.

Un tema que genera mucha polémica entre la gente que modela casos de uso es la elección entre la relación de <<include>> y <<extend>>. Lo peor es que muchas de esas discusiones generan muy poco valor en el resultado final en el modelo y en cambio quitan tiempo valioso del proyecto. Esto se debe a que dichas relaciones, muchas veces no son del todo comprendidas por la persona que la modela, y mucho menos son comprendidas por las personas que leen el modelo. Así que al final no se le saca el provecho que en todo caso debería de tener dicha elección.

Cuando relacionamos dos casos de uso con un “include”, estamos diciendo que el primero (el caso de uso base) incluye al segundo (el caso de uso incluido). Es decir, el segundo es parte esencial del primero. Sin el segundo, el primero no podría funcionar bien; pues no podría cumplir su objetivo.

En el caso del “extend” hay situaciones en que el caso de uso de extensión no es indispensable que ocurra, y cuando lo hace ofrece un valor extra (extiende) al objetivo original del caso de uso base. En cambio en el “include” es necesario que ocurra el caso incluido, tan sólo para satisfacer el objetivo del caso de uso base. Por tanto, “extend” se utiliza cuando un caso de uso base incorpora el comportamiento de otro caso de uso y “extiende” su funcionamiento.

## Administrador

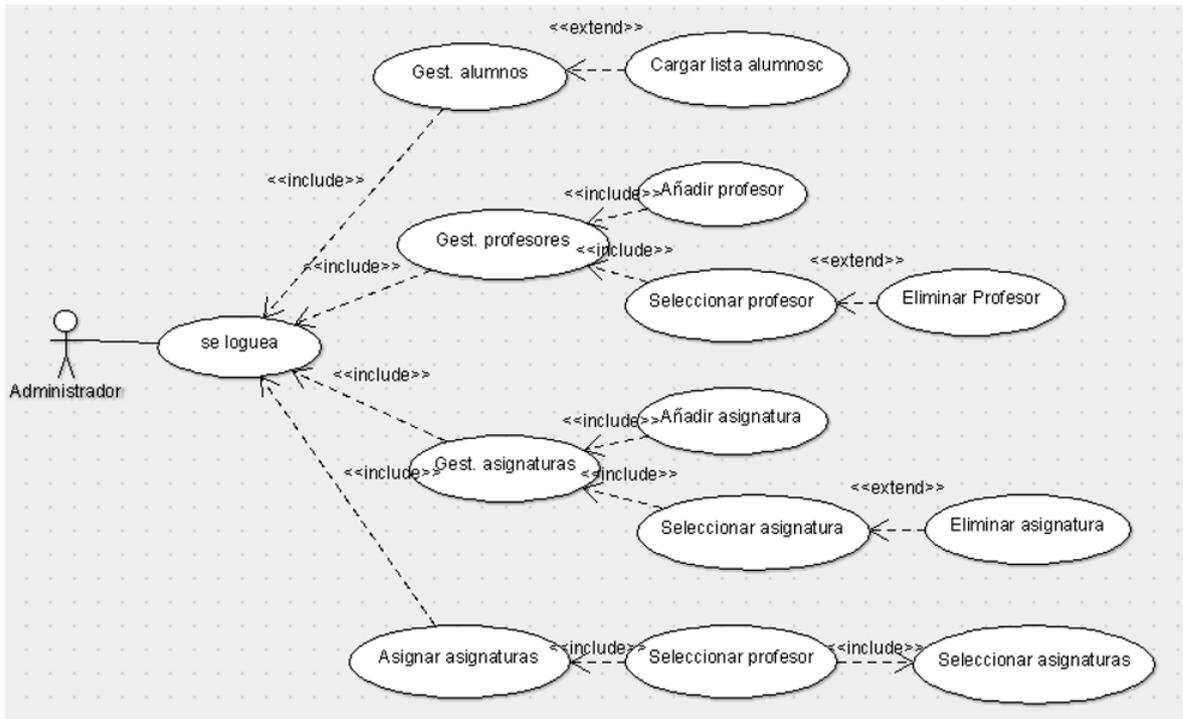


Figura 37: Casos de Uso Módulo Administración

## Profesor

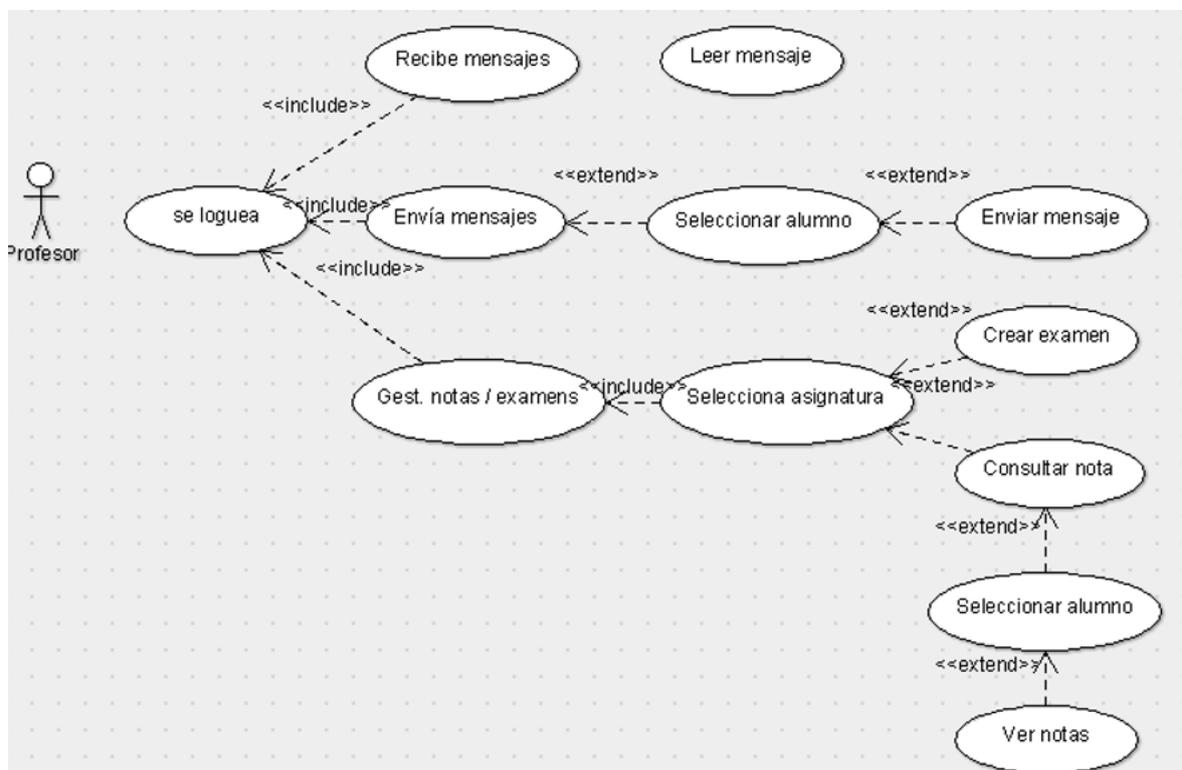


Figura 38: Casos de Uso Módulo Profesorado

Un diagrama o modelo entidad-relación es una herramienta para el modelado de datos que permite representar las entidades relevantes de un sistema de información así como sus interrelaciones y propiedades. El modelo de datos entidad-relación está basado en una percepción del mundo real que consta de una colección de objetos básicos, llamados entidades, y de relaciones entre esos objetos.

**Entidad:** Representa una “cosa” u "objeto" del mundo real con existencia independiente, es decir, se diferencia únicamente de otro objeto o cosa, incluso siendo del mismo tipo, o una misma entidad. Una entidad está descrita y se representa por sus características o atributos. Por ejemplo, la entidad Persona las características: Nombre, Apellido, Género, Estatura, Peso, Fecha de nacimiento.

**Atributos:** Los atributos son las características que definen o identifican a una entidad. Estas pueden ser muchas, y el diseñador solo utiliza o implementa las que considere más relevantes. En un conjunto de entidades del mismo tipo, cada entidad tiene valores específicos asignados para cada uno de sus atributos, de esta forma, es posible su identificación unívoca. Cada una de las entidades pertenecientes a este conjunto se diferencia de las demás por el valor de sus atributos. Nótese que dos o más entidades diferentes pueden tener los mismos valores para algunos de sus atributos, pero nunca para todos.

**Relación:** Describe cierta dependencia entre entidades o permite la asociación de las mismas.

Ahora, se muestra el diagrama entidad-relación creado durante la fase de planificación del proyecto para clarificar cómo debía ser la organización durante la fase de desarrollo:

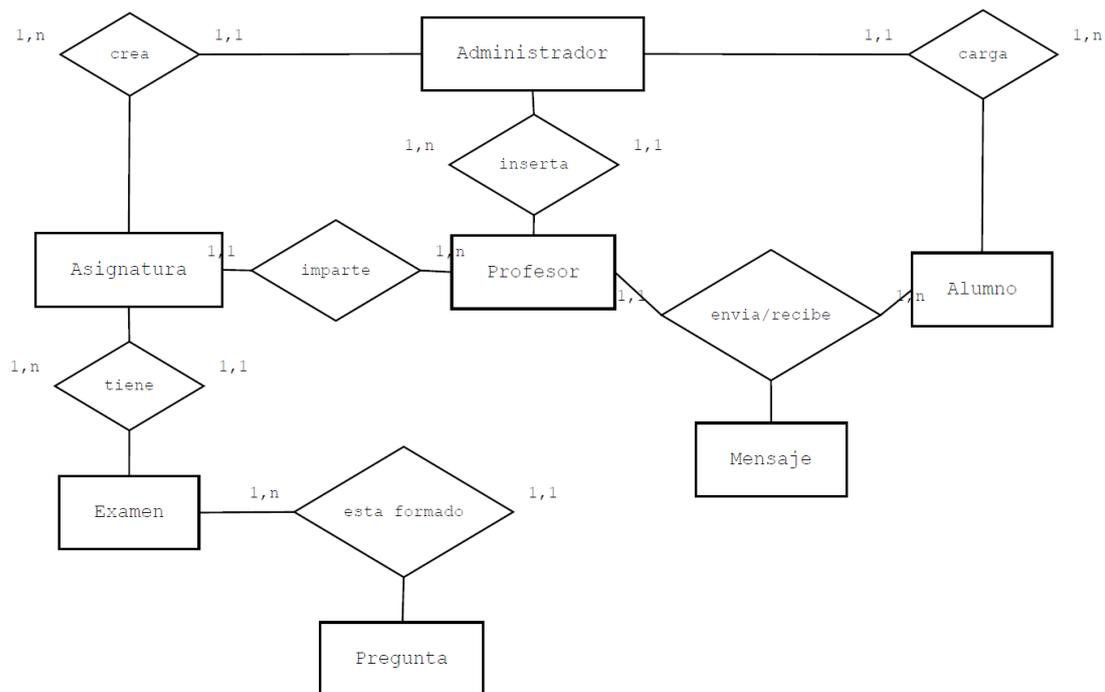


Figura 39: Modelo Entidad-Relación Aplicación Web

Para continuar aclarando las pautas y especificaciones que se han desarrollado en los distintos lenguajes de programación web, ahora se mostrará el modelo relacional generado para la parte web de la aplicación.

En este modelo todos los datos son almacenados en relaciones, y como cada relación es un conjunto de datos, el orden en el que éstos se almacenen no tiene relevancia (a diferencia de otros modelos como el jerárquico y el de red). Esto tiene la considerable ventaja de que es más fácil de entender y de utilizar por un usuario no experto. La información puede ser recuperada o almacenada por medio de consultas que ofrecen una amplia flexibilidad y poder para administrar la información.

Este modelo considera la base de datos como una colección de relaciones. De manera simple, una relación representa una tabla que no es más que un conjunto de filas, cada fila es un conjunto de campos y cada campo representa un valor que interpretado describe el mundo real. Cada fila también se puede denominar tupla o registro y a cada columna también se le puede llamar campo o atributo.

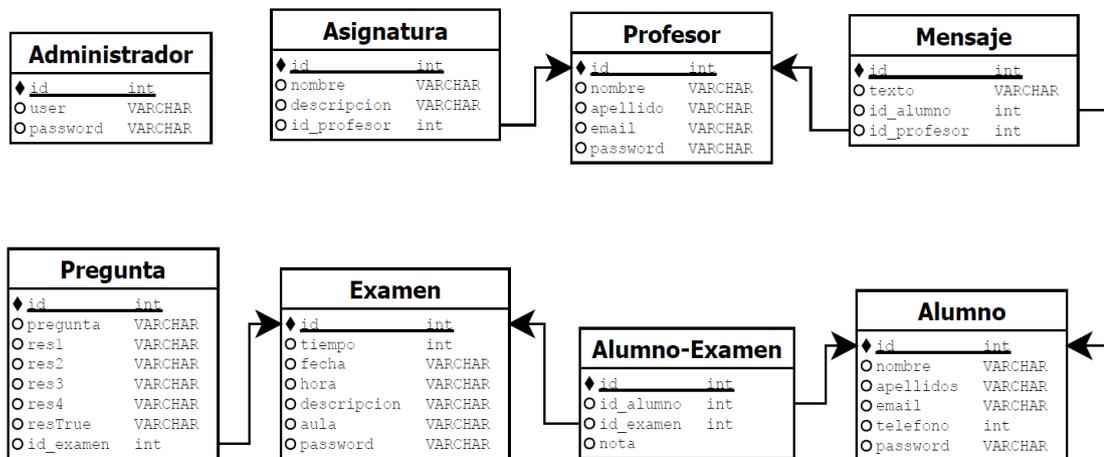


Figura 40: Modelo Relacional Aplicación Web

## **4.2 Aplicación Android**

La programación de aplicaciones en Android no es trivial, requiere de conocimiento previo acerca de programación. Es decir, cuanta más experiencia posea el programador, más sencillo le resultará desarrollar código en Android.

En cuanto a lenguajes, lo ideal es haber programado anteriormente en Java, el lenguaje en el que se desarrollan las aplicaciones nativas de Android. No obstante, si nunca se ha escrito código en este lenguaje (como el caso del alumno que lleva a cabo este PFC) pero se tienen conocimientos de otros lenguajes orientados a objetos, es posible relacionar mejor los contenidos y disponer de las nociones necesarias para desarrollar la aplicación.

Por ello, el primer paso que se ha llevado a cabo por parte del alumno al iniciar el PFC, fue una ardua labor documental. Consultando la documentación y referencias incluidas en el apartado de bibliografía, se consiguió adquirir el conocimiento requerido sobre fundamentos de programación en este lenguaje. De esta forma, tras un amplio estudio de Android, se comprendió la filosofía de los lenguajes orientados a objetos y sus conceptos clave como abstracción, herencia, encapsulamiento, polimorfismo, etc.

Así las cosas, es importante recalcar el esfuerzo del alumno en la labor documental y de estudio de Android de forma autónoma e individual, ya que no se cursó la asignatura optativa de Programación Orientada a Objetos, donde se impartían estos conceptos. No obstante, ya que este proyecto se fundamenta en el objetivo académico de aprender aspectos nuevos no estudiados durante la carrera universitaria, dado que todo lo que se ha hecho es nuevo, durante todo el desarrollo del proyecto se ha seguido consultando documentación y el aprendizaje ha sido continuado desde principio a fin.

De todas formas, a pesar de haber consultado multitud de referencias y tutoriales, durante el desarrollo de un proyecto siempre surgen dudas y problemas. Para la solución de estas cuestiones y así poder avanzar de forma satisfactoria, se ha consultado Google de forma activa.

En la siguiente página, se describen las herramientas empleadas en el entorno de desarrollo para programar en Android.

- **Android Studio:** Se trata de un Entorno de Desarrollo Integrado (IDE) para la plataforma Android. Se trata de un IDE relativamente nuevo que, debido a su funcionalidad y enfoque, ha sustituido rápidamente a Eclipse como entorno de desarrollo para programación en Android.
- **Android SDK Manager:** gestor de actualizaciones de herramientas. Permite la instalación o desinstalación de nuevas versiones, APIs, paquetes o herramientas.
- **Android Virtual Device Manager:** software que permite ejecutar la aplicación desarrollada en un ordenador, haciendo que funcione como si fuera un dispositivo móvil (Smartphone, Tablet o Wearable). Muy útil para testear el correcto funcionamiento de la aplicación cuando no se dispone de un terminal móvil a mano.
- **Máquina Virtual de Java:** software que se emplea para ejecutar código Java en el equipo. Esta máquina se conoce como entorno de ejecución de Java JRE (Java Runtime Environment).

### 4.2.1 Primeros pasos

En este apartado se van a explicar los pasos previos y acciones necesarias llevadas a cabo para desarrollar la aplicación en Android.

Tras instalar Android Studio, se instalarán las librerías y herramientas de los dueños, es decir, el SDK (Software Development Kit) de Android. Con ello tendremos habilitado el camino para generar nuevas aplicaciones en consola. Asimismo, es preciso instalar la máquina virtual de Java, conocida como JDK (Java Development Kit). En ocasiones, una vez instalado el entorno completo se requiere registrar una nueva variable de entorno para indicarle a Android Studio, donde se aloja la máquina virtual. Vamos a explicar esto con imágenes.

Una vez abierto Android Studio, se accede a una pantalla principal de bienvenida. Procedemos a crear un proyecto nuevo:



*Figura 41: Página Inicio Android Studio*

Para ello, se deberá introducir el nombre de la aplicación, el dominio de tu empresa o marca y el nombre del paquete. Después, aparecerá otra pantalla donde se seleccionarán los factores de forma (phone and tablet en nuestro caso) y el SDK mínimo que soportará la aplicación. Por defecto, se instala la última versión de la plataforma, Android Lollipop en nuestro caso. Luego, se debe elegir una actividad (patrón de ejecución visual que se relacionará con las entradas del usuario). Finalmente, se elige el nombre de la actividad, layout y el título. Una vez establecidos todos estos parámetros, se accede al entorno de desarrollo de Android Studio.

Pinchando en el icono de SDK Manager, podremos informarnos de qué complementos se encuentran instalados en nuestra máquina y cuáles no.

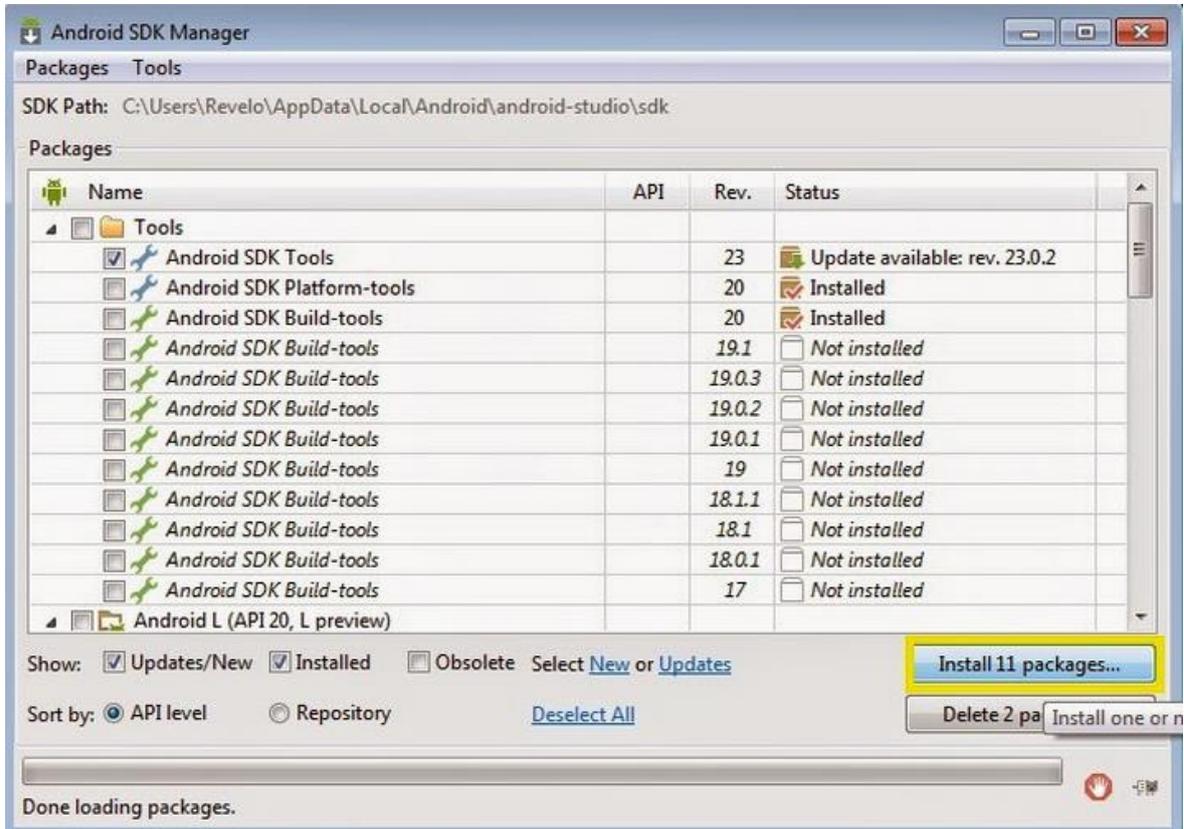


Figura 42: Android Studio SDK Manager

Por otro lado, se encuentra la opción de AVD (Android Virtual Device). AVD es una máquina virtual que ejecuta la plataforma Android seleccionada, para de este modo poder comprobar la funcionalidad de la aplicación construida. La lista de modelos disponibles donde emular la aplicación es bastante grande. Cabe señalar que la versión de Android del proyecto debe ser la misma en el AVD, de lo contrario, Android Studio reportará un error de incompatibilidad de SDKs.

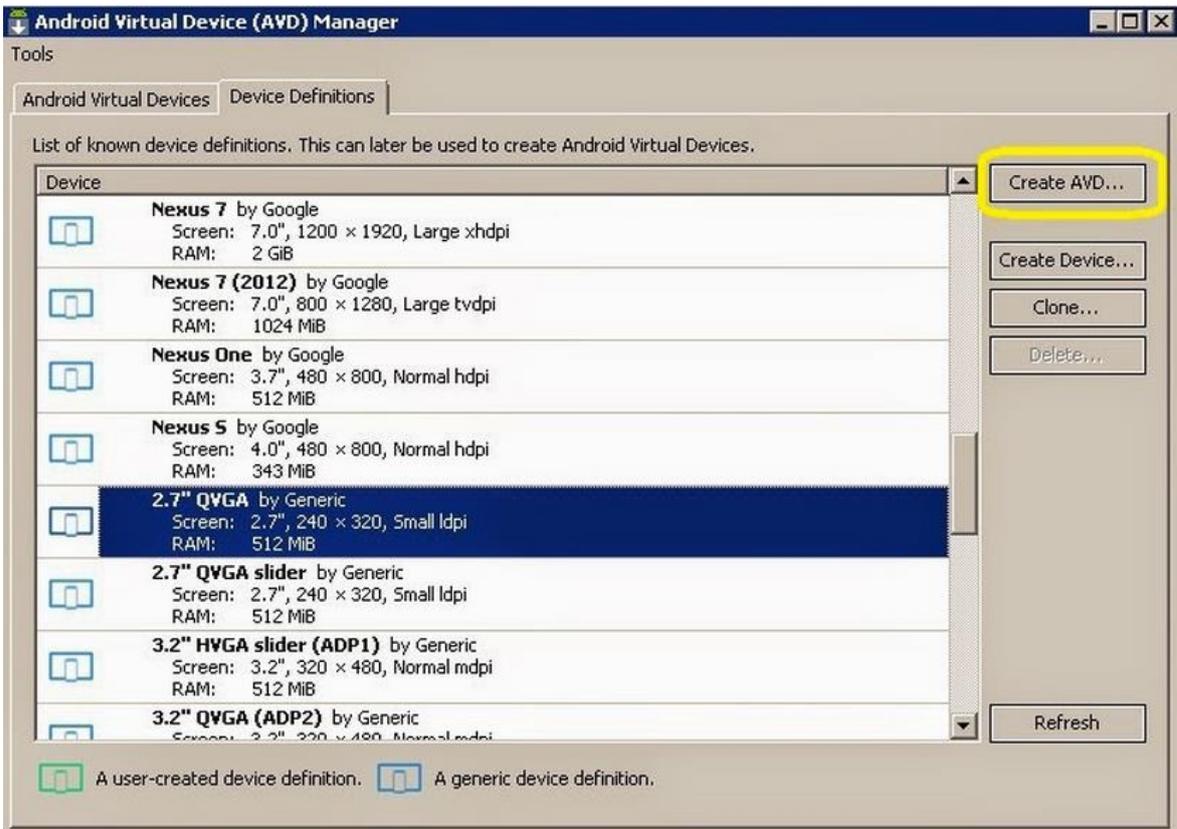


Figura 43: Android Virtual Device Manager

Tras cargar el AVD, se ejecutará en una ventana adicional:



Figura 44: Android Virtual Device

Finalmente, para ejecutar la aplicación en Android Studio, simplemente hay que dirigirse al editor y presionar en el botón “Run app”, eligiendo el AVD recién iniciado.

Enumeramos algunas recomendaciones para acelerar el AVD:

- Instalar el acelerador de dispositivos virtuales de Android si se está usando arquitectura Intel.
- Usar versiones antiguas de Android, relativas a dispositivos antiguos y/o de gama baja.
- Poseer un equipo bien equipado en cuanto a SO, RAM y memoria que pueda hacer frente de manera fácil a la ejecución.
- Usar el propio dispositivo móvil Android en modo depuración USB.

## 4.2.2 Conceptos Clave

En este epígrafe se van a estudiar distintos conceptos relativos a la programación en Android:

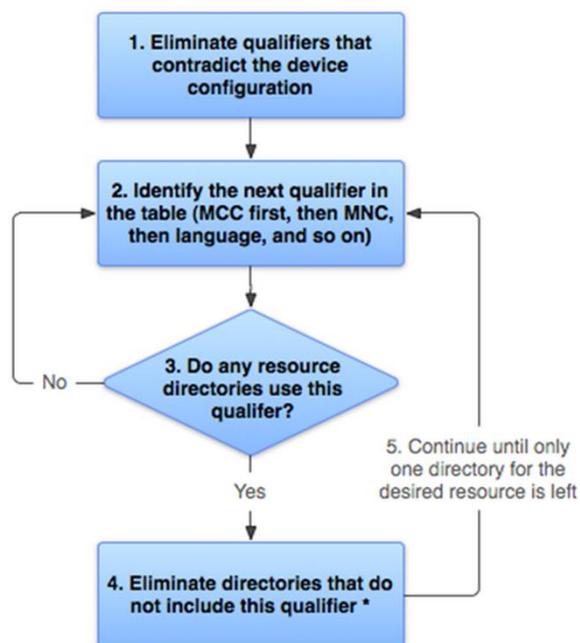
### Recursos

Siempre conviene externalizar los recursos del código, entendiendo por recursos imágenes, textos, estilos... De esta forma, también es posible especificar diferentes recursos dependiendo del tipo de dispositivo que se utilice, sin necesidad de modificar el código.

Los recursos deben alojarse bajo una estructura de carpetas específica. Todos los recursos irán dentro de la carpeta /res. Se destacan los siguientes recursos:

- Animaciones
- Colores
- Imágenes (Drawables)
- Menús
- Estilos
- Cadenas de texto (Strings)
- Layouts

A continuación, se puede ver un diagrama de flujo de cómo opera el SO Android para elegir el recurso adecuado:



\* If the qualifier is screen density, the system selects the "best match" and the process is done

Figura 45: Android Resources

## Clase Activity

Entendemos como actividad a una clase de Java que representa cada una de las pantallas de una aplicación con las que el usuario puede interactuar. Las actividades están formadas por una parte lógica y una parte gráfica. La parte lógica se trata de un archivo .java, que contiene la clase que permite manipular, interactuar y desarrollar el código de esa actividad. Por otro lado, la parte gráfica es un archivo .xml (también conocido como layout) en el que se declaran todos los elementos que se mostrarán por pantalla.

Para crear una actividad, simplemente hay que crear una clase que herede de la clase matriz “Activity” así como sobrescribir algunos de los métodos de su ciclo de vida. Asimismo, desde una actividad se pueden lanzar otras actividades. Para organizar las actividades, se emplea una cola de tipo LIFO (Last Input First Output). Por tanto, al pulsar el botón de atrás, se cierra la actividad actual y se accede a la actividad anterior. Todas las actividades creadas deben registrarse en el fichero “Manifest” para que se mantengan operativas.

Desde el momento que una actividad aparece en la pantalla hasta que desaparece pasa por varias etapas, lo que se conoce como ciclo de vida de una actividad. A continuación se muestra un diagrama explicativo de este ciclo de vida:

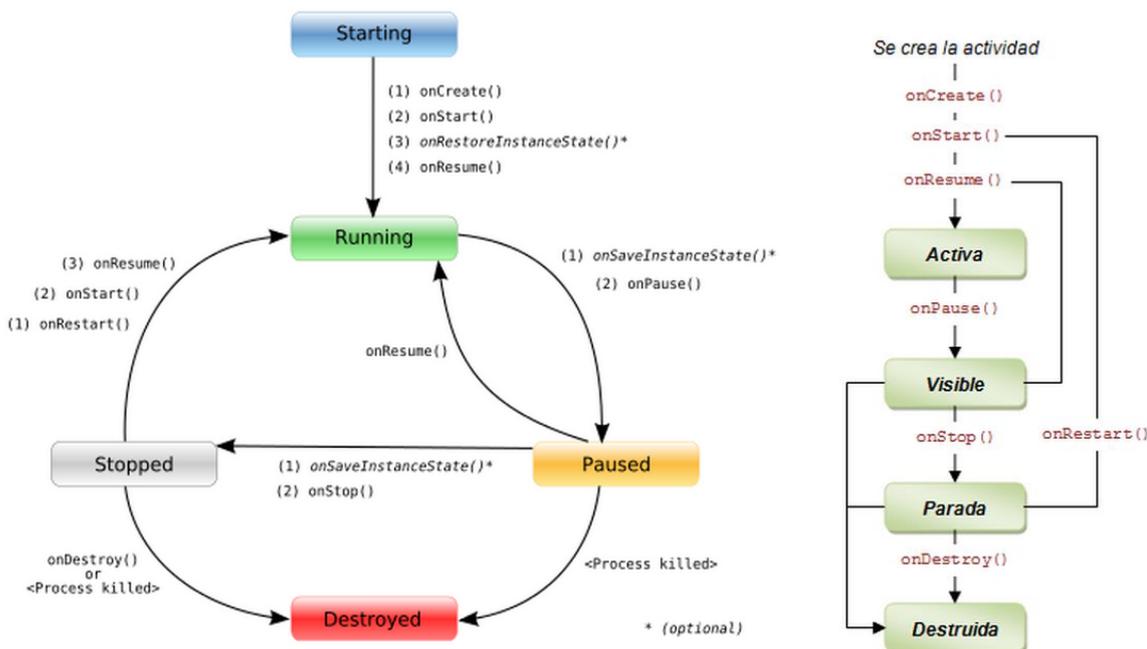


Figura 46: Diagrama Clase Activity

A continuación, se explican la casuística de cada estado de una actividad:

- **Activa:** Estado en el que se encuentra la aplicación cuando una actividad que se está ejecutando en primer plano.
- **Visible:** Una actividad que todavía es visible pero que no está en primer plano, está en pausa. Podría ser finalizada por el sistema si se necesitase memoria.
- **Parada:** Una actividad que ha sido eclipsada por otra actividad, está en stop. Aún conservará todos los estados y la información, pero no será visible por el usuario. Por ello, en caso de necesitar memoria, el sistema podría decidir terminar con ella.
- **Destruída:** Es posible eliminar del sistema una actividad que se encuentre en estado de stop o de pausa. Cuando se vuelva a mostrar al usuario el sistema tendrá que reiniciarla o restaurarla al estado anterior.

Ahora, se analizarán algunos de los métodos vistos en los diagramas:

- **onCreate(Bundle):** Se llama en la creación de la actividad. Se utiliza para realizar todo tipo de inicializaciones, como la creación de la interfaz de usuario o la inicialización de estructuras de datos. Puede recibir información de estado de la actividad (en una instancia de la clase Bundle), por si se reanuda desde una actividad que ha sido destruida y vuelta a crear.
- **onStart():** Nos indica que la actividad está a punto de ser mostrada al usuario.
- **onResume():** Se llama cuando la actividad va a comenzar a interactuar con el usuario. Es un buen lugar para lanzar las animaciones y la música.
- **onPause():** Indica que la actividad está a punto de ser lanzada a segundo plano, normalmente porque otra actividad es lanzada. Es el lugar adecuado para detener animaciones, música o almacenar los datos que estaban en edición.
- **onStop():** La actividad ya no va a ser visible para el usuario. En el caso de que haya muy poca memoria libre, es posible que la actividad se destruya sin llamar a este método.
- **onRestart():** Indica que la actividad va a volver a ser representada después de haber pasado por onStop().
- **onDestroy():** Se llama antes de que la actividad sea totalmente destruida. Por ejemplo, cuando el usuario pulsa el botón de volver o cuando se llama al método finish(). Cuando hay muy poca memoria, es posible que la actividad se destruya sin llamar a este método.

Como se observa en el diagrama de estados, cada vez que se inicializa una actividad, se invoca a la función “OnCreate()”. A partir de ahí, se enlazan las partes lógica y gráfica a través de las funciones “SetContentView( )” y “findViewById( )” respectivamente.

## **Fichero Manifest**

Se trata del fichero más relevante de todos, ya que está situado en la raíz de nuestras aplicaciones. Por tanto, **AndroidManifest.xml**, es un archivo de configuración donde podemos aplicar las configuraciones básicas de nuestra aplicación. Su configuración puede realizarse a través de una interfaz gráfica, pero es recomendable conocer la sintaxis ya que en muchas ocasiones será más fácil y rápido hacerlo desde el propio xml. Contiene la siguiente información:

- Nombre de paquete Java como identificador de la aplicación.
- Descripción los componentes de la aplicación: actividades, servicios, proveedores de contenido, etc. Esto permite al sistema operativo conocer que componentes tiene y bajo qué condiciones pueden ser lanzados.
- Especifica que permisos tiene la aplicación para acceder a partes protegidas del API que proporciona el sistema Android.
- Declara la mínima y máxima versión de Android para la que funcionará la aplicación.
- Indica las librerías que utiliza el proyecto y por lo tanto tienen que ser empaquetadas al crear la aplicación.
- Hardware y software empleado
- Permisos necesarios para la aplicación (cámara, GPS, ...)

Todas y cada una de las actividades de la app deben ser declaradas en esta sección, sino no funcionarán. La actividad inicial de la aplicación debe ser declarada de tipo "LAUNCHER":

## **Views (Vistas)**

La clase View contiene los componentes de la interfaz de usuario. Ocupa un área rectangular en la pantalla y es responsable de dibujar y manejar los eventos de una aplicación. Se trata de la clase básica de los widgets, los cuales se usan para crear componentes interactivos como botones, campos de texto, etc.

## Mensajes Toast

Un “toast” es un mensaje que se muestra en pantalla durante unos segundos al usuario para luego volver a desaparecer automáticamente sin requerir ningún tipo de actuación por su parte, y sin recibir el foco en ningún momento (o dicho de otra forma, sin interferir en las acciones que esté realizando el usuario en ese momento).

Aunque son personalizables, aparecen por defecto en la parte inferior de la pantalla, sobre un rectángulo gris ligeramente translúcido. Por sus propias características, este tipo de notificaciones son ideales para mostrar mensajes rápidos y sencillos al usuario, pero por el contrario, al no requerir confirmación por su parte, no deberían utilizarse para hacer notificaciones demasiado importantes.

Su utilización es muy sencilla, concentrándose toda la funcionalidad en la clase Toast. Esta clase dispone de un método estático `makeText()` al que deberemos pasar como parámetro el contexto de la actividad, el texto a mostrar, y la duración del mensaje, que puede tomar los valores `LENGTH_LONG` o `LENGTH_SHORT`, dependiendo del tiempo que queramos que la notificación aparezca en pantalla. Tras obtener una referencia al objeto Toast a través de este método, ya sólo nos quedaría mostrarlo en pantalla mediante el método `show()`.

### 4.2.3 Tipos de archivos

El panel de la parte izquierda de la interfaz de Android Studio, llamado “Project”, representa la estructura del proyecto. Como notas, están las carpetas `.idea`, `app`, `build`, `gradle` y otros archivos de configuración.

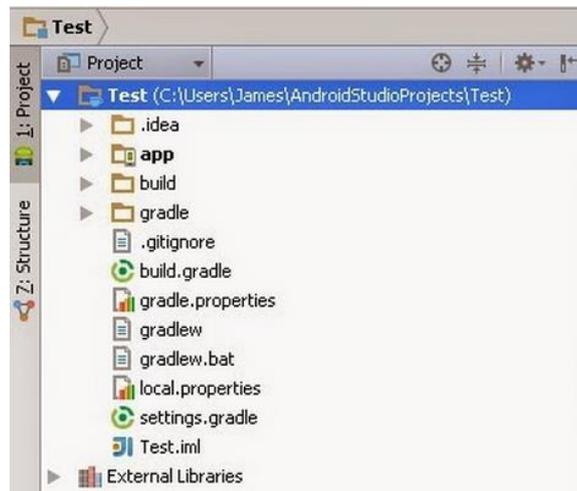


Figura 47: Estructura Proyecto Android Studio

La carpeta `app` es la que contiene todo lo relacionado con el proyecto, es la carpeta que nos interesa por el momento y donde incluiremos los archivos necesarios para que nuestra aplicación sea empaquetada. Si despliegas su contenido veras tres carpetas: `build`, `libs` y `src`. Por ahora ignoraremos los demás archivos.

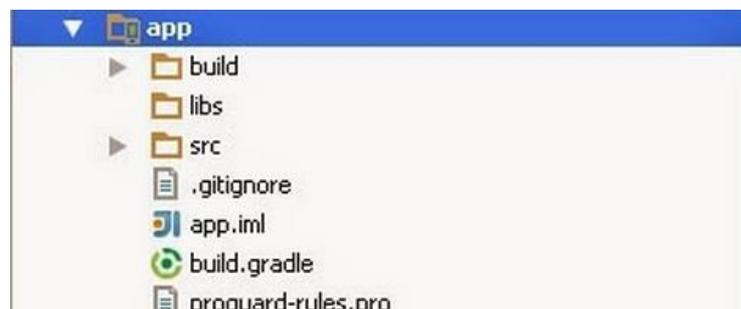


Figura 48: Carpeta `app` Android Studio

Normalmente la mayor parte del tiempo y fuerzas las usaremos en la carpeta `src` (Source). Dentro de ella se ubica la carpeta `main`, la cual contiene todos los archivos fuente Java para nuestra aplicación. La carpeta `res` (Resources) que contiene los recursos del proyecto (iconos, sonido, diseños, etc.) y el archivo `AndroidManifest.xml`, explicado anteriormente.

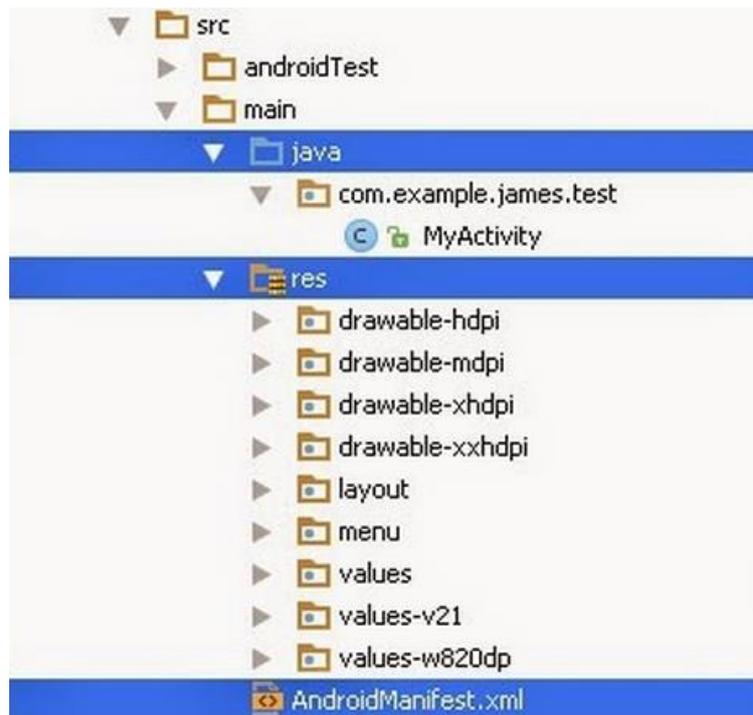


Figura 49: Carpeta java Android Studio

En la carpeta *java* se alojan todos los archivos relacionados con nuestras actividades y otros archivos fuente auxiliares.

Dentro de la carpeta *res* encontraremos todos aquellos recursos “tercerizados” para nuestra aplicación. Esta práctica de excluir los atributos de la aplicación a través de archivos externos, permite reducir la complejidad de diseño en las interfaces.

Uno de los recursos más relevantes es el archivo *strings.xml* que se encuentra dentro de la subcarpeta *values*. Este fichero almacena todas las cadenas que se muestran en los widgets (controles, formas, botones, vistas, etc.) de nuestras actividades.

El archivo *strings.xml* es muy útil para los desarrolladores. Una de sus grandes utilidades es facilitar el uso de múltiples idiomas en tu aplicación. Esto se debe a que puedes externalizar las cadenas del código java y seleccionar la versión del archivo *strings.xml* con el lenguaje necesitado.

La carpeta *layout* contiene los archivos de diseño de todas las actividades del proyecto. En la figura de ejemplo, se encuentra el archivo *activity\_my.xml*. Este archivo representa el diseño de la interfaz de la actividad principal. En él, se establecen todos los widgets que se agreguen a la actividad.

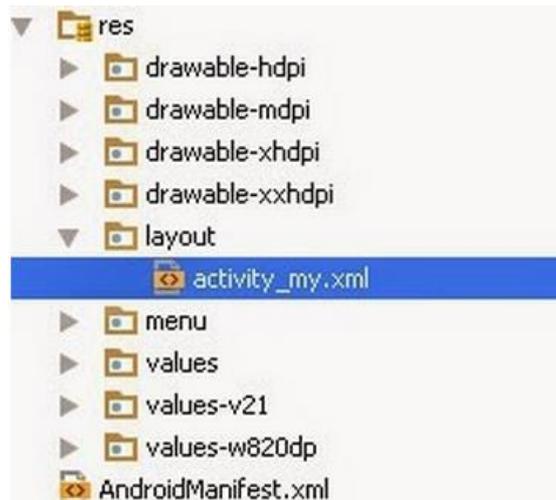


Figura 50: Carpeta res Android Studio

Construir la interfaz a través de nodos XML es mucho más sencillo que la creación a través de código Java. Adicionalmente, Android Studio posee un panel de diseño estilo Drag and Drop, lo cual es una bendición para los desarrolladores, ya que facilita demasiado la creación de una interfaz de usuario.

Este archivo de diseño comienza con un nodo raíz llamado `<RelativeLayout>`. Un Layout es el contenedor principal que define el orden y secuencia en que se organizarán los widgets en nuestra actividad. Existen varios tipos de Layouts, como por ejemplo `LinearLayout`, `GridLayout`, `FrameLayout`, etc.

Dentro de la carpeta *drawable*, se muestran las imágenes empleadas en la aplicación, relacionadas directamente con el tipo de densidad de pantalla donde se ejecutará la aplicación.

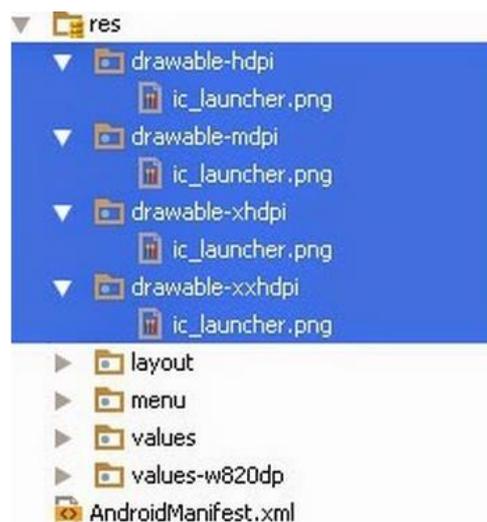


Figura 51: Carpeta drawable Android Studio

La mayoría de dispositivos móviles actuales tienen uno de estos 4 tipos de densidades:

- **Mediun Dots per Inch (mdpi):** Este tipo de pantallas tienen una densidad de 160 puntos por pulgada.
- **High Dots per Inch (hdpi):** En esta clasificación encontraremos teléfonos cuya resolución es de 240 puntos por pulgada.
- **Extra high dots per inch (xhdpi):** Resoluciones mayores a 340 puntos por pulgada
- **Extra Extra high dots per inch (xxhdpi):** Rangos de resoluciones mayores a 480 puntos por pulgada.

Visualicemos la siguiente imagen ilustrativa de densidades:



*Figura 52: Densidades Pantalla Android*

Lo ideal sería definir todos los recursos gráficos en los distintos tipos de densidades, puesto que esta diversidad permitirá una mayor compatibilidad con los distintos dispositivos móviles existentes.

#### 4.2.4 Desarrollo Android

Tras la fase de planificación, donde se fijan las restricciones de prestaciones, planificación temporal y costes, llega la fase de ejecución o desarrollo. Una planificación detallada da consistencia al proyecto y evita sorpresas que nunca son bien recibidas. La fase de desarrollo de un proyecto siempre suele ser la más costosa, no solo económicamente sino también en temporalmente.

En este caso, se decidió que la mejor aproximación era comenzar el desarrollo de la parte Android una vez se hubiera acabado el de la parte web, para alimentarse de su resultados. Para la codificación del software, se ha empleado Android Studio

Asimismo, con el objetivo de dejar más claro al lector de esta memoria lo que se ha llevado a cabo en código Android, se adjunta el caso de uso creado para un usuario de tipo Alumno:

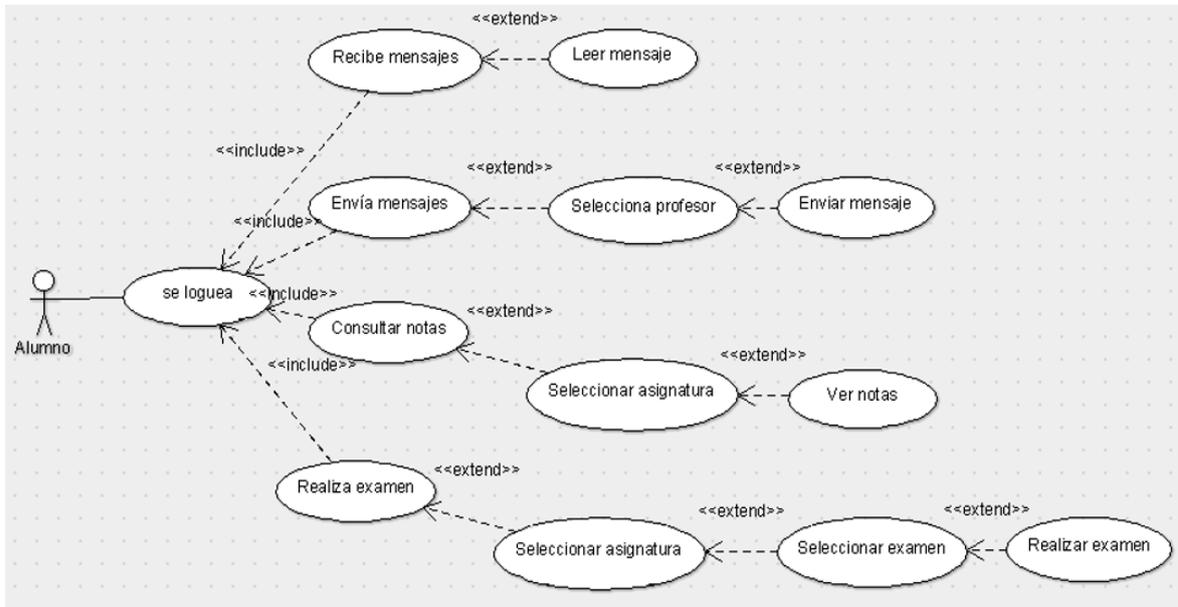


Figura 53: Casos de Uso del Módulo de Alumnos

Para conocer la funcionalidad y menús implementados en la aplicación Android, se describirá el esquema general de la aplicación mediante un diagrama de clases. No obstante, primero se realizará una breve explicación de lo que son las clases.

En informática, una clase es una plantilla para la creación de objetos de datos según un modelo predefinido. Las clases se utilizan para representar entidades o conceptos, como los sustantivos en el lenguaje. Cada clase es un modelo que define un conjunto de variables (el estado, y métodos apropiados para operar con dichos datos). Cada objeto creado a partir de la clase se denomina instancia de la clase.

Las clases son un pilar fundamental de la programación orientada a objetos. Permiten abstraer los datos y sus operaciones asociadas al modo de una caja negra. Los lenguajes de programación que soportan clases difieren sutilmente en su soporte para diversas características relacionadas con clases. La mayoría soportan diversas formas de herencia. Muchos lenguajes también soportan características para proporcionar encapsulación, como especificadores de acceso. Una clase también puede tener una representación (metaobjeto) en tiempo de ejecución, que proporciona apoyo en tiempo de ejecución para la manipulación de los metadatos relacionados con la clase.

Las clases se componen de elementos, llamados genéricamente “miembros”, de varios tipos:

- Campos de datos: almacenan el estado de la clase por medio de variables, estructuras de datos e incluso otras clases.
- Métodos: subrutinas de manipulación de dichos datos.
- Ciertos lenguajes permiten un tercer tipo de miembro: las “propiedades”, a medio camino entre los campos y los métodos.

Por tanto, un diagrama de clases es un tipo de diagrama de estructura estática que describe la estructura de un sistema mostrando las clases del sistema, sus atributos, operaciones (o métodos), y las relaciones entre los objetos. A continuación se incluye el diagrama de clases realizado para organizar la parte Android del proyecto:

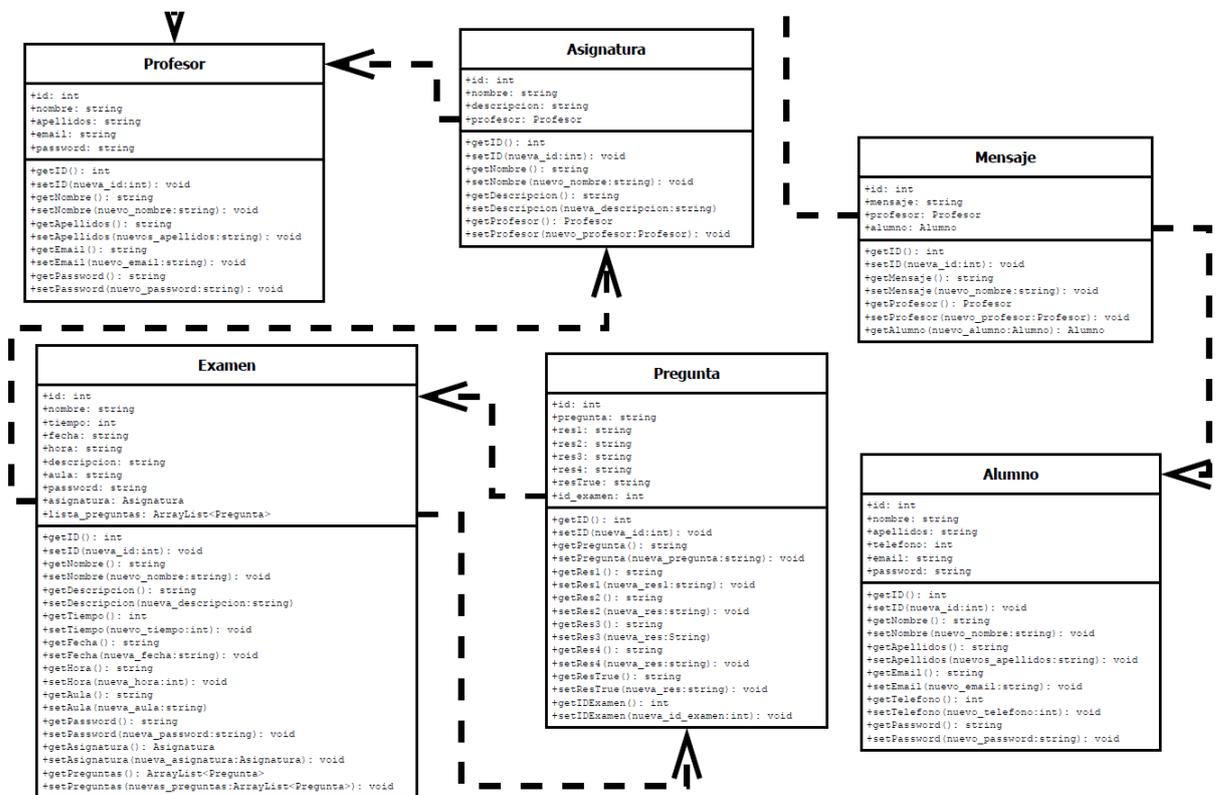


Figura 54: Diagrama de Clases Aplicación Android

Tanto la parte web como la parte Android comparten la misma base de datos. Ése es el nexo de unión que permite interactuar a ambas partes. Ambas partes harán consultas y editarán la base de datos alojada en el servidor web gratuito de Hostinger. Para el caso de la aplicación web, ya se ha explicado cómo se llevan a cabo estas consultas. A continuación, se explica el proceso por el cual la aplicación Android se comunica con la base de datos.

La aplicación Android, a través de un cliente, llevará a cabo peticiones GET a la dirección URL correspondiente del recurso a través del cliente `HttpURLConnection` con el fin de obtener datos. Este flujo entrante se interpretará con ayuda de un parser personalizado que implementa la clase `JsonReader`.

JSON (Javascript Object Notation) es un formato de intercambio de datos entre clientes y servidores, basado en la sintaxis de Javascript. En nuestro caso, JSON soportará el intercambio de datos entre el lenguaje PHP de la base de datos con el lenguaje Java de la aplicación Android. Se incluye una ilustración que muestra el proceso de parseo llevado a cabo:

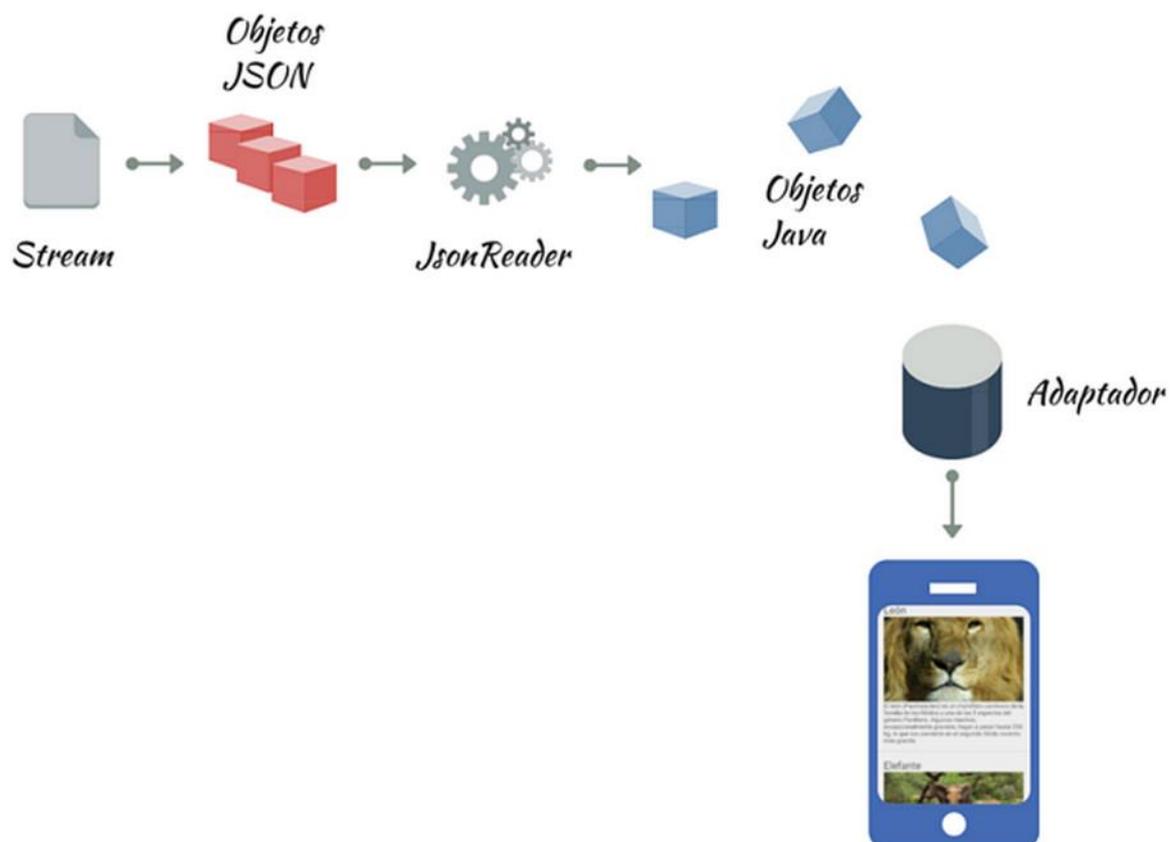


Figura 55: Parseo mediante JSON



## **5 Integración, pruebas y resultados**

---

Durante el desarrollo de la aplicación, se han llevado a cabo continuos testeos OAT (Operational Acceptance Testing). Esto implica que cada vez que se generaban nuevas partes del código, se ejecutaban en el navegador web o en el emulador Android para probar si su funcionalidad cumplía con los requisitos operacionales definidos en las especificaciones del PFC.

Asimismo, han sido muy importantes las pruebas de SIT (System Integration Testing). SIT es un proceso de testeo que prueba la coexistencia de un sistema de software con otros. Por ello, a medida que se iba desarrollando la parte Android y acoplándola a la aplicación web, se realizaban pruebas de comprobación para ver que ambas se coordinaban e interactuaban correctamente y su funcionalidad continuaba siendo la acordada.

Finalmente, previo a la puesta en producción, que en nuestro caso supondría la publicación de la aplicación en Android y la subida a un servidor web, se llevaron a cabo pruebas de aceptación de usuario, UAT (User Acceptance Testing). Por supuesto, a medida que se iban identificando fallos o errores, se corregían editando el código. A continuación, se incluyen capturas de pantalla con el resultado de dichas pruebas.

### **5.1 Aplicación web**

Para testear el correcto funcionamiento de la aplicación web, accedemos a la url de dicha web: <http://gestoexamen.esy.es/web/>

Como se ha comentado previamente, el proyecto consta de 3 Módulos (Administración, Profesores y Alumnos). Desde la parte web, acceden:

- Administración: añade y elimina alumnos, añade y elimina profesores, añade y elimina asignaturas y asigna profesores y alumnos a cada asignatura.
- Profesores: solo pueden acceder a las asignaturas que les asignó Administración. Pueden crear exámenes, programar cuándo activarlos y desactivarlos y consultar las notas de los alumnos tras el examen, así como la información relativa a cada asignatura y alumno.

Como usuario con acceso al módulo de Administración, se ha creado el usuario “admin” con contraseña “admin1”. Veremos que si se introduce admin como usuario y admin1 como contraseña y se hace click en “Entrar” pero no se selecciona la caja de “Admin”, el sistema no nos permite acceder



*Figura 56: Página Login Aplicación Web*

Se comprueba que salta el error y no permite entrar. De hecho, introduciendo un nombre de usuario y/o contraseña incorrecta, se accede a la misma pantalla de error, no siendo posible acceder a la aplicación.



*Figura 57: Página Error Aplicación Web*

Ahora, entraremos a la aplicación. Para ello, introducimos “admin” como usuario, “admin1” como contraseña y se marca la cajita de Admin:



¡Accede a tu cuenta!

Login

admin

.....

Admin:

Entrar

*Figura 58: Página Login con check de Admin Aplicación Web*

Comprobamos que hemos accedido con éxito a la pantalla principal de la aplicación (módulo de Administración):



Figura 59: Página Principal Administrador Aplicación Web

Pinchando en la rueda dentada de la parte superior derecha, se muestra el nombre de usuario y aparece su sesión como abierta, con posibilidad de cerrarla pinchando en “Cerrar Sesión”. Si pincháramos en “Cerrar Sesión”, la sesión finaliza y volveríamos a la página de inicio donde se pide loguearse:

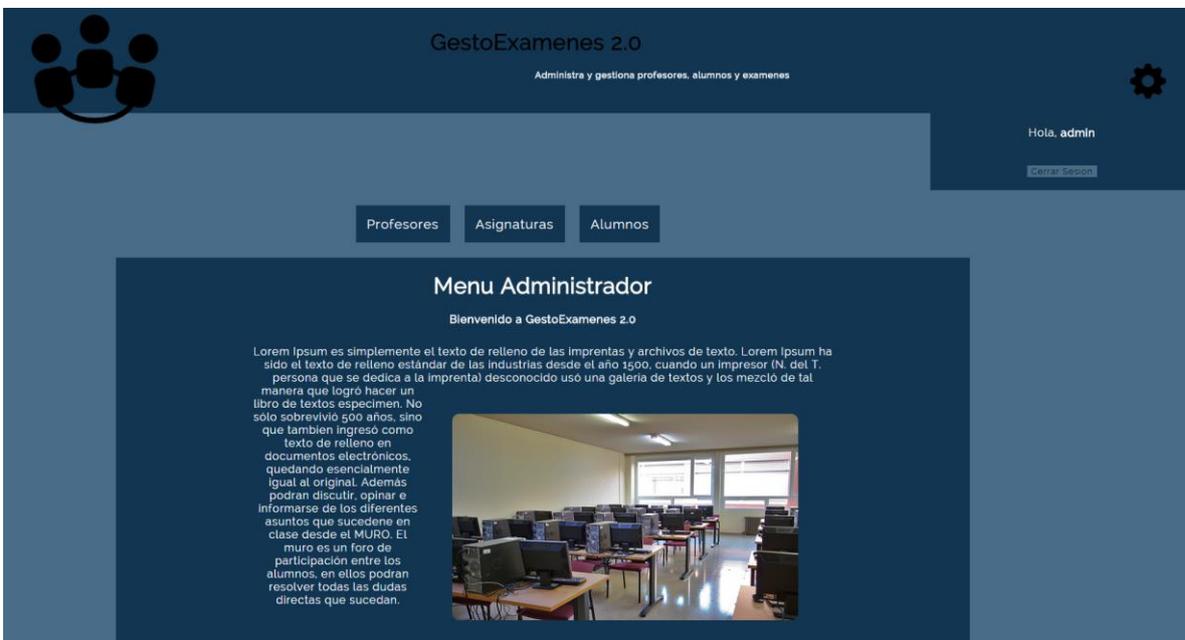


Figura 60: Cuadro Sesión Aplicación Web

Se comprobará ahora al menú “Profesores”. Se muestran los profesores dados de alta en el sistema en el momento de la prueba:

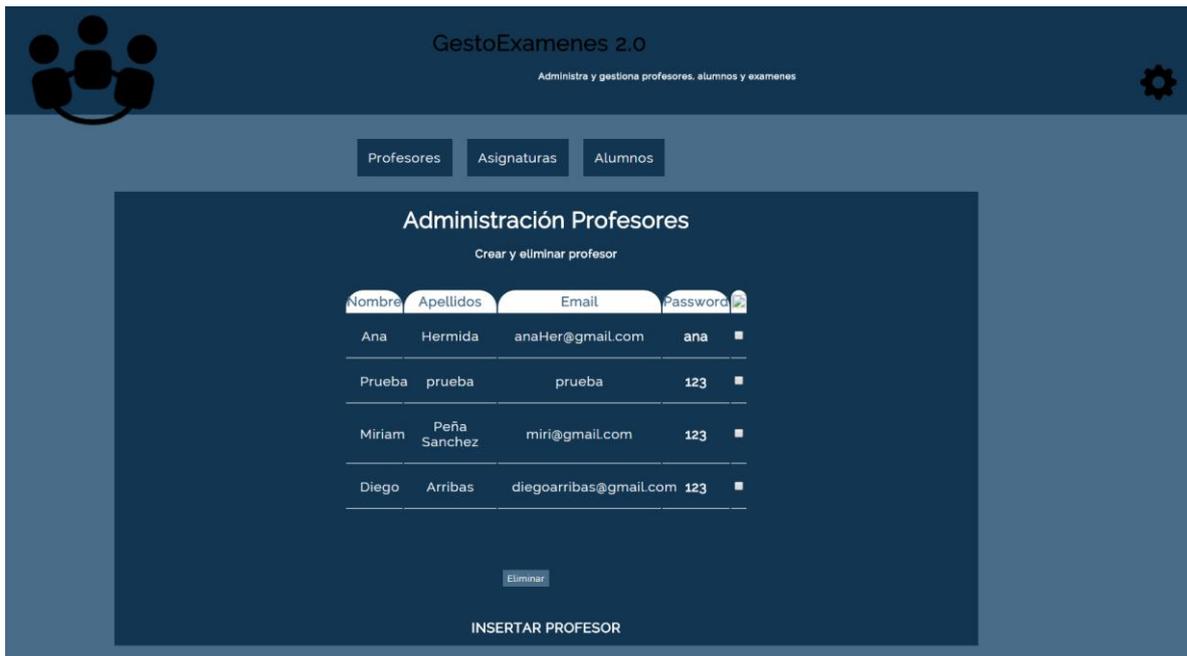


Figura 61: Menú Profesores Módulo Administración Aplicación Web

Vamos a proceder a eliminar un profesor, por ejemplo Miriam Peña Sánchez. Para ello, se selecciona la caja que tiene a su derecha y se pincha en el botón de “Eliminar”.



Figura 62: Eliminar Profesores del Módulo de Administración Aplicación Web

Al pinchar en “Aceptar” en el pop-up que se genera, vemos que se ha eliminado con éxito a Miriam Peña:



Figura 63: Menú Profesores Módulo Administración Aplicación Web

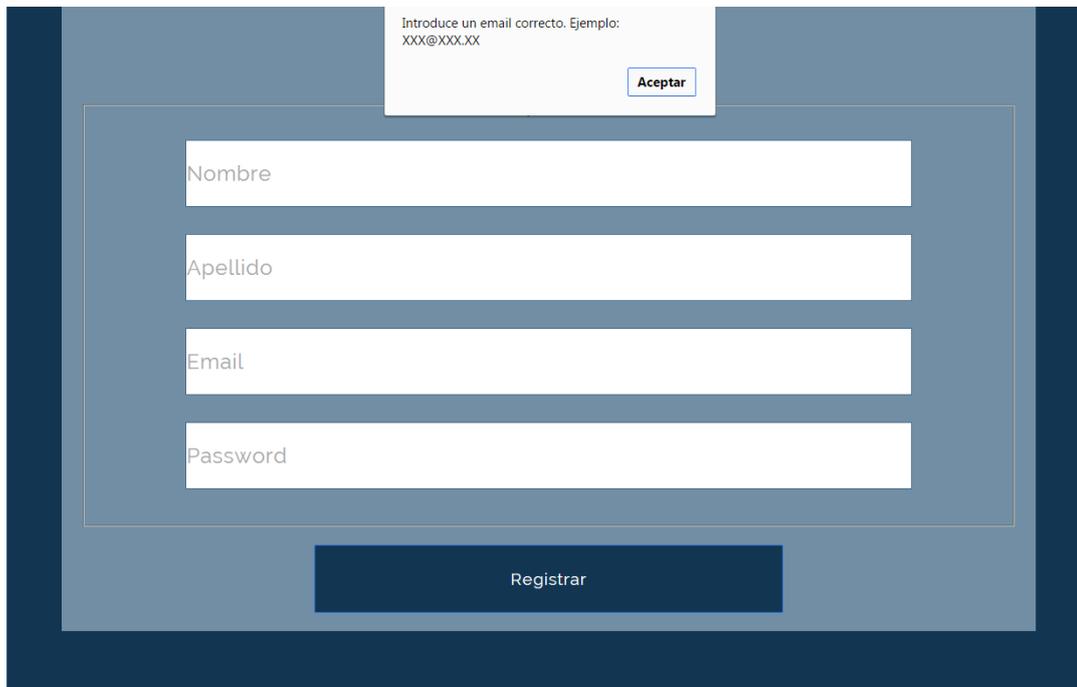
Si intentamos eliminar profesores sin seleccionar ninguno, la página nos advierte de que debemos seleccionar algún profesor para eliminarlo.



Figura 64: Control Errores Eliminar Profesor Aplicación Web

A continuación procederemos a crear un nuevo profesor. Para ello, pinchando en “INSERTAR PROFESOR”, se despliega un formulario para el cual rellenaremos los datos requeridos:

En primer lugar, si pinchamos en “Registrar” sin haber rellenado los datos, la aplicación nos advierte de ello y no permite registrar un profesor.



The image shows a web application interface for adding a professor. It consists of a main form with four input fields: 'Nombre', 'Apellido', 'Email', and 'Password'. Below these fields is a dark blue button labeled 'Registrar'. A modal dialog box is open at the top, displaying an error message: 'Introduce un email correcto. Ejemplo: XXX@XXX.XX' with an 'Aceptar' button.

*Figura 65: Formulario Insertar Profesor Aplicación Web*

De hecho, si no se rellenan todos los campos, no es posible Registrar ningún profesor:

Debes de rellenar todos los campos del formulario correctamente

Aceptar

### REGISTRAR

Datos personales

Nombre

Apellido

eduardo.boemo@uam.es

Password

Registrar

Detailed description: This is a screenshot of a web registration form. At the top, there is a white error message box with the text 'Debes de rellenar todos los campos del formulario correctamente' and an 'Aceptar' button. Below this is a blue header with the word 'REGISTRAR' in white. Underneath is a light blue box containing the form fields. The fields are labeled 'Nombre', 'Apellido', 'eduardo.boemo@uam.es', and 'Password'. At the bottom of the form area is a dark blue 'Registrar' button.

*Figura 66: Control Errores Formulario Insertar Profesor Aplicación Web*

Rellenaremos todos los datos para Registrar correctamente un nuevo profesor en la aplicación (Eduardo Boemo):

¡Profesor registrado correctamente!

Aceptar

Datos personales

Eduardo

Boemo

eduardo.boemo@uam.es

.....

Registrar

Detailed description: This is a screenshot of the same registration form as in Figure 66, but now showing a successful registration. A white message box at the top says '¡Profesor registrado correctamente!' with an 'Aceptar' button. The form fields are filled with the data: 'Eduardo' for Name, 'Boemo' for Surname, 'eduardo.boemo@uam.es' for Email, and '.....' for Password. The 'Registrar' button is still visible at the bottom.

*Figura 67: Registro Profesor Aplicación Web*

Comprobamos que el nuevo Profesor se ha añadido al listado de profesores en el sistema:

GestoExámenes 2.0  
Administra y gestiona profesores, alumnos y exámenes

Profesores Asignaturas Alumnos

### Administración Profesores

Crear y eliminar profesor

Nombre	Apellidos	Email	Password
Ana	Hermida	anaHer@gmail.com	ana
Prueba	prueba	prueba	123
Diego	Arribas	diegoarribas@gmail.com	123
Eduardo	Boemo	eduardo.boemo@uam.es	dslab

Eliminar

INSERTAR PROFESOR

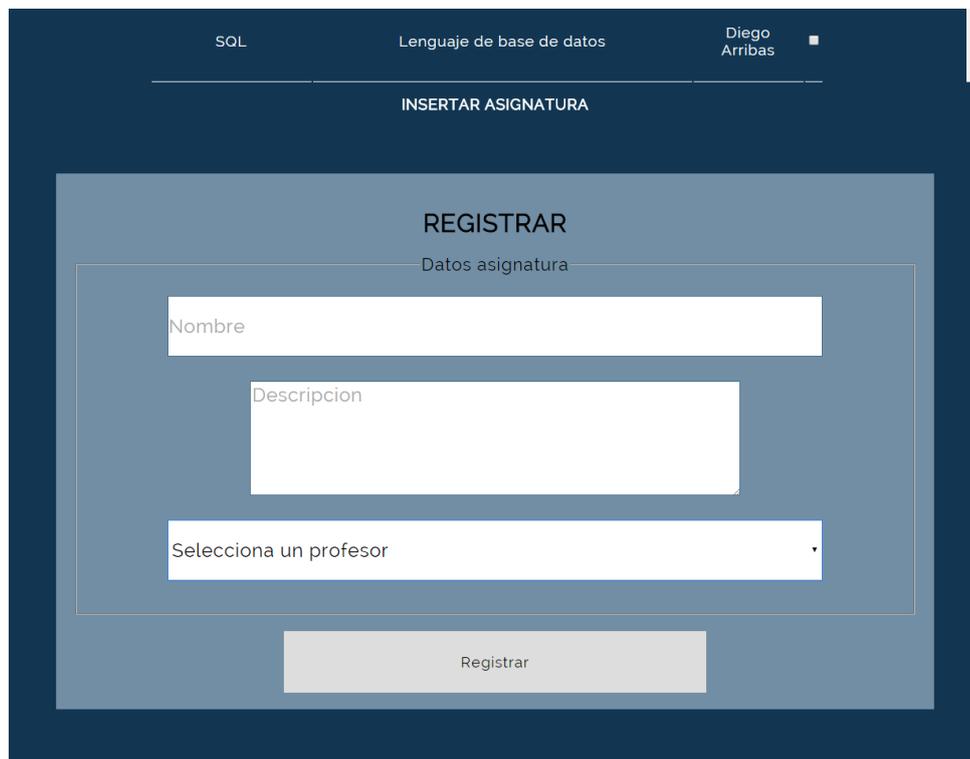
*Figura 68: Nuevo Profesor Aplicación Web*

Pasamos ahora a evaluar el módulo “Asignaturas”. Vemos que cuando hay más de 4 entradas, se genera la barra de scroll:



*Figura 69: Asignaturas Aplicación Web*

El funcionamiento de crear y eliminar asignaturas funciona de forma análoga al del módulo de profesores explicado anteriormente. De todas formas, se muestra el formulario que se despliega al pinchar en “INSERTAR ASIGNATURA”, ya que contiene campos distintos al de profesor:



*Figura 70: Insertar Asignatura Aplicación Web*

En el formulario, se asigna el profesor a la asignatura. Para ello, obviamente, debe haber sido dado de alta previamente:

SQL Lenguaje de base de datos Diego Arribas

INSERTAR ASIGNATURA

REGISTRAR

Datos asignatura

Nombre

Descripcion

Selecciona un profesor

Selecciona un profesor

Hermida, Ana prueba, Prueba

Arribas, Diego

Boemo, Eduardo

*Figura 71: Seleccionar Profesor Nueva Asignatura Aplicación Web*

Al igual que antes, es obligatorio rellenar todos los campos para poder registrar la asignatura en el sistema. Vamos a crear una nueva asignatura:

¡Asignatura registrada correctamente! Aceptar

INSERTAR ASIGNATURA

REGISTRAR

Datos asignatura

DIE

Dispositivos Integrados Especializados

Boemo, Eduardo

Registrar

*Figura 72: Asignatura Registrada Web*

Comprobamos que se ha creado y añadido al listado de asignaturas satisfactoriamente:



Figura 73: Listado Asignaturas Aplicación Web

A continuación, se muestra el menú “Alumnos” perteneciente al Módulo de Administración. El listado de alumnos se carga desde un fichero de texto plano

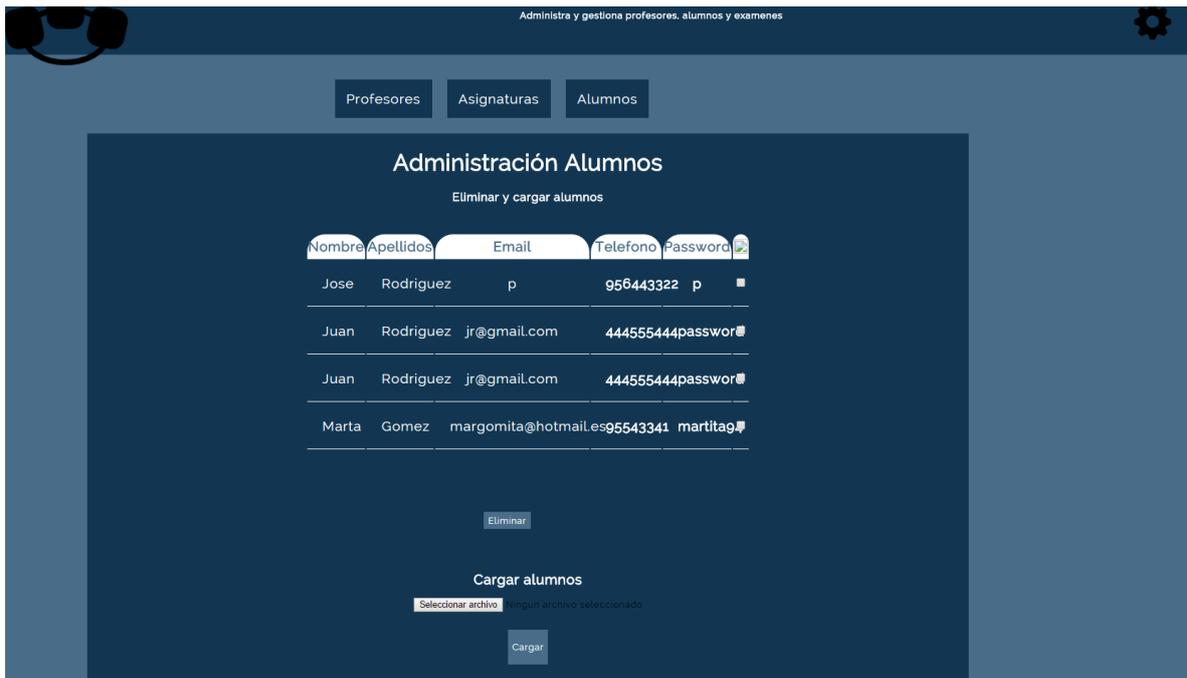


Figura 74: Alumnos Aplicación Web

Ahora, se accederá al Módulo de Profesores. Para ello, usamos la última cuenta que hemos creado antes (usuario: [Eduardo.boemo@uam.es](mailto:Eduardo.boemo@uam.es) / password: dslab). Ya que se trata de acceso como profesor en lugar de como administración, no hay que marcar la caja “Admin”.



Figura 75: Acceso Módulo Profesores Aplicación Web

Al igual que antes, existe control de errores sobre usuario o password incorrecta. No obstante, entramos con user y password correctos. Como se puede observar, se despliegan todas las asignaturas que se asignaron a Eduardo Boemo.

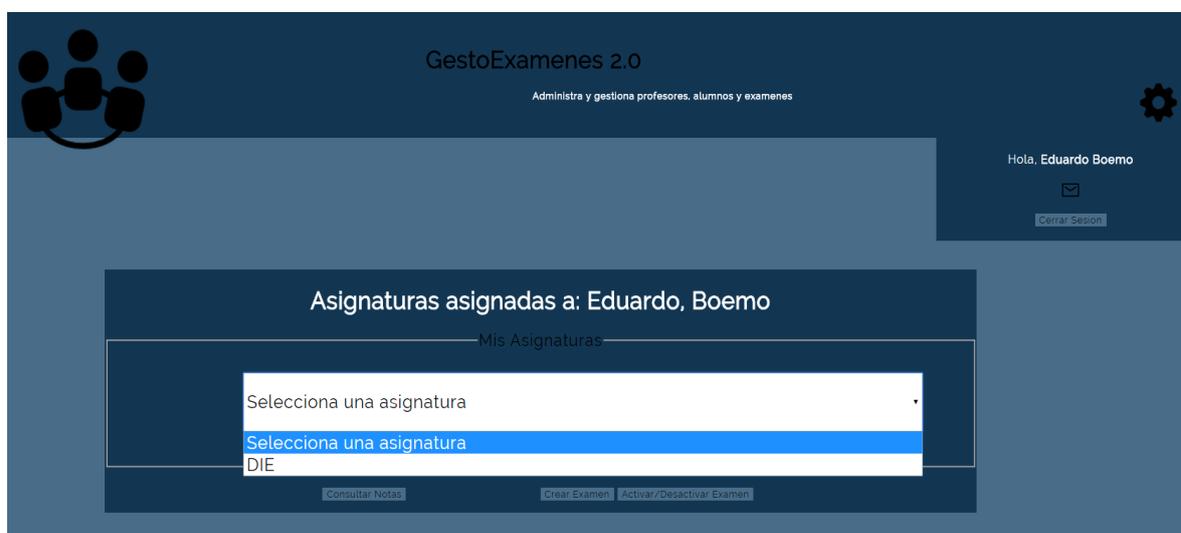
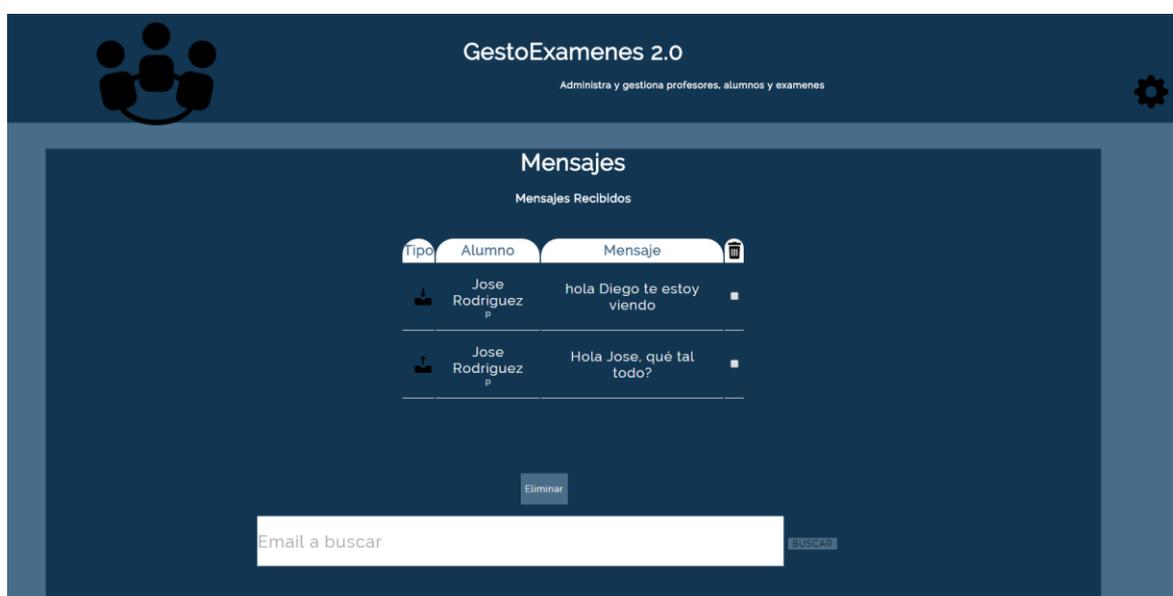


Figura 76: Asignaturas por Profesor Aplicación Web

Una vez implementada e integrada la parte Android de la aplicación que se ha desarrollado en este PFC, es posible testear el funcionamiento de la funcionalidad que permite el envío y recepción de mensajes por parte de los Profesores.

Pinchando en el sobre que aparece arriba a la derecha, accedemos a otra ventana que guarda las comunicaciones entre el profesor y los alumnos. Debido a que las pruebas se hicieron con un profesor ya existente (Diego Arribas) y Eduardo Boemo se acaba de crear, éste último no posee conversaciones con ningún alumno o notas.

Para ello, se accede a la cuenta del profesor [diegoarribas@gmail.com](mailto:diegoarribas@gmail.com) con contraseña “123”. Acto seguido, vamos directamente a consultar los mensajes con los alumnos.



*Figura 77: Mensajes Aplicación Web*

Otra de las opciones disponibles es la de Consultar Notas. Para ello, en primer lugar se selecciona una asignatura y se pincha en el botón “Consultar Notas”. En la nueva ventana, se muestran las calificaciones obtenidas por los alumnos en los distintos exámenes.



*Figura 78: Consultar Notas Aplicación Web*

Ahora, crearemos un examen. Para ello, al pinchar en “Crear Examen” se despliega un formulario:

Crear examen  
Datos del examen

Password

Repite Password

Duracion Examen (min)

dd/mm/aaaa

Selecciona una hora

Descripcion

Aula

Numero de preguntas

*Figura 79: Formulario Crear Examen Aplicación Web*

....

....

100

18/05/2015

12:00

Examen Parcial DIE. Temas 5, 6 y 7.

Aula 1.2

5

Crear Examen

*Figura 80: Formulario Crear Examen Relleno Aplicación Web*

Una vez creado el examen, se pasa a la ventana de relleno de preguntas. Se han elegido 5 para facilitar el ejemplo:

## Preguntas examen

Datos del examen

1) Un semisumador...

Es capaz de sumar el  No dispone de salida  No puede tener en c  Propaga el acarreo p

2) Con un decodificador de 'n' entradas podemos implementar...

Funciones de n varia  Funciones de n+1 var  Varias funciones de r  Funciones de 2^n var

3) Un codificador prioritario de 8 entradas tiene las correspondientes al 0, 2, 5 y 6

000  110  111  Ninguna de las anter

4) Si implementamos una función con decodificador, habitualmente necesitaremos

Una puerta OR  Una puerta AND  Una puerta NOT  Una entrada de habil

5) Disponemos de un circuito codificador de 10 salidas. ¿Cuántas entradas tiene?

Depende del tipo de  3  4  2^10

Crear Examen

*Figura 81: Preguntas y Respuestas Examen Aplicación Web*

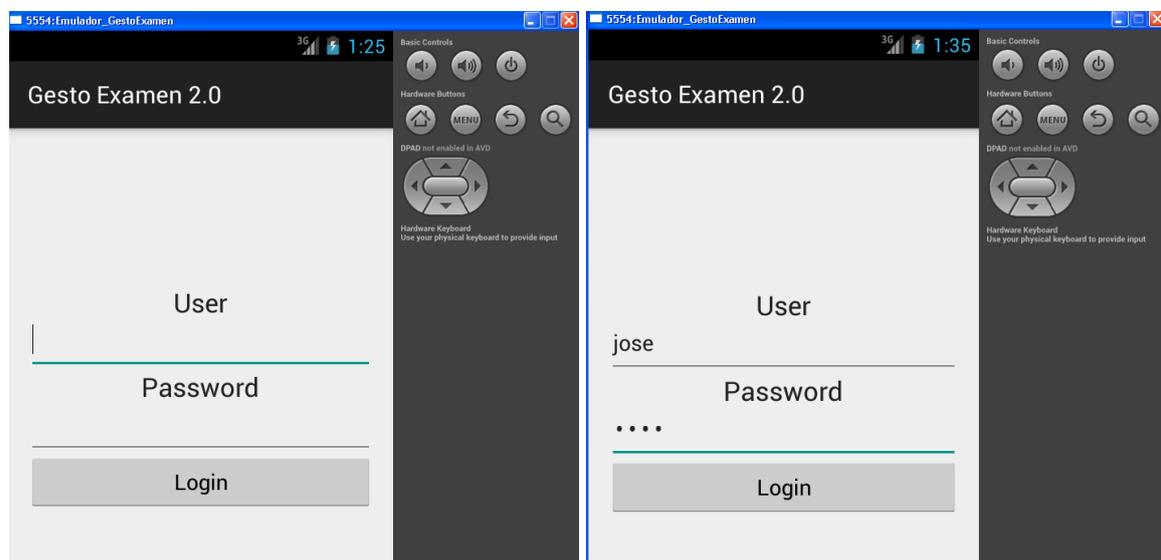
Además de los testeos realizados, se ha comprobado que la parte web de la aplicación es accesible desde cualquier navegador, no solo desde un ordenador, sino desde una Tablet o Smartphone por medio del navegador instalado. Se ha verificado que la funcionalidad se mantiene independientemente del dispositivo que se emplee para acceder.

Por tanto, tras las pruebas realizadas, es posible concluir de forma objetiva que, a nivel de usuario, la aplicación web cumple satisfactoriamente con los requisitos definidos en las especificaciones del PFC.

## 5.2 Aplicación Android

En este apartado se mostrarán pantallazos de la aplicación Android que se ha desarrollado en este proyecto, haciendo las veces de manual de usuario así como de evidencia de pruebas de aceptación de usuario para corroborar que se ha implementado la funcionalidad acordada.

En primer lugar, para acceder a la aplicación, al igual que en la parte web, es necesario introducir nombre de usuario y contraseña. Para poder acceder, es necesario que el alumno se haya dado de alta previamente en la web por parte del administrador. Por supuesto, si se introduce mal la contraseña, no es posible entrar a la aplicación.



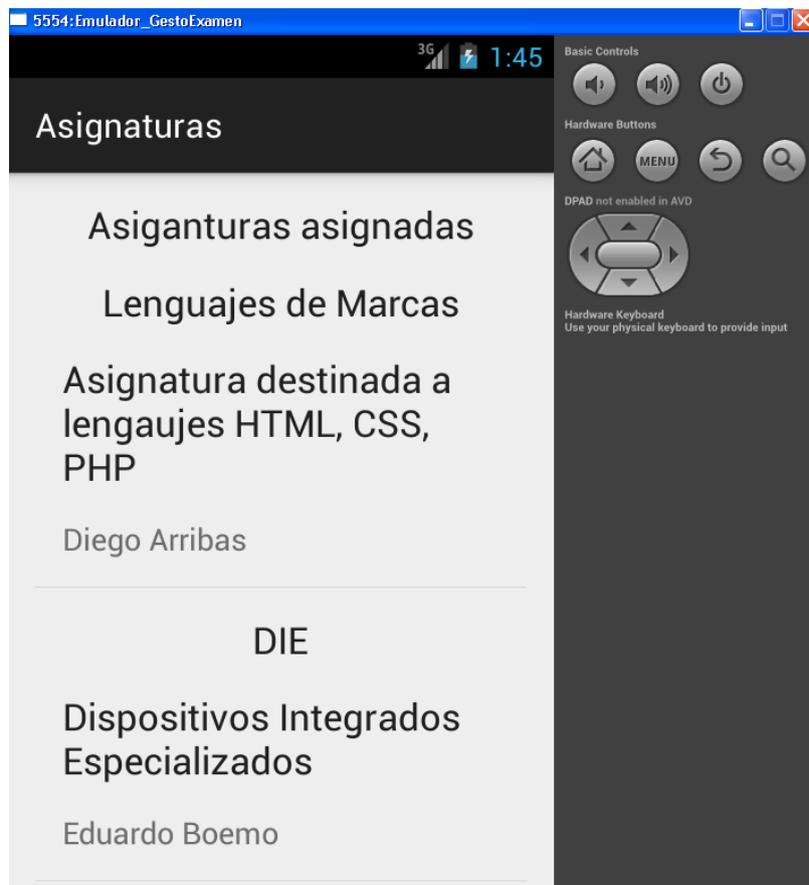
*Figuras 82 y 83: Acceso Aplicación Android*

Tras introducir nombre de usuario y contraseña con éxito, se accede a la página principal de la aplicación, donde existen tres botones que permiten acceder a sus respectivos menús.



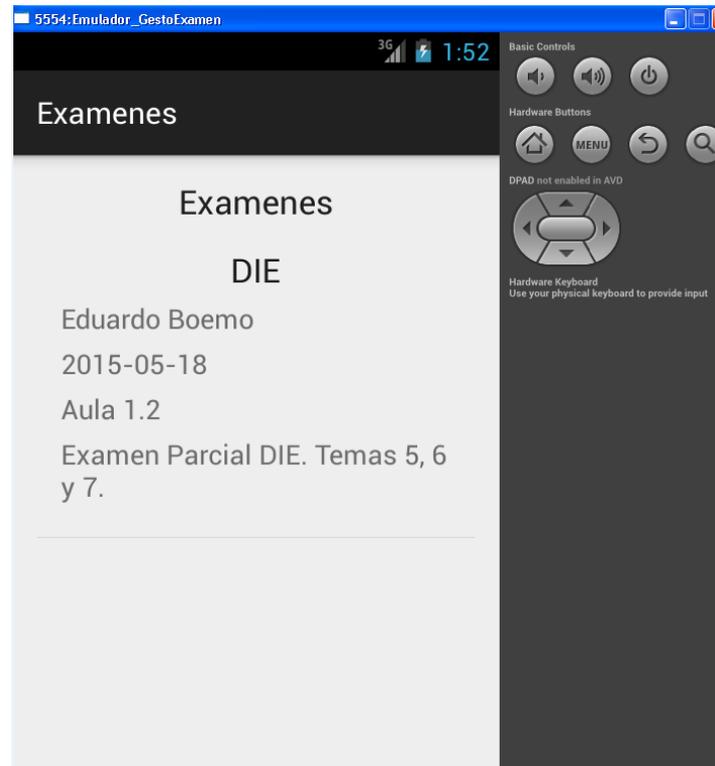
*Figura 84: Menú Principal Aplicación Android*

Pinchando en el botón “Consultar notas” se accede a una pantalla donde se muestran las asignaturas



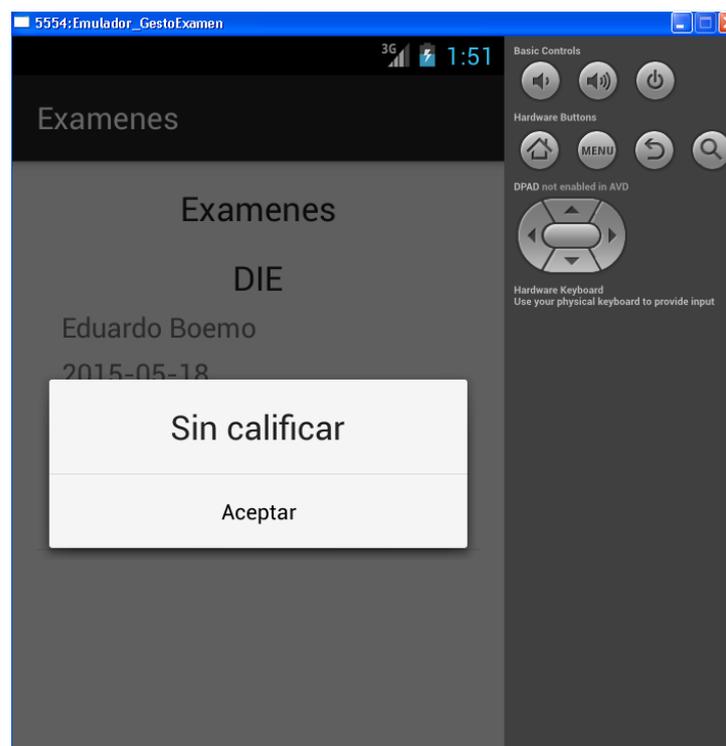
*Figura 85: Asignaturas Matriculado Android*

Accedemos por ejemplo a la asignatura de DIE (Dispositivos Integrados Especializados)



*Figura 86: Exámenes DIE Android*

Al no haber llevado a cabo el alumno dicho examen, no existirá calificación asociada:



*Figura 87: Sin Calificación DIE Android*

Accederemos por tanto a otra asignatura (BBDD), donde el alumno sí ha llevado a cabo exámenes:

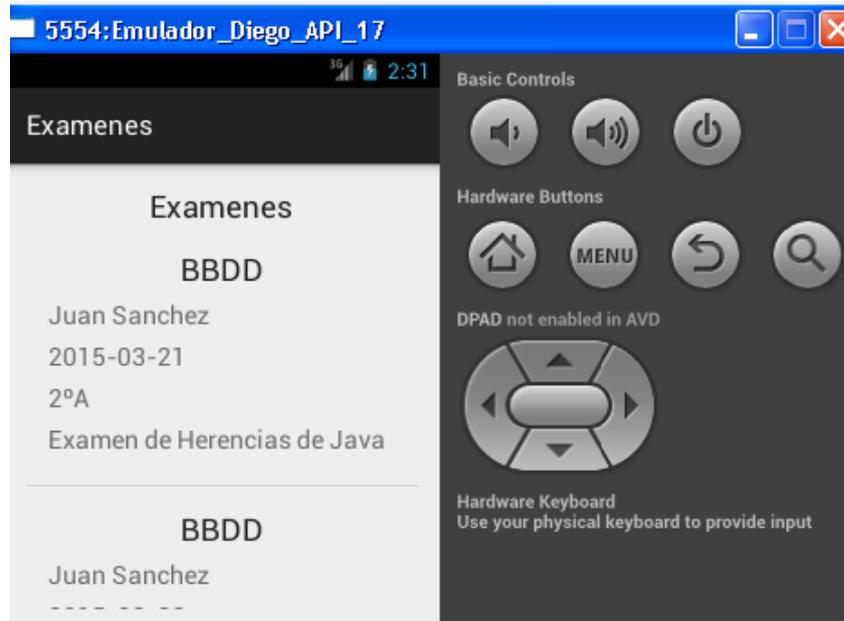


Figura 88: Exámenes BBDD Android

Tras pinchar en el primer examen, se observa un mensaje que muestra la calificación obtenida por el alumno en dicho examen:

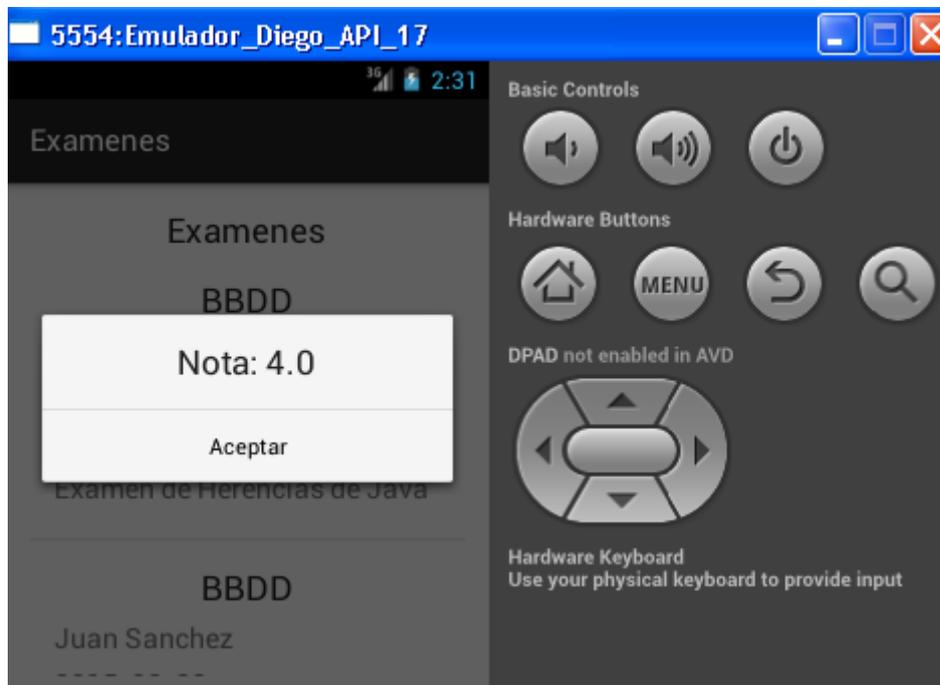
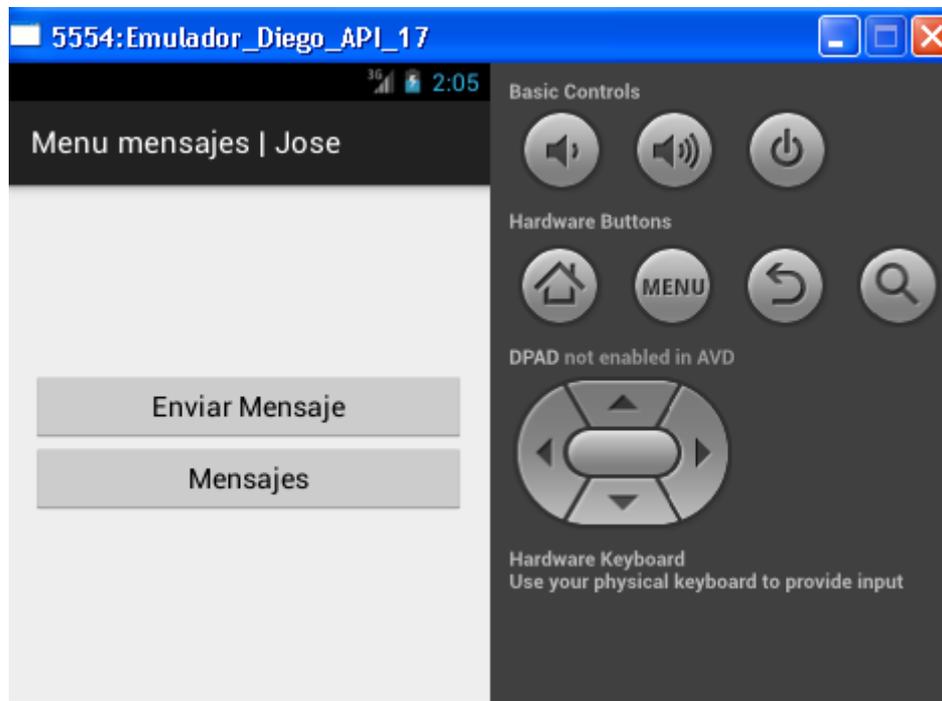


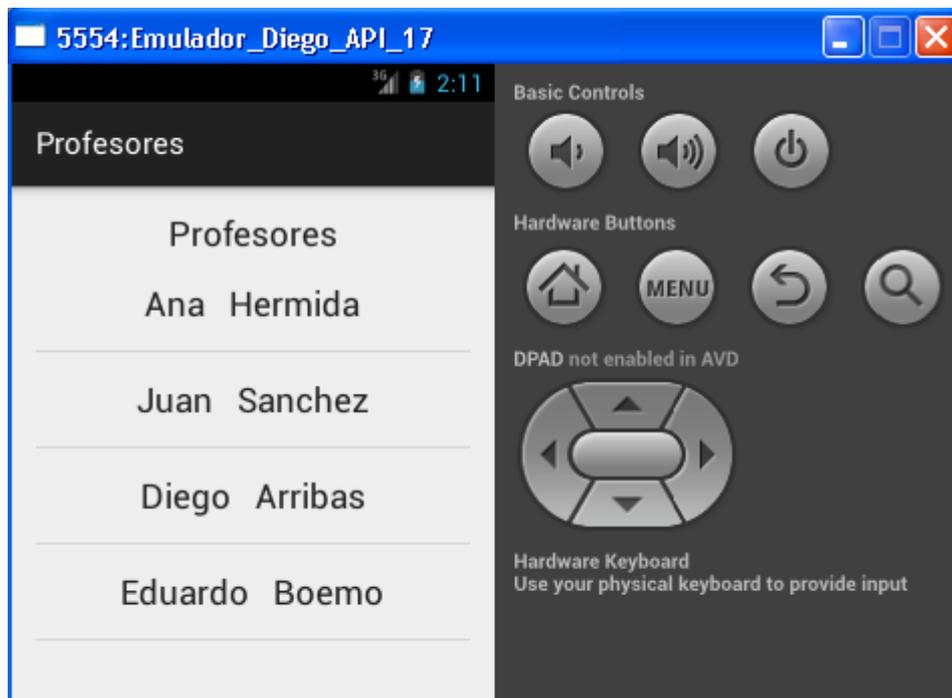
Figura 89: Calificación Examen BBDD Android

Ahora, accederemos al menú “Mensajes” el menú principal de la aplicación. Existirá la posibilidad de acceder a la opción de enviar mensajes o a la de consultar los mensajes ya recibidos y enviados.



*Figura 90: Menú Mensajes Android*

Pinchando en la opción “Enviar Mensaje”, se accede a una pantalla donde se despliega el listado de profesores a los que poder enviar un mensaje.



*Figura 91: Profesores Mensajes Android*

Elegimos a Eduardo Boemo como destinatario de un nuevo mensaje:

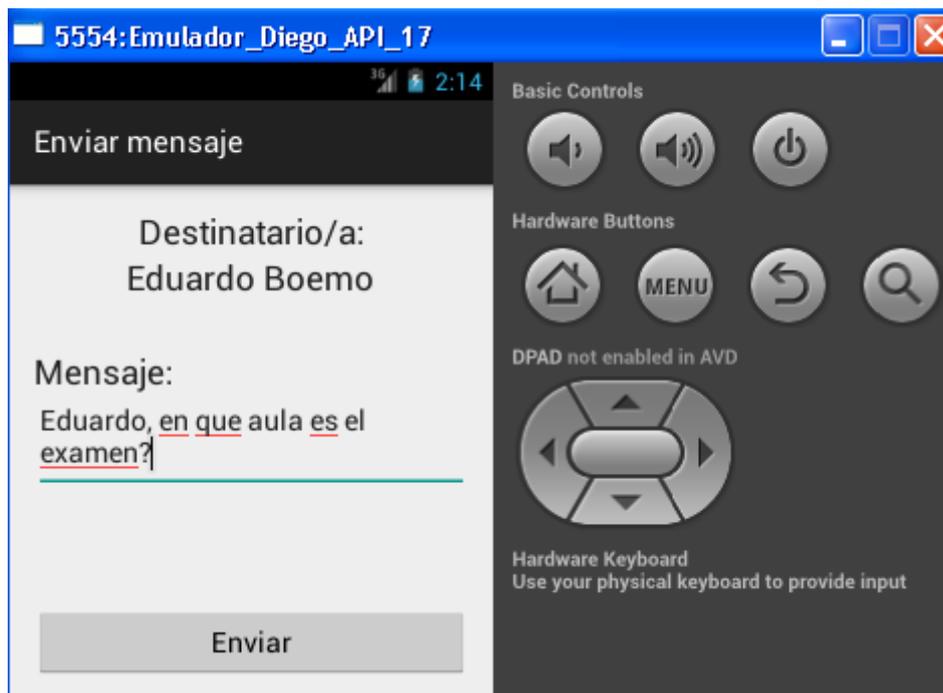


Figura 92: Enviar Mensaje Android

Ahora, observaremos los mensajes enviados y recibidos por el alumno jose. La flecha hacia la izquierda

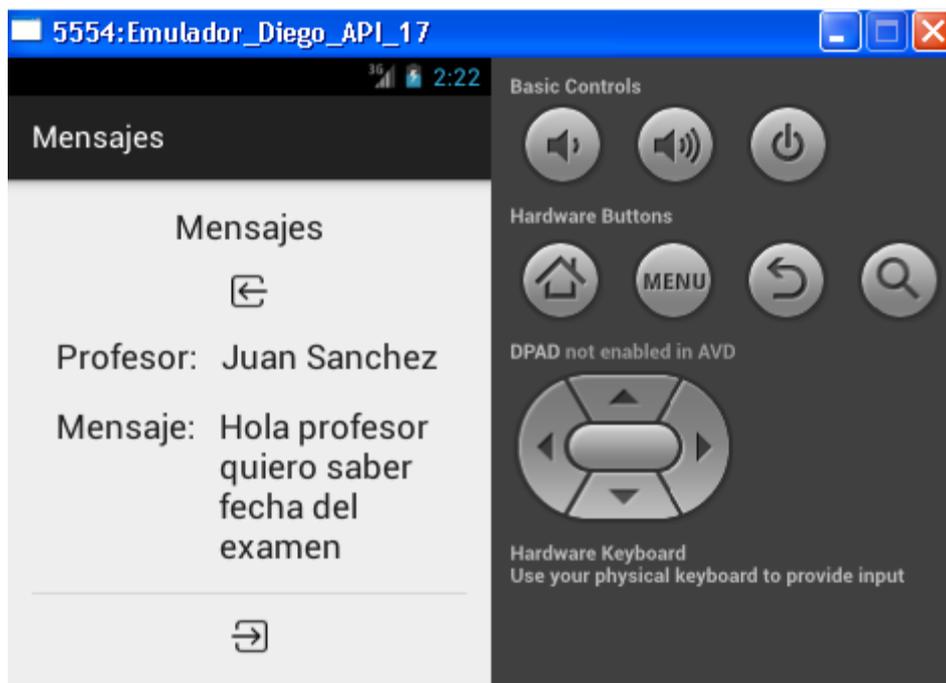


Figura 93: Mensajes Enviados Android

A continuación, vamos a llevar a cabo un examen de respuesta múltiple en nuestro Smartphone Android. Para ello, mediante la sesión ya iniciada de usuario “jose” y password “jose”, seleccionaremos un examen de los ya creados para llevarlo a cabo.

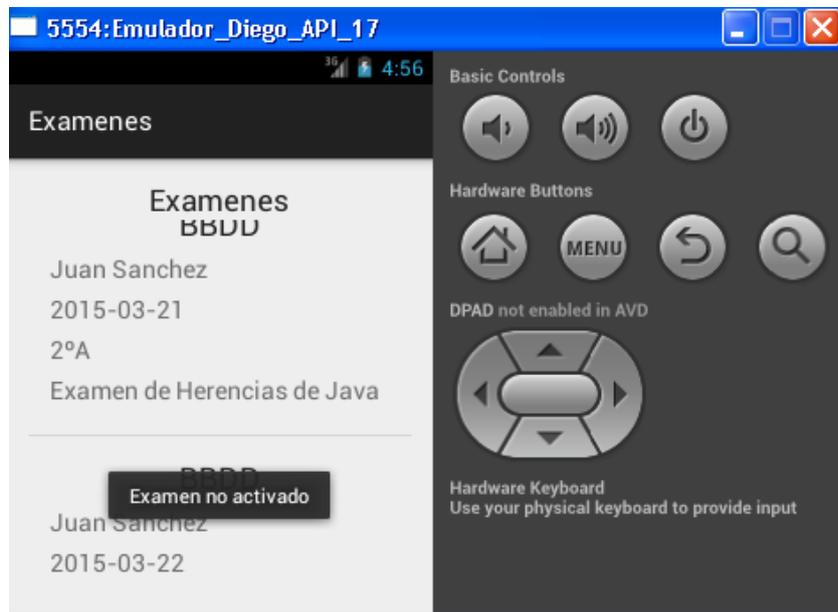


Figura 94: Examen no activado Android

En este caso, el examen no se encuentra activado. Se recuerda que su activación se debe llevar a cabo a través de la aplicación web por medio del profesor que creó el examen. Abriremos el único examen creado para la asignatura de DIE, el cual sí se encuentra activado. Para acceder a él, es preciso introducir correctamente la contraseña de dicho examen, que en este caso es “fpga”. Acto seguido de introducir la contraseña con éxito, se procede a llevar a cabo el examen:



Figura 95: Pregunta Examen Android

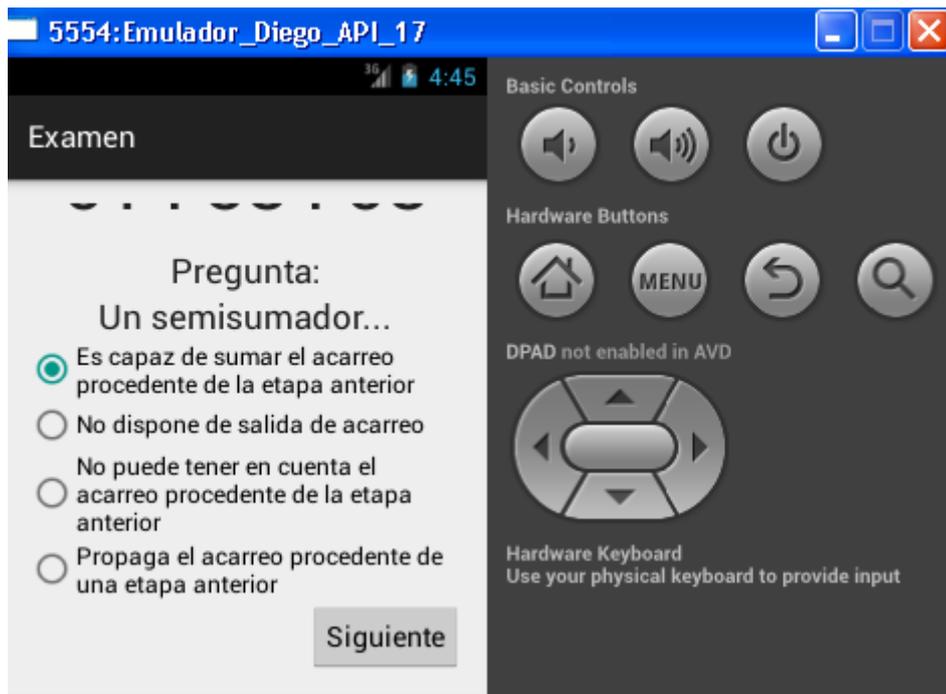


Figura 96: Pregunta Seleccionada Examen Android

Finalmente, tras responder a todas las preguntas del examen, se genera la nota de forma automática.



Figura 97: Calificación Examen Android

# **6 Conclusiones y trabajo futuro**

---

## **6.1 Conclusiones**

A lo largo de las distintas fases del proyecto, se han empleado diferentes herramientas:

- Notepad++
- Navegador Google Chrome
- WampServer
- Filezilla
- Hostinger
- Android Studio y herramientas SDK
- Microsoft Office
- Smartphone BQ Aquaris
- Portátil HP

Asimismo, el alumno ha conseguido ampliar su conocimiento en las siguientes áreas:

- Programación Orientada a Objetos
- Programación Web (lenguajes HTML, CSS, JavaScript, PHP y SQL)
- Bases de Datos
- Programación Android para dispositivos móviles (lenguajes Java y XML)
- Gestión de servidores web
- Gestión de proyectos

En resumen, se han cumplido los requisitos principales definidos al comienzo del proyecto:

- Aprendizaje de lenguajes de programación web y móvil
- Estudio del estado del arte y de aplicaciones similares en el mercado
- Posibilidad de gestionar alumnos y profesores y asignarlos a asignaturas

- Inclusión del listado de alumnos de cada asignatura por medio de un fichero de texto plano
- Creación de exámenes de respuesta múltiple por parte de los profesores: establecimiento del número de preguntas deseado, tiempo requerido para su realización, información asociada, existencia de cuatro respuestas posibles y solo una solución correcta.
- Bloqueo del examen antes y después del tiempo fijado para su realización
- Envío y recibo de Mensajes entre profesores y alumnos
- Consulta de la información de cada alumno y asignatura por parte de Administración y los profesores
- Resolución de exámenes en los smartphones o tablets de los alumnos
- Cálculo de la nota automáticamente al terminar el examen
- Consulta del histórico de notas para cada asignatura
- Coordinación y correcto funcionamiento entre la parte web y la parte Android de la aplicación
- Testeo: casos de pruebas y control de errores.
- Desarrollo de una aplicación totalmente funcional y operativa en smartphones y tablets
- Mantenimiento del estilo de aplicaciones anteriores del DSLab, incluyendo aspectos tales como el tamaño de letra, colores, formato de menú inicial, icono, resolución y relación de aspecto de gráficos, formato de botones, etc.
- Labor documental de la memoria del PFC
- Publicación de la aplicación en Google Play Store

## 6.2 Trabajo futuro

El proyecto se fundamenta en el empleo de la metodología Agile, la cual propone una mejora continua a través de la puesta en producción de distintas releases de la misma aplicación. Algunas mejoras que se contemplan para hipotéticas futuras versiones de la aplicación son las siguientes:

- Habilitar opción de utilizar la aplicación en otros idiomas distintos al castellano.
- Adición de restricciones de complejidad, caducidad y bloqueo de contraseñas de acceso para otorgar mayor seguridad a la autenticación de los usuarios.
- Adaptar el diseño a posibles necesidades del cliente y usuarios, dotándole de un aspecto más amigable para permitir una mayor accesibilidad y facilidad de interacción con los usuarios.
- Inclusión de un apartado de gestión de notas históricas para consultar calificaciones referentes a cursos pasados.
- Posibilidad de añadir imágenes y tablas a las preguntas de los exámenes.
- Inclusión de profesores importados desde un fichero excel o .csv para que no sea necesario introducirlos individualmente (ya implementado para los listados de alumnos).
- Rediseñar el análisis para dar cabida a que los alumnos solo puedan visualizar en su dispositivo móvil los exámenes creados de asignaturas en las que están matriculados.
- Mayor personalización del formato de examen, customizando el número de respuestas y el valor de las mismas. Gracias a otorgar distinta puntuación a las respuestas, se permitiría extrapolar este modelo de exámenes tipo test a la realización de encuestas, generando conclusiones finales en función de la puntuación.
- Establecer que la hora no sea informativa y que el examen se active automáticamente a la hora que se ha programado.
- Dentro del Módulo de Administración, creación de un gestor de gráficas, estadísticas e indicadores que permitan evaluar la utilidad y usabilidad de la aplicación así como el desempeño de los alumnos.
- Adaptar la aplicación a otros sistemas operativos como Apple y Windows Phone.



# Referencias

---

1. Curso “Android: Introducción a la programación” de la UPV.
2. Chris Haseman, “Android Essentials”, firstPress
3. Nicolas Gramlich, “Android Programming”, [andbook.anddev.org](http://andbook.anddev.org)
4. Mednieks, Zigurd, “Programming Android”
5. Lee, Wei-Meng, “Beginning Android application development”
6. <https://www.elandroidelibre.com>
7. <https://developer.android.com/training/index.html>
8. <http://developer.android.com/design/index.html>
9. <http://developer.android.com/develop/index.html>
10. Javier García de Jalón, José Ignacio Rodríguez, Íñigo Mingo, Aitor Imaz, Alfonso Brazález, Alberto Larzabal, Jesús Calleja, Jon García, “Aprende Java como si estuvieras en primero”
11. Jesús Tomás Gironés, “El gran libro de Android”
12. [www.androidcurso.com](http://www.androidcurso.com)
13. Tutorial de Android: <https://www.youtube.com/watch?v=jSSMfRhi7SI>
14. Rogers Cadenhead, “Sams Teach Yourself Java in 21 Days (Covering Java 7 and Android), 6th Edition. Sams.
15. Exportar la app Android desde Android Studio a publicación en Google Play:  
<https://www.youtube.com/watch?v=Qatec8l3XOI>
16. Cursos de PHP, JQUERY, CSS y HTML de <http://www.w3schools.com>.
17. The jQuery Project: <http://docs.jquery.com>
18. Curso de SQL: <http://www.desarrolloweb.com/manuales/9/>
19. Curso de SQL: <http://sql.11sql.com/>



## Glosario

---

API	Application Programming Interface
AVD	Android Virtual Device
BBDD	Bases de Datos
CPU	Central Processing Unit
CSS	Cascading Style Sheets
DSLab	Digital System Laboratory
EPS	Escuela Politécnica Superior
HTML	HyperText Markup Language
OS	Operating System
PHP	Hypertext Pre-processor
PFC	Proyecto Final de Carrera
SDK	Software Development Kit
SQL	Structured Query Language
UAM	Universidad Autónoma de Madrid
XML	eXtensible Markup Language

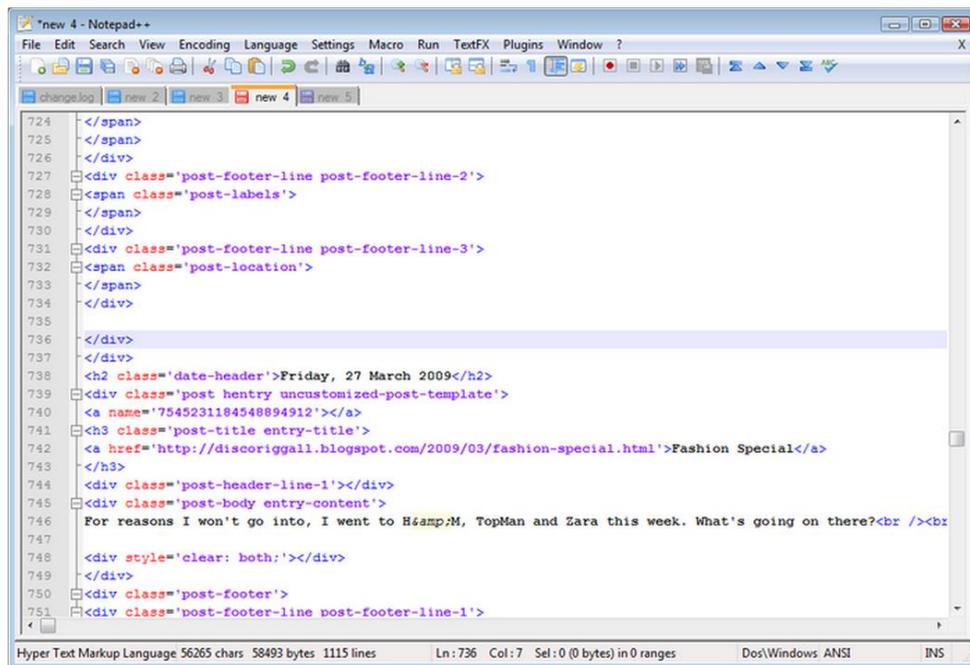
## Anexos

---

### A. Manual de programas

#### Notepad++

En el desarrollo de la aplicación se ha empleado este programa en lugar de un IDE debido a que los lenguajes web no requieren compilación al ser interpretados directamente por los navegadores.



The screenshot shows the Notepad++ application window with a menu bar (File, Edit, Search, View, Encoding, Language, Settings, Macro, Run, TextFX, Plugins, Window, ?) and a toolbar. The main editing area contains HTML code for a blog post. The code includes tags for spans, divs, h2, h3, a, and br. The content of the post is: "For reasons I won't go into, I went to H&M, TopMan and Zara this week. What's going on there?". The status bar at the bottom indicates "Hyper Text Markup Language 56265 chars 58493 bytes 1115 lines Ln: 736 Col: 7 Sel: 0 (0 bytes) in 0 ranges Dos\Windows ANSI INS".

```
724 </span>
725 </span>
726 </div>
727 <div class='post-footer-line post-footer-line-2'>
728 <span class='post-labels'>
729 </span>
730 </div>
731 <div class='post-footer-line post-footer-line-3'>
732 <span class='post-location'>
733 </span>
734 </div>
735
736 </div>
737 </div>
738 <h2 class='date-header'>Friday, 27 March 2009</h2>
739 <div class='post hentry uncustomized-post-template'>
740 <a name='7545231184548894912'></a>
741 <h3 class='post-title entry-title'>
742 <a href='http://discoriggall.blogspot.com/2009/03/fashion-special.html'>Fashion Special</a>
743 </h3>
744 <div class='post-header-line-1'></div>
745 <div class='post-body entry-content'>
746 For reasons I won't go into, I went to H&M, TopMan and Zara this week. What's going on there?<br /><br>
747
748 <div style='clear: both;'></div>
749 </div>
750 <div class='post-footer'>
751 <div class='post-footer-line post-footer-line-1'>
```

Figura 98: Ejemplo de Notepad++

Notepad++ es un editor de texto y de código fuente libre con soporte para varios lenguajes de programación. Entre la variedad de lenguajes, soporta los lenguajes de programación web utilizados para el desarrollo de la parte web de la aplicación.

#### WampServer

WAMP es el acrónimo usado por una infraestructura de Internet que emplea las siguientes herramientas:

- Windows como Sistema Operativo.
- Apache como servidor web.
- MySQL como gestor de BBDD.
- PHP como lenguaje de programación.

El uso de un WAMP permite servir página HTML a Internet, además de permitir la gestión de datos de las mismas.

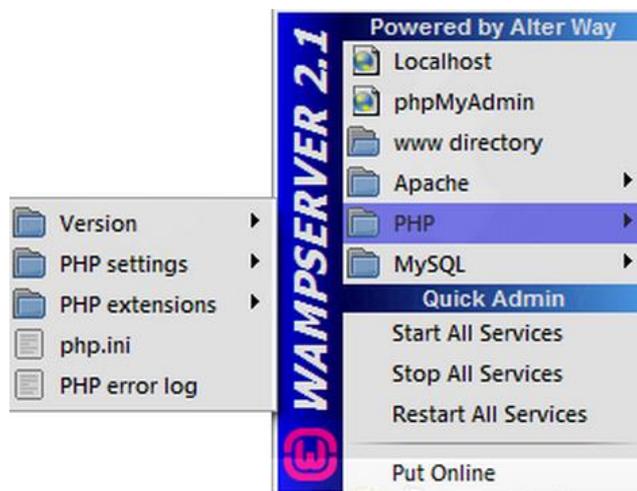


Figura 99: Menú de WampServer

WampServer es el programa que se ha utilizado para la gestión en local de la aplicación web durante su desarrollo previo a la subida a un servidor web. Este programa cuenta con el módulo PhpMyAdmin, que permite gestionar fácilmente las BBDD.

Como curiosidad, el servidor Apache de WampServer y Skype operan bajo el mismo puerto, el 80, es decir, el HTTP. Por ello, no es posible hacer uso de ambos programas al mismo tiempo salvo que se configuren puertos distintos.

¿Por qué utilizar WampServer en lugar de otro gestor de bases de datos? Su interfaz es muy sencilla e intuitiva, está disponible en varios idiomas, es gratuito, su lenguaje de programación es PHP (que suele ser el mismo lenguaje que usan las páginas web a la hora de insertar consultas internas dentro de la misma página, por ejemplo, a la hora de añadir un buscador a nuestra web), funciona de forma local y con servidores externos, se actualiza automáticamente, no tiene límite en cuanto a las bases de datos (es decir, podemos tener muchos proyectos con infinidad de bases de datos cada uno de ellos) y cada base de datos la podemos crear con los campos y el tipo de caracteres que necesitemos. Además, se pueden crear usuarios, exportar e importar archivos.

## Filezilla

Se trata de un cliente FTP multiplataforma de código abierto. Este programa se ha utilizado durante el desarrollo del proyecto para el traspaso de la información desde el servidor local hacia el servidor web de Hostinger.

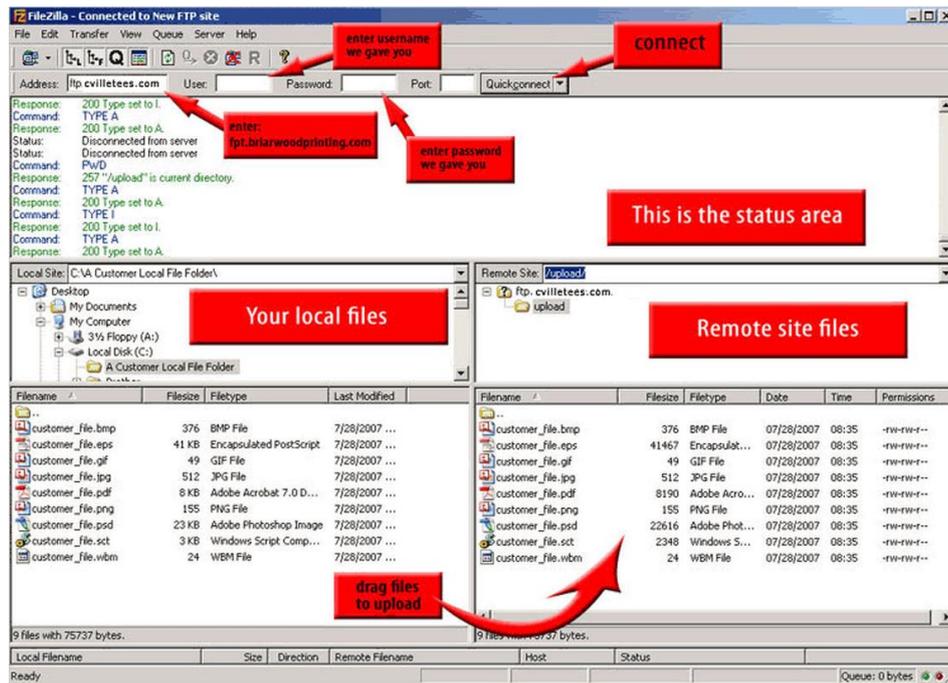


Figura 100: Ejemplo de Filezilla

## Hostinger

Servidor gratuito que provee alojamiento web y dominio sin costes con soporte para PHP y MySQL. Se ha elegido este sitio web como podía haberse elegido otros sitios de alojamiento gratis o de pago.

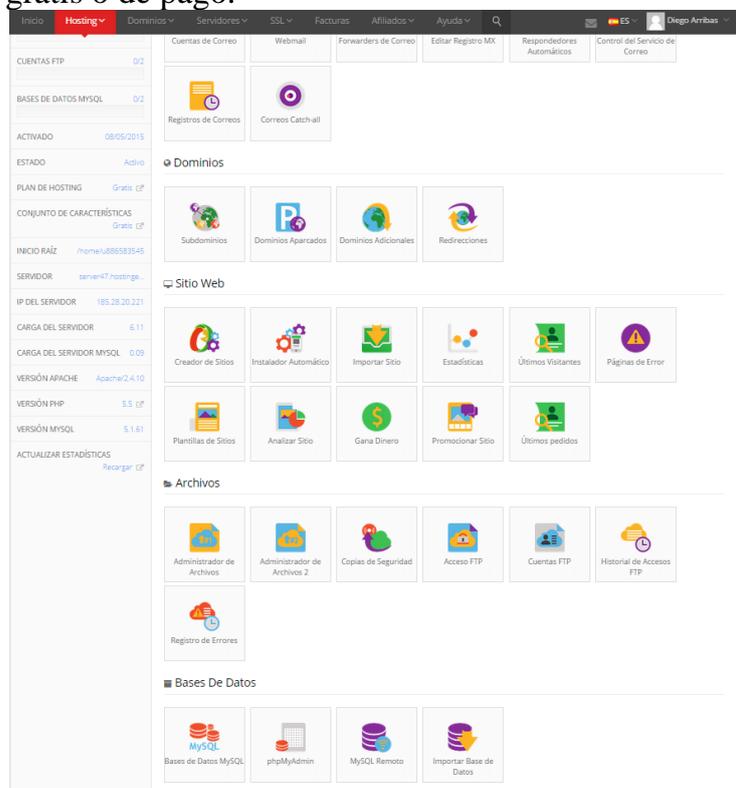


Figura 101: Menú Cuenta Servidor Web Hostinger

## Android Studio

Se trata de un Entorno de Desarrollo Integrado (IDE) para la plataforma Android. Es el programa que se ha empujado para el desarrollo de la parte enfocada a los dispositivos móviles de los alumnos. Se trata de un IDE relativamente nuevo que, debido a su funcionalidad y enfoque, ha sustituido rápidamente a Eclipse como entorno de desarrollo para programación en Android. A la izquierda, muestra la jerarquía de proyectos, en el centro un espacio de trabajo para la codificación y a la derecha una previsualización de la aplicación en el dispositivo escogido.

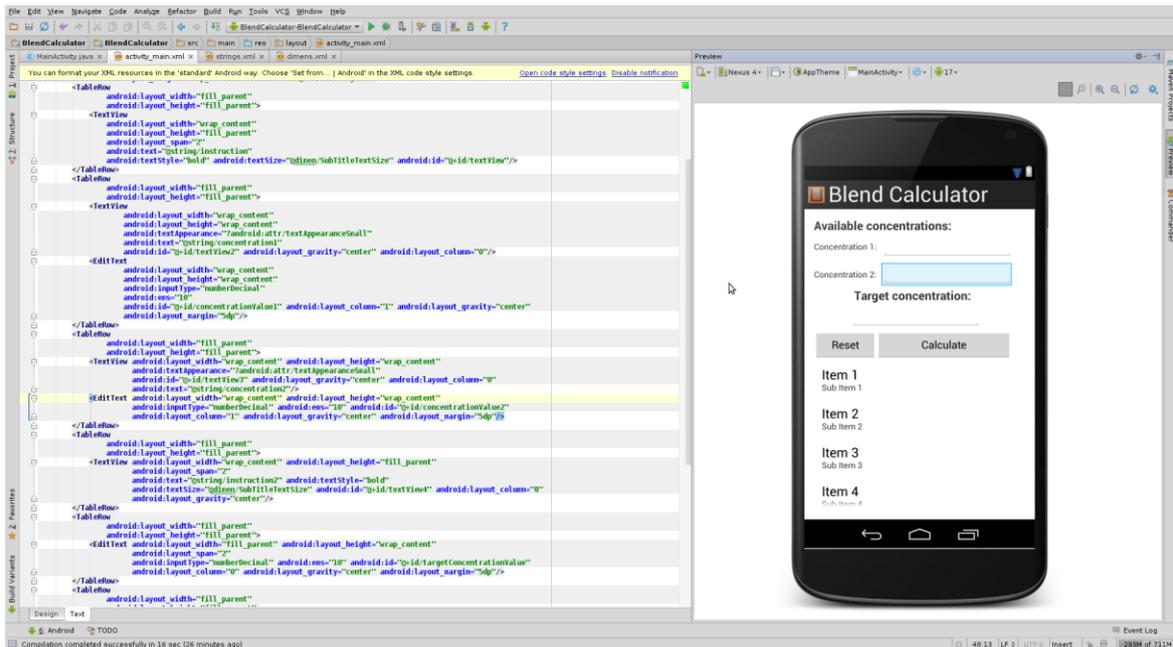


Figura 102: Ejemplo de Android Studio

## **B. Presupuesto**

### **1. Ejecución Material**

1.	Compra de ordenador personal (Software incluido).....	600 €
2.	Smartphone y Tableta Android .....	600 €
3.	Material de oficina .....	10 €
4.	Total de ejecución material .....	1.210 €

5.

### **6. Gastos generales**

1.	16 % sobre Ejecución Material .....	193,6 €
----	-------------------------------------	---------

2.

### **3. Beneficio Industrial**

4.	6 % sobre Ejecución Material .....	72,6 €
----	------------------------------------	--------

5.

### **6. Honorarios Proyecto**

7.	540 horas a 15 € / hora .....	8.100 €
----	-------------------------------	---------

8.

### **9. Material fungible**

10.	Gastos de impresión .....	40 €
-----	---------------------------	------

11.	Encuadernación .....	10 €
-----	----------------------	------

12.

### **13. Subtotal del presupuesto**

1.	Subtotal Presupuesto .....	9.626,2 €
----	----------------------------	-----------

2.

### **3. I.V.A. aplicable**

4.	21% Subtotal Presupuesto.....	2.021,5 €
----	-------------------------------	-----------

5.

### **6. Total presupuesto**

7.	Total Presupuesto .....	11.647,7 €
----	-------------------------	------------

Madrid, Enero de 2016

El Ingeniero Jefe de Proyecto

Fdo.: Diego Miguel Arribas Marcos  
Ingeniero de Telecomunicación



## **C. Pliego de condiciones**

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un GESTOR DE EXÁMENES Y CALIFICACIONES EN ANDROID. En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

### **Condiciones generales**

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es

obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

### **Condiciones particulares**

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.
2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.
3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.
4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.
5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.
6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.