



---

# Universidad de Valladolid

E.U. de Informática (Segovia)

Grado en Ingeniería Informática de Servicios y  
Aplicaciones

---

App CV-UVA

La aplicación del Campus Virtual de la UVA,  
optimizada para dispositivos móviles Android.

---

Alumno: Mónica Fernández Mata

Tutor: Fernando Díaz Gómez

## **AGRADECIMIENTOS**

En primer lugar quiero dar las gracias a mi familia, especialmente a mis padres por haberme dado la oportunidad de seguir estudiando. Por su apoyo y su comprensión.

A mis amigos que han estado a mi lado y me han animado a conseguirlo, sobre todo a Rodri, que me apoyó en los momentos más difíciles.

También quería agradecer muchísimo a mi tutor Fernando por su paciencia y ya que sin él este proyecto no habría sido posible.

## **BLOQUE I**

---

### **MEMORIA DEL TFG**

## **BLOQUE II**

---

### **DOCUMENTACIÓN TÉCNICA**

## **BLOQUE III**

---

### **DOCUMENTACIÓN DEL USUARIO**

## **BLOQUE IV**

---

### **CONCLUSIONES Y POSIBLES AMPLIACIONES**

# ÍNDICE DEL DOCUMENTO

Agradecimientos _____	111
-----------------------	-----

## BLOQUE I

1. Introducción _____	5
1.1 Identificación del Trabajo Fin de Grado _____	5
1.2 Organización de la documentación _____	5
1.3 Estructura del CD _____	7
2. Descripción general del Trabajo Fin de Grado _____	9
2.1 Motivación _____	11
2.2 Objetivos _____	11
2.3 Cuestiones metodológicas _____	12
2.4 Tecnología de desarrollo _____	16
3. Descripción general del producto _____	17
3.1 Funcionalidad del producto _____	23
3.2 Arquitectura del producto _____	24
3.2.1 Arquitectura general _____	24
3.2.2 Arquitectura específica _____	26
3.3 Despliegue del producto _____	31
4. Planificación y presupuesto _____	33
4.1 Estimación de Trabajos _____	33
4.1.1 Estimación mediante puntos de función (PF) _____	33
4.1.1.1 Explicación del proceso _____	33
4.1.1.2 Estimación por PF _____	39
4.1.2 Estimación de costes por COCOMO _____	42
4.1.2.1 Explicación del algoritmo COCOMO _____	42
4.1.2.2 Aplicación de COCOMO _____	43
4.2 Planificación _____	45
4.3 Presupuesto _____	47
5. Cuestiones de diseño reseñables _____	49
6. Cuestiones de implementación reseñables _____	55
6.1 Ciclo de vida de las activities _____	56
6.1.1 Guardar estado de las activities _____	58
6.2 Utilización de “listviews” _____	59
6.3 Notificaciones _____	62

## BLOQUE II

7.	Análisis del sistema _____	63
7.1	Objetivos del sistema _____	64
7.2	Catálogo de requisitos del sistema _____	65
7.2.1	Requisitos de información _____	65
7.2.2	Requisitos no funcionales _____	68
7.2.3	Requisitos funcionales _____	69
7.2.3.1	Identificación de actores _____	69
7.2.3.2	Casos de uso _____	70
7.3	Modelo de objetos _____	78
7.3.1	Diagrama de cases de análisis _____	78
7.3.2	Catálogo de cases de análisis _____	78
7.4	Modelo dinámico _____	82
7.4.1	Lista de escenarios _____	82
7.4.2	Diagramas de secuencia _____	85
7.4.3	Diagramas de estado del sistema _____	93
8.	Diseño del sistema _____	97
8.1	Diseño de la base de datos _____	98
8.1.1	Esquema conceptual _____	98
8.1.2	Esquema relacional _____	100
8.1.3	Esquema interno _____	101
9.	Diseño de las pruebas del sistema _____	107
9.1	Pruebas de instalación _____	107
9.2	Pruebas del sistema _____	109

## BLOQUE III

10.	Introducción _____	115
11.	Instalación del sistema _____	117
11.1	Instalación del Servidor Apache _____	117
11.2	Instalación de PHP5 _____	120
11.2.1	Configuración del archivo .ini _____	120
11.2.2	Configuración del archivo httpd.conf de Apache _____	122
11.2.3	Probando nuestro servidor Apache _____	125
11.3	Instalación de MySQL _____	127
11.4	Instalación de PHP MyAdmin _____	135
11.5	Instalación de CV_UVA _____	136
12.	Manual de usuario CV_UVA _____	139
12.1	¿Qué es CV_UVA? _____	139
12.2	Requisitos mínimos _____	139
12.3	Utilización CV_UVA (Web) _____	140
12.4	Utilización CV_UVA (Android) _____	146



## BLOQUE IV

13. Evaluación	151
13.1 Evaluación de rendimientos	151
13.2 Evaluación de robustez	151
13.3 Evaluación de seguridad	151
14. Conclusiones	153
15. Consecución de objetivos	155
15.1 Adquisición y aplicación de conocimientos	155
15.2 Posibles ampliaciones	155
16. Bibliografía	157
16.1 Libros	157
16.2 Páginas Web	157
17. Anexos	159
17.1 Glosario de Términos	159
17.2 Licencia del proyecto	161



# 1. INTRODUCCIÓN

## 1.1 IDENTIFICACIÓN DEL TRABAJO FIN DE GRADO

**Título:** App CV-UVA: La aplicación del Campus Virtual de la UVA, optimizada para dispositivos móviles Android.

**Autor:** Mónica Fernández Mata

**Tutor:** Fernando Díez Gómez

**Departamento:** Informática

**Área:** Ciencias de la Computación e Inteligencia Artificial

## 1.2 ORGANIZACIÓN DE LA DOCUMENTACIÓN

La documentación de CV-UVA se va a organizar basándose en la estructura recomendada por la Escuela Universitaria de Informática de Segovia. En concreto, la documentación se divide en tres bloques o secciones independientes: la memoria del proyecto (Bloque I), la documentación técnica (Bloque II) y la documentación de usuario (Bloque III). Esta organización se justifica en que cada sección tiene una audiencia bien diferenciada en cada caso (el público en general, programadores y desarrolladores, y usuarios de la aplicación, respectivamente) y representan un documento autocontenido (con entidad propia), por lo que aparecen como subdocumentos independientes, cada uno con su propio índice, y separados convenientemente dentro del volumen encuadernado.

La Bloque I, denominada "Memoria del TFG" contiene, además de la introducción e identificación del TFG:

- La descripción general del trabajo fin de grado (TFG) en la que se tratan los objetivos perseguidos por el mismo, así como una serie de cuestiones metodológicas y tecnológicas relevantes.
- La descripción general del producto software desarrollado, describiéndose las funcionalidades soportadas, una descripción de la arquitectura adoptada y de su despliegue en el entorno de explotación final.
- La planificación y presupuesto del trabajo en base a la estimación de la carga de trabajo siguiendo el método COCOMO.
- Las cuestiones de diseño reseñables en el desarrollo del trabajo y que condicionaron la elaboración de la documentación técnica.
- Las cuestiones de implementación reseñables en el desarrollo del trabajo con el fin de lograr un producto final de calidad.

La Bloque II, denominada "Documentación Técnica", pretende aportar los detalles suficientes a desarrolladores y programadores para una profunda comprensión de cómo se ha diseñado e implementado la aplicación, partiendo de un análisis lo más riguroso posible. Esta sección se ha estructurado, en grandes líneas, en los siguientes apartados:

- Análisis del Sistema.

En este apartado se incluye información detallada de los objetivos perseguidos por el sistema, así como los requisitos de información, los casos de uso, actores del sistema y los requisitos no funcionales.

- Diseño del Sistema.

En el apartado de diseño se ha incluido el diseño de la base de datos, el diseño de CV-UVA con la descripción de sus tipos de objetos y asociaciones, el modelo de comportamiento y el modelo de despliegue de CV-UVA.

- Diseño de la interfaz de usuario

En este apartado se ha incluido el diseño de la interfaz de usuario, que incluye los bosquejos de las pantallas y un diseño de alto nivel de la interfaz entre otras cosas.

- Implementación del sistema.

En este apartado se listan algunos fragmentos del código fuente que, a mi juicio, son bastante representativos respecto al desarrollo general.

- Pruebas del sistema.

En este último apartado se incluye tanto el tipo, como la batería de pruebas realizadas.

La Bloque III, denominada "Documentación de usuario", contiene el manual de uso de la aplicación. Esta sección tiene como fin formar al usuario de la aplicación en su uso de forma sencilla y gráfica.

La Bloque IV, nos muestra las conclusiones obtenidas después de la realización de Proyecto fin de Carrera de forma detallada.

















































- Se hace una evaluación de la aplicación teniendo en cuenta los puntos importantes de rendimiento, robustez y seguridad.

- Una valoración entre los objetivos iniciales del problema y la consecución de los mismos.

- Mención a los nuevos conocimientos adquiridos gracias a la realización de esta aplicación.

- Se enumeran posibles ampliaciones del producto, tanto de mejora de la aplicación como la ampliación de la misma.
- La bibliografía y las páginas web utilizadas para la realización de este PFC.

### 1.3 ESTRUCTURA DEL CD

Nombre	Fecha de modifica...	Tipo	Tamaño
 Código	27/08/2013 16:42	Carpeta de archivos	
 AppAndroid	27/08/2013 16:42	Carpeta de archivos	
<ul style="list-style-type: none"> <li> .settings</li> <li> assets</li> <li> bin</li> <li> gen</li> <li> libs</li> <li> res</li> <li> src</li> <li> .classpath</li> <li> .project</li> <li> AndroidManifest</li> <li> ic_launcher-web</li> <li> proguard-project</li> <li> project.properties</li> </ul>	<ul style="list-style-type: none"> <li>09/07/2013 22:53</li> <li>09/07/2013 22:53</li> <li>26/08/2013 15:22</li> <li>27/08/2013 16:42</li> <li>09/07/2013 22:53</li> <li>09/07/2013 22:53</li> <li>09/07/2013 22:55</li> <li>09/07/2013 22:53</li> <li>09/07/2013 22:53</li> <li>21/08/2013 18:10</li> <li>09/07/2013 22:53</li> <li>09/07/2013 22:53</li> <li>09/07/2013 22:53</li> </ul>	<ul style="list-style-type: none"> <li>Carpeta de archivos</li> <li>Carpeta de archivos</li> <li>Carpeta de archivos</li> <li>Carpeta de archivos</li> <li>Carpeta de archivos</li> <li>Carpeta de archivos</li> <li>Archivo CLASSPA...</li> <li>Archivo PROJECT</li> <li>Archivo XML</li> <li>Archivo PNG</li> <li>Documento de tex...</li> <li>Archivo PROPERTI...</li> </ul>	<ul style="list-style-type: none"> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li></li> <li>1 KB</li> <li>1 KB</li> <li>2 KB</li> <li>13 KB</li> <li>1 KB</li> <li>1 KB</li> </ul>
 AppWeb	27/08/2013 16:43	Carpeta de archivos	
<ul style="list-style-type: none"> <li> album_tracks</li> <li> albums</li> <li> data</li> <li> databueno</li> <li> datacopiasseg</li> <li> datarod</li> <li> notificacion</li> <li> track</li> <li> fotos</li> <li> acces</li> <li> borrarpdf</li> <li> conection</li> <li> config</li> <li> connectbd</li> <li> contenido</li> <li> contenidocopiasseg</li> <li> contenidopordefecto</li> <li> estilo</li> <li> funciones_bd</li> <li> grabadorarchivos</li> <li> guardarpdf</li> <li> identificador2</li> <li> inicio</li> <li> pdf</li> <li> personal</li> <li> servicio</li> <li> verpdf</li> </ul>	<ul style="list-style-type: none"> <li>22/07/2013 13:00</li> <li>23/07/2013 0:22</li> <li>30/07/2013 22:40</li> <li>18/07/2013 22:44</li> <li>18/07/2013 18:54</li> <li>19/07/2013 1:27</li> <li>21/08/2013 18:46</li> <li>30/07/2013 17:19</li> <li>22/07/2013 21:27</li> <li>21/07/2013 21:58</li> <li>23/07/2013 19:34</li> <li>07/03/2013 22:01</li> <li>03/07/2013 0:29</li> <li>02/07/2013 23:30</li> <li>30/07/2013 20:59</li> <li>25/04/2013 22:52</li> <li>10/06/2013 18:49</li> <li>10/06/2013 21:13</li> <li>18/07/2013 22:31</li> <li>11/06/2013 18:11</li> <li>30/07/2013 20:55</li> <li>07/03/2013 22:00</li> <li>07/03/2013 12:03</li> <li>18/04/2013 17:54</li> <li>18/07/2013 22:22</li> <li>02/07/2013 20:45</li> <li>30/07/2013 20:59</li> </ul>	<ul style="list-style-type: none"> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>Carpeta de archivos</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>Documento de ho...</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>Archivo HTML</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> <li>PHP Script</li> </ul>	<ul style="list-style-type: none"> <li>1 KB</li> <li>1 KB</li> <li>2 KB</li> <li>2 KB</li> <li>12 KB</li> <li>2 KB</li> <li>2 KB</li> <li>1 KB</li> <li></li> <li>1 KB</li> <li>1 KB</li> <li>1 KB</li> <li>1 KB</li> <li>1 KB</li> <li>2 KB</li> <li>2 KB</li> <li>1 KB</li> <li>3 KB</li> <li>3 KB</li> <li>2 KB</li> <li>1 KB</li> <li>2 KB</li> <li>2 KB</li> <li>3 KB</li> <li>1 KB</li> <li>1 KB</li> </ul>
 Memoria	28/08/2013 22:05	Carpeta de archivos	
<ul style="list-style-type: none"> <li> CV-UVAmemoriaBLOQUE1</li> <li> CV-UVAmemoriaBLOQUE2</li> <li> CV-UVAmemoriaBLOQUE3</li> <li> CV-UVAmemoriaBLOQUE4</li> </ul>	<ul style="list-style-type: none"> <li>28/08/2013 22:04</li> <li>28/08/2013 22:04</li> <li>28/08/2013 22:04</li> <li>28/08/2013 22:05</li> </ul>	<ul style="list-style-type: none"> <li>Adobe Acrobat D...</li> <li>Adobe Acrobat D...</li> <li>Adobe Acrobat D...</li> <li>Adobe Acrobat D...</li> </ul>	<ul style="list-style-type: none"> <li>1.380 KB</li> <li>505 KB</li> <li>1.195 KB</li> <li>140 KB</li> </ul>



## 2. DESCRIPCIÓN GENERAL DEL TFG

Con la aparición de la nueva generación de teléfonos inteligente o *smartphones* proporcionada por la evolución tecnológica del hardware, GOOGLE presenta a finales de 2008 el primer dispositivo móvil con sistema operativo Android. A partir de su lanzamiento, el número de dispositivos vendidos con ese sistema operativo crece exponencialmente.

Android se encuentra en continua evolución, respondiendo a las necesidades de los usuarios. Tal vez, algunos de los motivos de esa evolución exitosa del sistema sean su carácter de código abierto, respaldado por una amplia alianza de fabricantes de hardware y software, por supuesto, también por el apoyo de una gran comunidad de desarrolladores capaces de crear un extenso número de aplicaciones, debido a la homogeneidad y simpleza de las herramientas de programación.

El presente trabajo fin de grado (a partir de ahora lo denominaremos TFG) está dedicado a desarrollar una aplicación que permita visualizar en un dispositivo móvil (cliente) basado en Android, archivos PDF que se encuentren en un servidor. Así mismo, recibir notificaciones en el dispositivo cada vez que se añadan y/o modifiquen archivos en el servidor.

Este desarrollo está enfocado para usuarios que quieran consultar en cualquier lugar y a cualquier hora del día archivos que estén en un servidor. Por ejemplo, el caso que nos ocupa en este TFG, alumnos de la Universidad de Valladolid que quieran consultar los apuntes de sus asignaturas o cualquier otro archivo que el profesor haya subido al Campus Virtual. Aunque de igual forma puede ser empleado por trabajadores de una empresa que necesiten consultar archivos...

Para poder consultar estos archivos PDF, previamente han tenido que ser almacenados en el servidor por el profesor. Para ello, se ha desarrollado también una aplicación web, utilizando APACHE+PHP+MYSQL.

Esta última parte, cobra menos protagonismo, puesto que la instancia principal del TFG es la creación de la aplicación Android para dispositivos móviles. Aunque nadie duda de su necesidad, puesto que es el motor que hace poblar nuestra BBDD de archivos para poder ser visitados.

Al realizar el análisis de los requisitos se obtuvo el modelo de negocio al que se debe ajustar este desarrollo. Este modelo se analiza y explica mediante un conjunto de gráficos que se presentan a continuación.

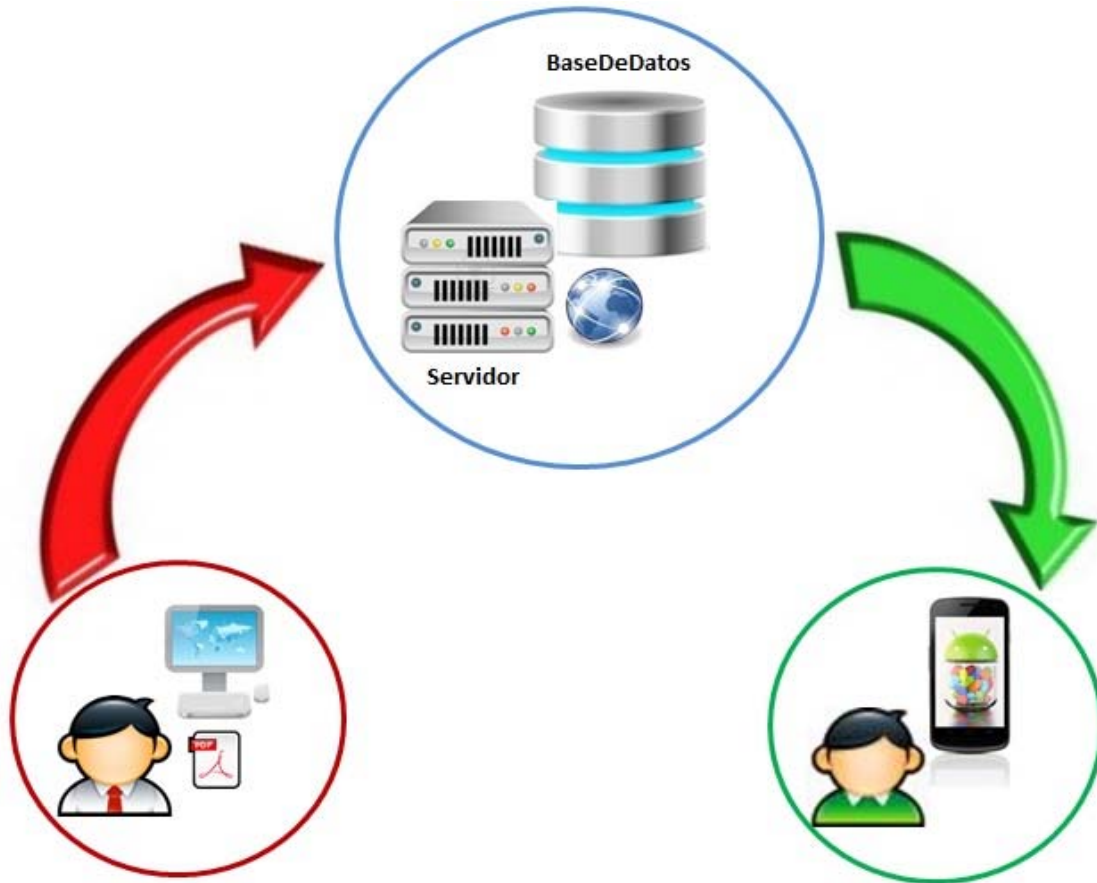


Ilustración 1. Visión general del TFG

Es importante aclarar en este punto (después de la Ilustración 1) y para que se tenga en cuenta a lo largo de toda la documentación, que este TFG se trata de dos aplicaciones totalmente diferentes, en tecnología, arquitectura etc... pero totalmente sincronizada la una con la otra.

Se ha intentado describir de la misma manera tanto la App Web como a la App Android, sobre todo en la memoria técnica, por eso, se va a hacer una duplicidad de memoria.

Para que no quepa lugar a duda, se han separado siempre los diagramas o características de cada una de las aplicaciones con el siguiente encabezado:

**APLICACIÓN WEB**

**APLICACIÓN ANDROID**

Indicando con este encabezado, que aplicación estoy describiendo.



## 2.1 MOTIVACIÓN

La idea de realizar este TFG surge, cuando después de los exámenes del primer cuatrimestre, tenía la necesidad visitar el Campus Virtual desde mi dispositivo móvil, para ver si habían salido las notas.

El Campus Virtual no está optimizado para dichos dispositivos, por lo que cada vez que lo visitaba tenía que estar ampliando la imagen para iniciar sesión, luego disminuir para seleccionar la asignatura...

Es entonces, cuando me surge la idea de crear una aplicación para estas situaciones, y luego, pensar en que de esta forma, también tendría acceso a todos los apuntes de una forma más eficaz que visitando el campus virtual desde el navegador de mi dispositivo móvil.




Si juntamos esta situación con mis ganas de aprender a desarrollar aplicaciones móviles, nace CV-UVA.

## 2.2 OBJETIVOS

El objetivo principal de este TFG, es el desarrollo de la aplicación para dispositivos móviles basado en el sistema operativo Android, del Campus Virtual de la Universidad de Valladolid (Segovia).

Consecuencia de este objetivo tan general, se van a detallar, objetivos más concretos. Los cuales vamos a perseguir a lo largo de todo el proceso de desarrollo, para poder llegar a su consecución.

### Aplicación Web

-  El sistema deberá gestionar la información relativa a los usuarios (profesor y alumno). El alumno, no podrá tener acceso a esta parte de la aplicación, puesto que él no será el que tenga que subir archivos.
-  El sistema deberá gestionar la información relativa a los datos del profesor, mostrando las asignaturas que éste impartirá.
-  El sistema deberá gestionar la información relativa a los archivos PDF, clasificándolos en función de la asignatura a la que pertenezcan. A su vez, y tratándose de los archivos PDF, el sistema deberá permitir subirlos, mostrarlos y borrarlos.

## ● Aplicación Android

- ▶ El sistema deberá gestionar la información relativa a los usuarios (profesor y alumno). En este caso, podrá tener acceso a la App cualquier persona que esté dada de alta en el sistema, no existe restricciones como en la Aplicación Web.
- ▶ El sistema deberá gestión la información relativa a los datos de los usuarios, mostrando las asignaturas que imparta (profesor) o en las que esté matriculado (alumno).
- ▶ El sistema deberá gestionar la información relativa a los archivos PDF, clasificándolos en función de la asignatura a la que pertenezcan. En este caso, la aplicación solo deberá mostrar los archivos PDF, no se podrán subir ni borrarlos desde la aplicación Android.
- ▶ El sistema deberá ser capaz de notificar al usuario de la aplicación cuando ha habido cambios en la base de datos, esto es, cuando se hayan añadido o borrado archivos en la base de datos.

## ● En común

- ▶ La perfecta integración entre ambas aplicaciones, es decir, que los cambios en la App Web, se vean reflejados de forma instantánea en la App Android.
- ▶ Se persigue diseñar de una interfaz sencilla y atractiva.

## 2.3 CUESTIONES METODOLÓGICAS

El objetivo del desarrollador de este trabajo es cumplir estrictamente con los plazos de entrega, manteniendo el alcance del proyecto establecido en una primera fase de análisis. Para ello, es imprescindible utilizar una metodología de trabajo eficaz, adaptada a las características del proyecto, cuyas claves son:

### ▶ **Máxima dedicación**

El desarrollador del proyecto aplicará la máxima dedicación posible durante la duración del mismo.

### ▶ **Mecanismos de comunicación ágiles**

La constante y rápida comunicación entre el tutor del trabajo y el desarrollador garantizará la resolución d los posibles problemas que se presenten durante el desarrollo del proyecto y, en consecuencia, el éxito del proyecto.

Es importante aclarar en este punto, que la parte web no se trata de un sistema orientado a objetos, aunque sí que se ha considerado así para la realización de todo el tema de análisis. Ni que decir tiene, que la parte de Android sí que es un sistema orientado a objetos, por lo que debe seguirse una metodología que tenga en cuenta las características de este tipo de sistemas.

En concreto, para este proyecto se ha seguido una metodología propia, basada en el Proceso Unificado Rational (RUP, *Rational Unified Process*) y está enfocada a la interacción continua con el usuario, ayudando a la captura de requisitos. Es la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos.

Las razones para elegir RUP como base son:

- Su gran éxito en la construcción de sistemas orientados a objetos.
- El uso de buenas prácticas como un ciclo de vida iterativo y un desarrollo dirigido por casos de uso.
- El uso de un lenguaje estándar, como es UML (*Unified Modeling Language*), para la definición y diseño del sistema.

Las características primordiales del Proceso Unificado son:

- Se trata de un proceso iterativo e incremental ya que nos permite tener ‘mini-proyectos’ antes de terminar todo el software. Con esto hemos podido presentar a nuestro cliente un avance del proyecto y él ha ido evaluando nuestro trabajo. En cada iteración, se realizan cambios en el diseño y se agregan nuevas funcionalidades y capacidades al sistema. De esta manera se ha conseguido reducir los riesgos del proyecto.
- Está centrado en la arquitectura ya que se basa en diseñar un subsistema ejecutable tempranamente. RUP propone un modelado gráfico de la estructura y el comportamiento de la arquitectura y los componentes (UML es la base de modelado gráfico de RUP). Esta arquitectura debe ser: flexible, fácil de modificar, intuitivamente comprensible y debe promover la reutilización de componentes.
- Está guiado por casos de uso, los casos de uso guían el desarrollo de la arquitectura y la arquitectura se realimenta en los casos de uso: los dos juntos permiten conceptualizar, gestionar, y desarrollar adecuadamente el software.

- Permite afrontar los riesgos inherentes al proceso de desarrollo, dado que los riesgos del proyecto son entendidos y detectados de forma temprana, y luego son revisados frecuentemente. Esto constituye un beneficio directo para la administración del proyecto en la reevaluación del alcance de los requisitos o en la asunción de otros cambios tácticos.

Dentro de los ciclos de vidas iterativos, se eligió el modelo incremental. Los motivos más importantes para realizar esta elección son:

- Estar basado en el ciclo de vida en cascada y en la construcción de prototipos.
- Priorización de requisitos. Los más críticos se incluyen en los primeros incrementos.
- Los primeros incrementos sirven como prototipos para la detección de nuevos requisitos.
- Riesgo bajo de fallar en el proyecto total.

En cuanto a los artefactos entregables del análisis y diseño del sistema, presentes en la documentación técnica del proyecto, se han seguido las plantillas recomendadas en la asignatura de Ingeniería del Software, intentando ser lo más preciso posible.

En el apartado de diseño se han añadido los diagramas de secuencia y estados que se ha creído más oportunos para representar el comportamiento del sistema. Además, se ha incluido un nuevo apartado llamado “Diseño de interfaz de usuario” en el que se describe todo el proceso de diseño de la interfaz de usuario, lo que incluye la representación de los objetos del dominio del problema, los bosquejos de las pantallas y un diseño de alto nivel de todas las ventanas y diálogos existentes.

Para finalizar con la metodología empleada, se pasa a describir las fases que se han realizado durante el desarrollo.

Cada una de estas fases se realiza por duplicado, puesto que ese TFG consta de dos aplicaciones, la Web y la Android, como ya hemos dicho a lo largo de la memoria.

### **1. Fase de estudio previo:**

En esta fase se estudia toda la temática relacionada con el problema a resolver lo que incluye un estudio básico de la plataforma ANDROID/WAMP.

### **2. Iteración 1:**

Se trata de la primera iteración, en ella, se realizará el análisis de requisitos y del modelo de datos.

### **3. Iteración 2:**

Esta iteración está destinada al diseño de la interfaz de usuario y su implementación.

### **4. 4 Iteración 3:**

El objetivo en esta iteración es la implementación de la ventana de conexión y su integración con el resto de módulos además de realizar las pruebas oportunas.

### **5. Iteración 4:**

Esta iteración es destinada a la realización de las pruebas de todo el sistema.

### **6. Documentación:**

La labor de documentación se realiza de forma paralela al desarrollo del proyecto hasta su finalización.

Cada iteración puede constar de las siguientes etapas (no tienen por qué ser todas):

1. Análisis, que comprende:
  - 1.1. Identificación de los actores participantes.
  - 1.2. Identificar requisitos (requisitos de información, casos de uso, etc.).
  - 1.3. Definir detalladamente los requisitos del sistema.
2. Diseño.
  - 2.1. Diseño de la estructura modular del sistema.
  - 2.2. Diseño de interfaces entre componentes del sistema.
  - 2.3. Diseño de la interfaz del usuario.
  - 2.4. Definición detallada de componentes del sistema.
  - 2.5. Diseño conceptual y lógico del modelo datos del sistema.
3. Implementación.
  - 3.1.1. Implementación de los componentes del sistema.
4. Pruebas.
  - 4.1. Prueba de los componentes desarrollados y del sistema general.
  - 4.2 Corrección de errores.

## 2.4 TECNOLOGÍA DE DESARROLLO

CV-UVA ha sido desarrollada casi en su totalidad con tecnologías *OpenSource*. Desde el principio, se ha optado por este tipo de producto dada la ausencia de licencias de pago.

El lenguaje de programación usado para el desarrollo de la App Android, ha sido el lenguaje de programación orientado a objetos propio de la plataforma ANDROID, que en muchos casos es prácticamente idéntico a JAVA, lo que hace posible reutilizar sus librerías. Este lenguaje tiene una gran cantidad de librerías que facilitan el desarrollo de aplicaciones, además de estar ampliamente documentado, lo que incluye disponer código de demostración sobre cuestiones básicas de ANDROID. También se ha usado XML para la definición de las vistas (interfaz de usuario).

La parte de la App Web, se basa en una arquitectura cliente-servidor. La interfaz de usuario se despliega completamente en el navegador Web. En el servidor residen todos los demás módulos: base de datos y middleware de acceso a datos, lógica de negocio...

El hecho de ser una aplicación cuya lógica está totalmente centralizada (en el servidor) confiere al sistema una gran flexibilidad, tanto de desarrollo como de acceso y utilización.

El sistema funciona sobre un servidor APACHE, con base de datos MYSQL y PHP como lenguaje de programación. Es importante tener en cuenta que PHP se puede utilizar e instalar en la mayoría de los servidores. Por ejemplo, PHP se instala como un módulo integrado de APACHE, lo que se traduce en un mayor rendimiento.

Las herramientas usadas para el desarrollo del proyecto son:

### • Hardware:

- PC de desarrollo con las características normales de un ordenador actual.
- PC como servidor web.
- Router WiFi-N.

### • Software

- “ECLIPSE IDE HELIOS” como entorno de desarrollo.
- “SDK ANDROID” como plataforma de desarrollo y emulador de *ANDROID*.
- “OPENPROJ 1.4” para la planificación del proyecto.
- “STARUML 5.0” para el modelado de la aplicación usando el lenguaje visual UML.
- “MICROSOFT OFFICE 2010” para el desarrollo de la documentación y manual de usuario.
- “ADOBE PHOTOSHOP CS5 DEMO” para la creación de los iconos de la aplicación y para algunas ilustraciones de la documentación.
- “DREAMWEAVER CS5 DEMO” para el diseño y desarrollo de la aplicación.
- “EASYPHP” (instala Apache, PHP y MySQL).

### 3. DESCRIPCIÓN GENERAL DEL PRODUCTO

El desarrollo de ambas aplicaciones se ha pensado en el cumplimiento de las siguientes características esenciales:

- Accesibilidad.
- Usabilidad.
- Escalabilidad.
- Reusabilidad de código.
- Facilidad de actualización.

#### INTERFAZ DE USUARIO

La interfaz de usuario, como mediador entre el usuario y la lógica de la aplicación, es de vital importancia a la hora de diseñar y elaborar un nuevo producto software. Aunque a veces olvidada, esta capa es la responsable de la mayoría de éxitos y fracasos de los productos software que salen al mercado. Por ello, para el desarrollo de este producto se ha prestado especial atención en que la interfaz sea eficaz, esté bien estructurada y facilite el trabajo al usuario.

El diseño de las interfaces de usuario ha de centrarse en el conocimiento del usuario y de sus capacidades. Por ello todo interfaz de usuario debe cumplir los siguientes requisitos:

- **Familiaridad.** Utilizar términos y conceptos tomados del entorno y experiencia del usuario.
- **Consistencia.** Similares acciones y conceptos han de tener similar disposición y representación en el interfaz. Esta consistencia significa que todos los elementos, comandos y menús a lo largo del sistema han de tener similar formato.
- **Comportamiento Fiable (mínima sorpresa).** El usuario no ha de ser sorprendido con el comportamiento de la interfaz.
- **Restablecimiento.** Los usuarios cometen inevitablemente errores. Por ello se han de incluir mecanismos que permitan al usuario retornar a estados anteriores que permitan corregir sus propios errores.
- **Tomar en cuenta las actualizaciones.** Un sitio o página de Internet nunca está realmente terminado. Hay que diseñar todo sitio pensando en el futuro y dejando espacio para agregar nueva información o secciones.

Se han utilizado formularios y menús de forma que el usuario tenga que escribir lo menos posible, reduciendo así también la posibilidad de errores.

En la parte web de CV\_UVA, podemos encontrarnos varios interfaz... entre ellos:

- Pantalla de inicio, donde el usuario, puede entrar en el sistema, como ya hemos comentado, a esta parte solo podrán acceder los usuarios profesores.



Ilustración 2. Pantalla inicio App Web CV\_UVA

- Una vez iniciada la sesión de forma correcta, nos encontramos con la siguiente pantalla.

Da la bienvenida al profesor



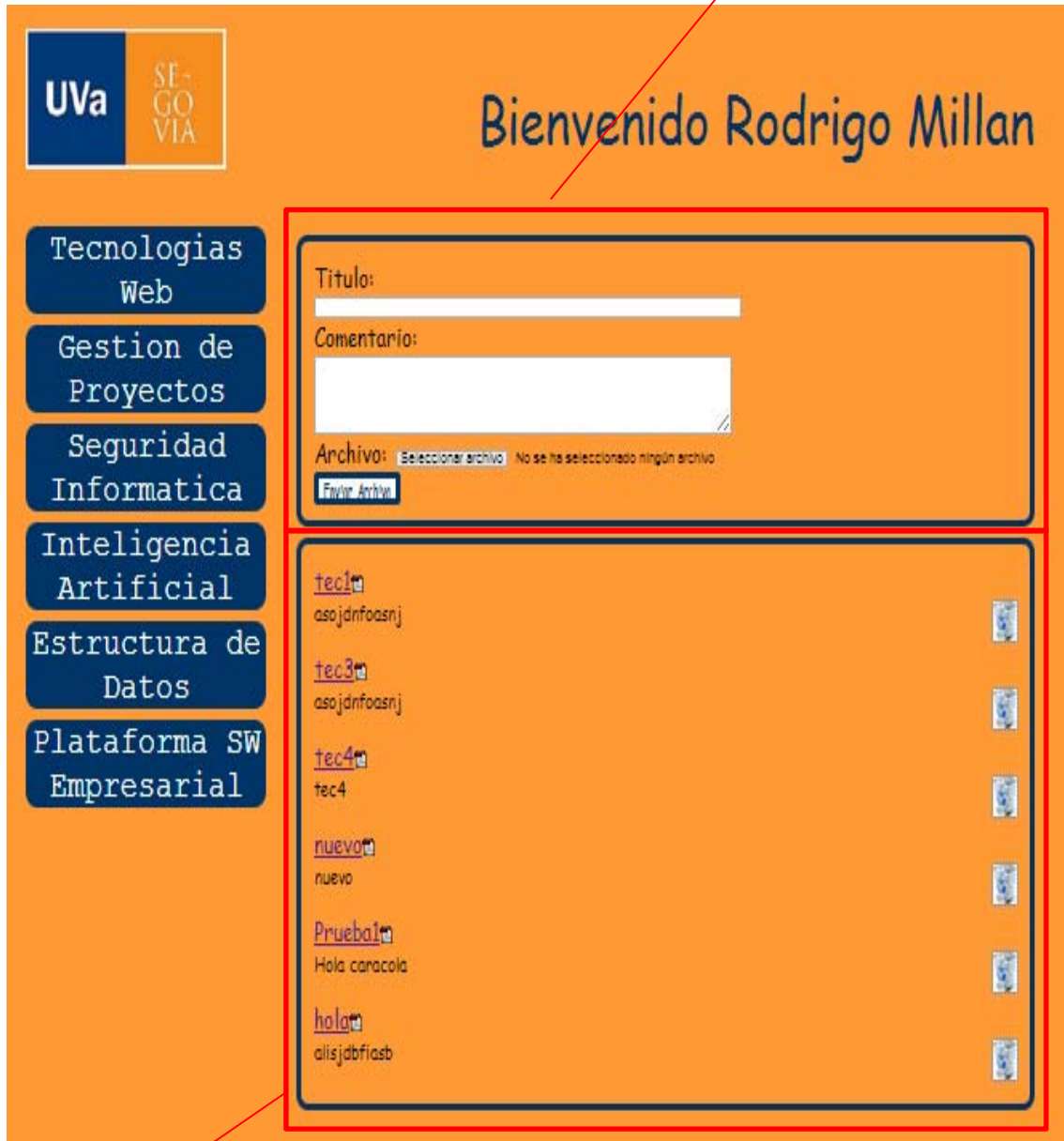
Muestra las asignaturas que dicho profesor imparte y por lo tanto, donde podrá subir/borrar archivos PDF.

Ilustración 3. Pantalla de asignaturas App Web CV\_UVA



- Una vez selecciona una asignatura, se muestra la siguiente pantalla:

Formulario para poder subir un archivo PDF nuevo.



Listado de los archivos PDF que en ese momento contiene la asignatura.

Ilustración 4. Pantalla archivos PDF por asignatura App Web CV\_UVA

En la parte de la CV\_UVA para Android, nos encontraremos con las siguientes pantallas:

- La pantalla de inicio, donde el usuario, ya sea profesor o alumno, entran en el sistema para poder consultar los archivos PDF.

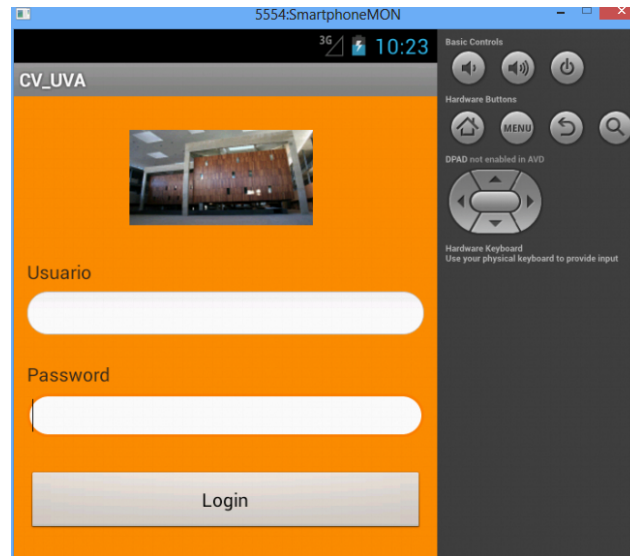


Ilustración 5. Pantalla de inicio App Android CV\_UVA

- Una vez que se ha iniciado sesión correctamente, se mostrara un listado con las asignaturas en las que está matriculado (alumno) o imparte (profesor) según corresponda.



Ilustración 6. Pantalla asignaturas App Android CV\_UVA

- Una vez que hemos seleccionado una asignatura, pasamos a otra pantalla donde se muestran los archivos.



Ilustración 7. Pantalla archivos PDF App Android CV\_UVA

- Y por último, cuando hayamos seleccionado el archivo que queremos que se nos muestre, aparece la siguiente pantalla, que pinchando en el botón, se abre el archivo.



Ilustración 8. Pantalla archivos PDF concreto App Android CV\_UVA

## ARCHIVOS PDF

Como se describe en la motivación y en los objetivos del TFG, queremos que los archivos PDF que se suban por parte del profesorado a un servidor, sean accesibles desde una App Android. Esto es, que se puedan ver desde un dispositivo móvil, ya sea el profesor o el alumno de la asignatura.

Los archivos PDF estarán organizados o clasificados, según a la asignatura a la que pertenezcan, tanto en la App Web como en la App Android.

Por ello, desde la App Web se podrán subir los archivos PDF mediante un formulario, podrán ser vistos y borrados.

En la siguiente imagen, se muestra el formulario desde donde el profesor, podrá subir archivos PDF de una asignatura, como se puede ver, contiene los campos Título y Comentario para posibles aclaraciones respecto al archivo.



Ilustración 9. Formulario para subir los archivos PDF App Web CV\_UVA

A través de la App Android, los archivos PDF solo podrán ser vistos, no se podrán modificar ni borrar, al igual que tampoco se podrán subir nuevos archivos PDF.

## USUARIOS

En ambas aplicaciones, Web como Android, los usuarios tendrán que iniciar sesión para poder acceder a los datos, tanto de asignaturas impartidas/matriculadas como a los archivos PDF que tienen cada una de las asignaturas.

## NOTIFICACIONES

Como casi todas las aplicaciones móviles que conocemos, por no decir todas, cuentan con un sistema de notificaciones, y como no podía ser menos, CV\_UVA, también cuenta con ellas.

Estas notificaciones, avisarán al usuario de la aplicación cuando se han producido cambios en los archivos existentes en la base de datos, tanto si se han añadido como si se han borrado. En la siguiente imagen, se muestra cómo quedará la notificación en el dispositivo móvil.

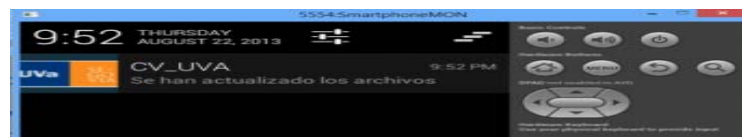


Ilustración 10. Notificación en App Android CV\_UVA

### 3.1 FUNCIONALIDADES DEL PRODUCTO

Más detalladamente, a continuación, se enumeran las funcionalidades de las que dispone el producto desarrollado.

App Web	App Android
<b>USUARIOS</b>	
Solo el profesor tiene acceso a esta App. Este usuario, podrá navegar por la aplicación, subir, modificar y borrar los archivos PDF.	Cualquiera de los dos usuarios (profesor o alumno) tiene acceso a esta App. Podrán navegar por la aplicación y visualizar los archivos PDF que se encuentran en el servidor.
<b>ARCHIVOS PDF</b>	
Los archivos PDF que estén almacenados en el servidor, podrán ser visualizados, modificados o borrados.	Los archivos PDF que estén almacenados en el servidor, solo podrán ser visualizados, no se podrán modificar ni borrar.
<b>NOTIFICACIONES</b>	
	Las notificaciones se mostrarán en el dispositivo móvil siempre que se haya producido algún tipo de cambio en los archivos PDF de la base de datos.

### 3.2 ARQUITECTURA DEL PRODUCTO

En este apartado, se detalla la arquitectura del producto desarrollado, tanto de la parte Web como de la parte Android. Esta descripción se hace a tres niveles, aportado inicialmente una visión general de la arquitectura, para posteriormente detallar su arquitectura interna.

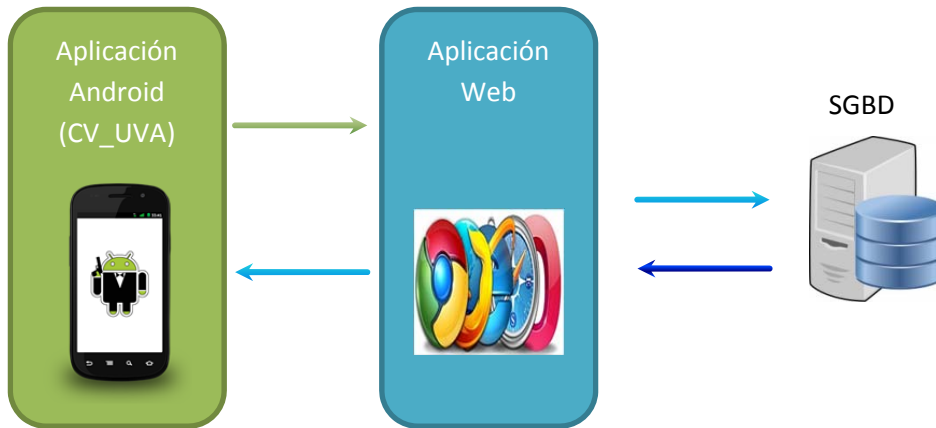


Ilustración 11. Arquitectura Gral. del TFG

En la anterior imagen, se muestra de forma general como está organizada la arquitectura general del TFG. En este punto me gustaría hablar un poco más, sin entretenerme demasiado, de cada una de las dos arquitecturas del TFG.

#### 3.2.1 Arquitectura General

### APLICACIÓN ANDROID

Android está construido sobre una serie de tecnologías abiertas. La capa inferior está basada en el núcleo de LINUX, encargado de gestionar los procesos, y será aquí donde se apoyen los drivers específicos proporcionados por los fabricantes de hardware.

Según vamos subiendo de nivel en la arquitectura de capas del sistema, nos encontramos con una serie de librerías nativas escritas en C o C++. Por encima de esta capa se encuentra el *Android Runtime* donde reside la máquina virtual DALVIK capaz de interpretar el código generado en nuestro entorno de desarrollo.

La siguiente capa se compone principalmente de las API que nos ofrece GOOGLE para manipular todos los componentes del dispositivo.

En la capa superior se encontrarán las aplicaciones JAVA, tanto las que incorpora de seria Android, como las que sean descargadas de GOOGLE PLAY y las que desarrollemos nosotros mismos.

La siguiente figura representa de forma gráfica la arquitectura del sistema:



Ilustración 12. Arquitectura Gral. App Android

## APLICACIÓN WEB

La App Web está basada en una arquitectura cliente-servidor, modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes. Un cliente realiza peticiones a otro programa, el servidor, quien le da respuesta.

En esta arquitectura la capacidad de proceso está repartida entre los clientes y los servidores, aunque son más importantes las ventajas de tipo organizativo debidas a la centralización de la gestión de la información y la separación de responsabilidades, lo que facilita y clarifica el diseño del sistema.

La separación entre cliente y servidor es una separación de tipo lógico, donde la interfaz de usuario se despliega completamente en el navegador Web. En el servidor residen todos los demás módulos: base de datos y middleware de acceso a datos, lógica de negocio, etc.

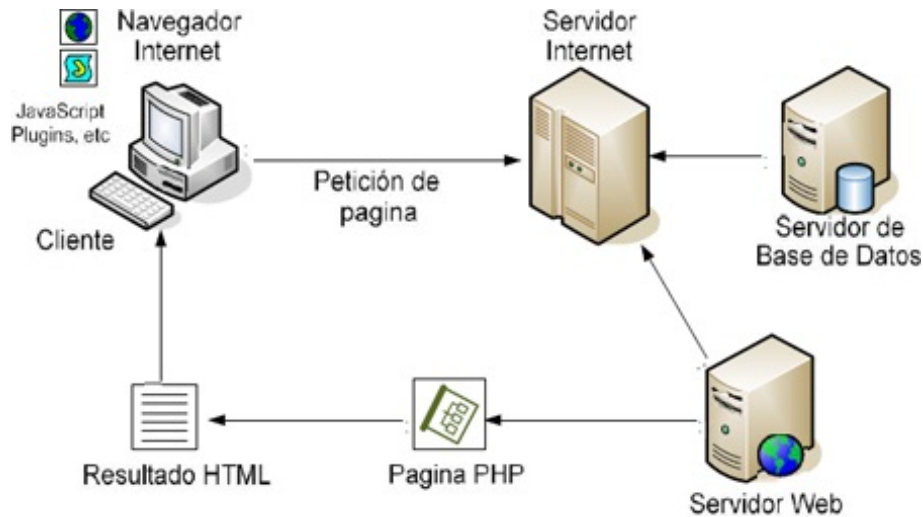


Ilustración 13. Arquitectura Gral. App Web

### 3.2.2 Arquitectura Específica

## APLICACIÓN WEB

El sistema funciona sobre un servidor APACHE, con base de datos MYSQL y PHP como lenguaje de programación. Es importante tener en cuenta que PHP se puede utilizar e instalar en la mayoría de los servidores. Por ejemplo, PHP se instala como un módulo integrado de APACHE, lo que se traduce en un mayor rendimiento.

El sistema está basado en los componentes hardware y software siguientes:

- ⊗ Ordenador cliente con conexión a Internet y un navegador del tipo MICROSOFT INTERNET EXPLORER o SAFARI.
- ⊗ Ordenador que funciona como servidor Web en el que se encuentra instalado APACHE + PHP + MYSQL.
- ⊗ Las páginas PHP, deben encontrarse almacenadas dentro del servidor Web (o en un ordenador al que tenga acceso el servidor mediante una red local) en un directorio virtual, junto con los permisos necesarios.



Cuando el usuario entra en el sistema e inicia el navegador Web, éste establece una conexión con el servidor Web. Es el propio servidor Web el que gestiona la conexión con la base de datos para poder realizar todo tipo de consultas, inserción, selección de datos.

En la siguiente figura, se puede ver de forma detallada la estructura que tiene CV\_UVA. (Web).

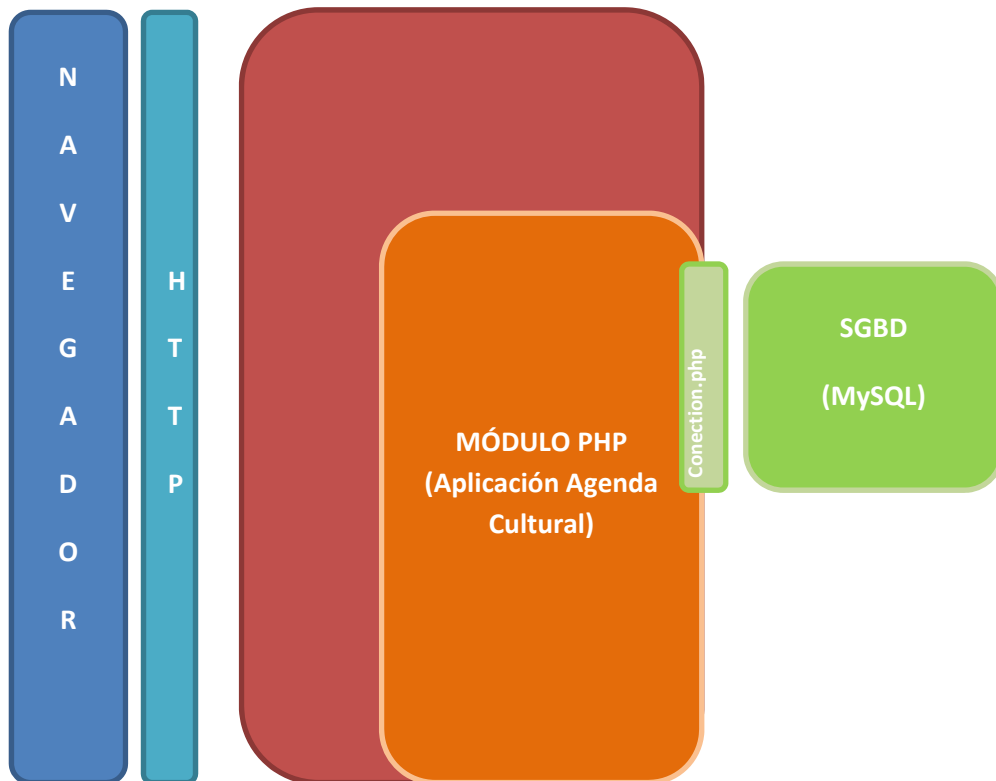


Ilustración 14. Arquitectura específica CV\_UVA (Web)

APLICACIÓN ANDROID

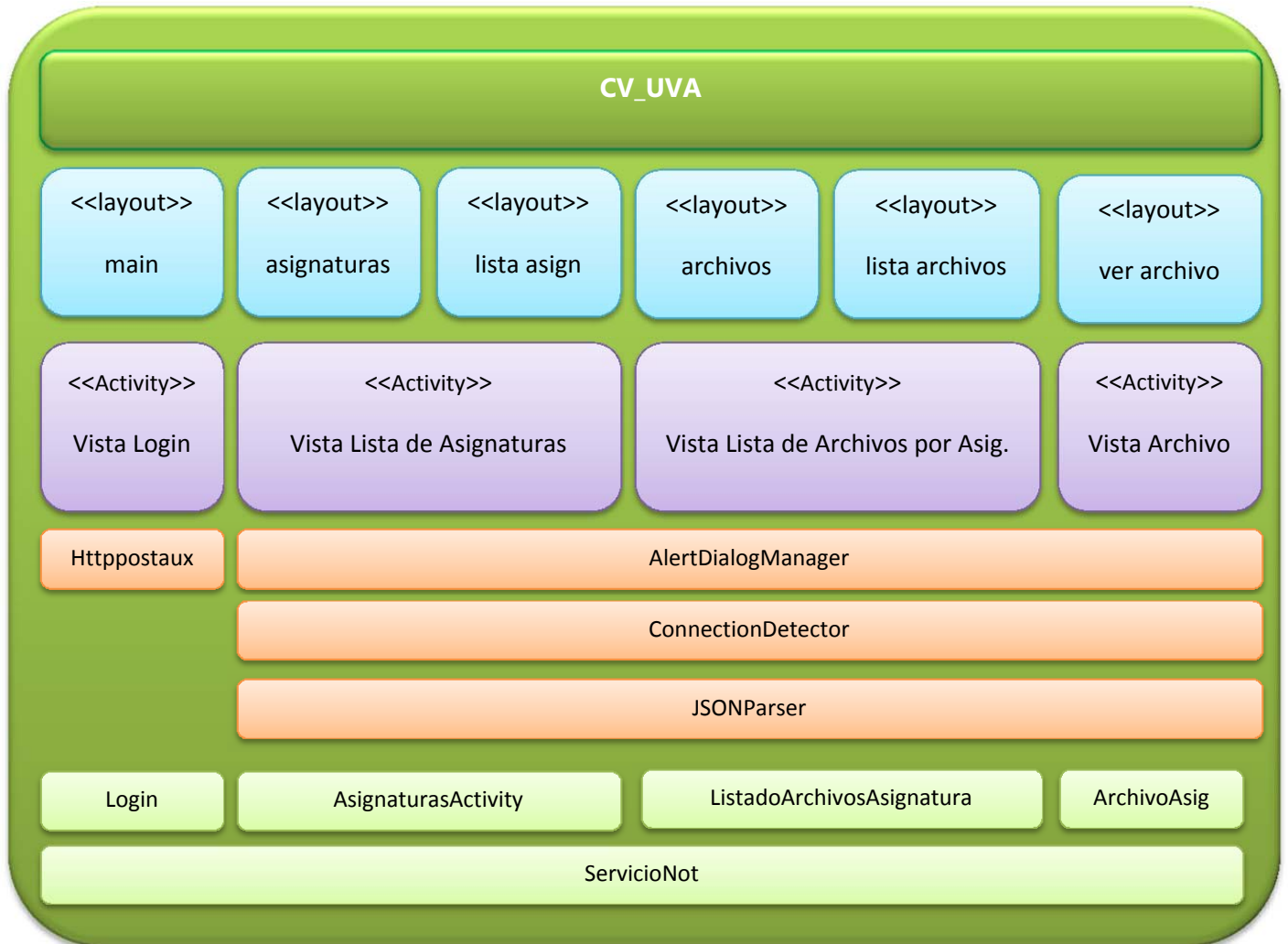


Ilustración 15. Arquitectura específica CV\_UVA (Android)

En la anterior imagen se representa la arquitectura interna del producto desarrollado, CV\_UVA. Como se puede observar, existen varios apartados que veremos a continuación.

## ● Elementos Layout

Son la definición de las interfaces de usuario. Estos elementos se definen en formato XML y en ellas se añaden, organizan y posicionan todos los componentes que forman la interfaz de usuario.

### ↳ Main.

Se trata de la definición de la pantalla principal, cuenta con dos campos para rellenar y de un botón para iniciar sesión.

### ↳ Asignaturas, ListaAsignaturas, Archivos y ListaArchivos.

Estas cuatro *layout* están compuestas por listas.

### ↳ VerArchivo.

Está compuesta por textos y por un botón.

## ● Elementos Activity

Son los elementos a los que se les define un layout específico que deben presentar en pantalla y que interactúan con el usuario implementando la funcionalidad correspondiente en los eventos de los componentes del layout.

### ↳ Vista Login.

En él se incluyen las acciones o funcionalidades requeridas para iniciar sesión, se lanza en activity Vista lista Asignatura.

### ↳ Vista Lista Asignaturas.

En él se incluyen las acciones o funcionalidades requeridas para ver las asignaturas y se lanza el activity Vista Lista Archivos.

### ↳ Vista Lista Archivos.

En él se incluyen las acciones o funcionalidades requeridas para ver los archivos que contiene esa asignatura y se lanza el activity Ver Archivo.

### ↳ Ver Archivo.

En él se incluyen las acciones o funcionalidades requeridas para ver el archivo en concreto que hayamos seleccionado.

## Clases JAVA

### Login.

Esta es, la clase encargada de realizar todo el proceso de login en CV\_UVA, para ello es necesaria la utilización de otra clase llamada HttpPostaux.

### AsignaturasActivity.

Es la clase encargada de mostrar todas las asignaturas que imparte (profesor) o se ha matriculado (alumno) en función de quien haya hecho el login. Esta clase necesita de AlertDialogManager, ConnectorDetector y JSONParse para poder realizarlo.

### ListadoArchivosAsignaturas.

Es la clase encargada de mostrar todos los archivos que hay en la asignatura que previamente hemos seleccionado. Esta clase necesita de AlertDialogManager, ConnectorDetector y JSONParse para poder realizarlo.

### ArchivoAsignatura.

Esta es la clase que nos muestra el archivo en concreto que hemos seleccionado. Una vez en esta pantalla, podemos mostrar el archivo. Esta clase necesita de AlertDialogManager, ConnectorDetector y JSONParse para poder realizarlo.

### ServicioNot

Esta clase se encarga de todo el tema de las notificaciones, éstas, notificarán a los usuarios de la aplicación CV\_UVA cuando se hayan cambiado archivos en la base de datos.

A modo de resumen, el software tendrá una arquitectura que sigue el patrón modelo-vista-controlador (MVC), que en el *framework* de ANDROID se divide en *layouts* (vistas en XML), *activities* (controladores) y las clases JAVA (modelo).

### 3.3 DESPLIEGUE DEL PRODUCTO

La aplicación desarrollada CV\_UVA (Android) está pensada para ser ejecutada en un dispositivo móvil con acceso a Internet o a una red local inalámbrica, a sea con tecnología WiFi o 3G, o cualquier otra que soporta la pila de protocolos TCP/IP.

Los servidores donde se encuentran los archivos PDF a los que accede esta aplicación pueden estar, por tanto, situados dentro o fuera de la red en la que se ejecuta la aplicación.

En la siguiente imagen se muestra un gráfico que representa la situación del hardware sobre el que se ejecutará el producto desarrollado.

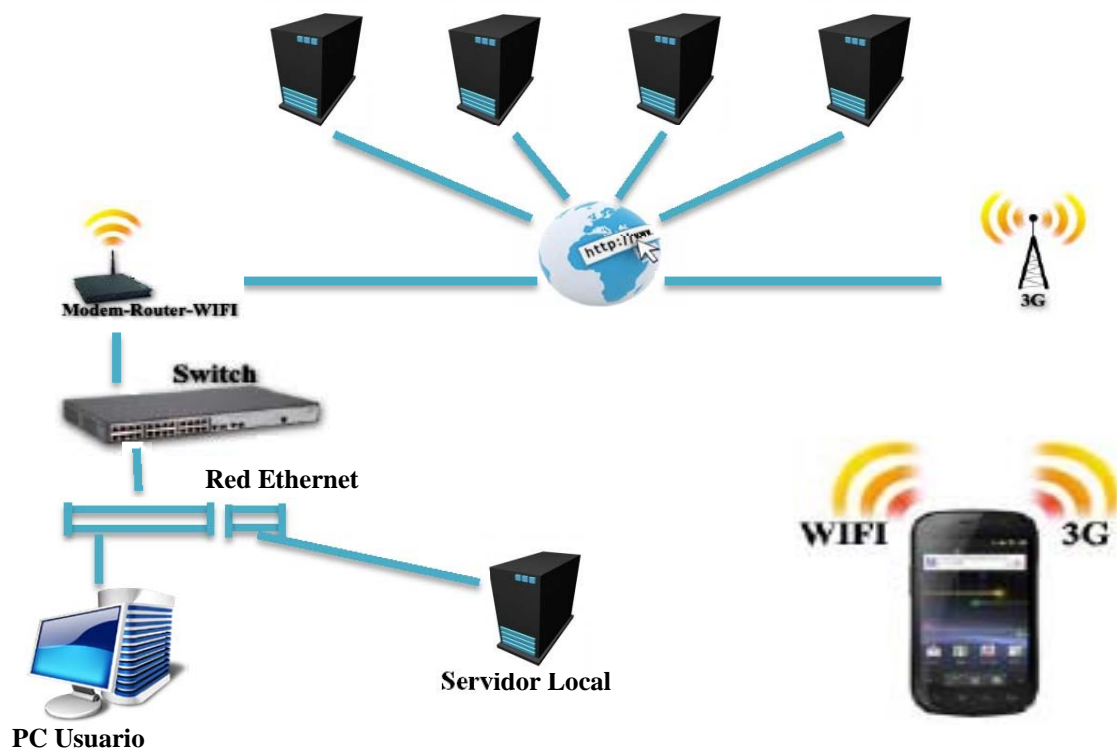


Ilustración 16. Diagrama de despliegue del producto.



## 4. PLANIFICACIÓN Y PRESUPUESTO

### 4.1 ESTIMACIÓN DE TRABAJOS

Como punto de partida para la estimación de los trabajos del proyecto, es necesario calcular el tamaño que previsiblemente tendrá nuestro sistema.

Existen varias técnicas de estimación que se basan en las características que debe tener el sistema a desarrollar, pero para este proyecto se va a utilizar la estimación por puntos de función.

#### 4.1.1 Estimación mediante puntos de función (PFA)

La estimación por puntos de función se realiza para obtener una aproximación del tamaño que pueda tener el proyecto. Esta estimación será usada más adelante para evaluar el esfuerzo y coste que supondrá desarrollar el proyecto.

##### 4.1.1.1 Explicación del proceso

Antes de realizar la estimación se va a describir qué es y cómo se va a realizar.

El método de estimación se basa en una métrica que cuantifica la funcionalidad que se debe entregar al usuario al construir la aplicación [Albrech, 1979]. Para ello, este método utiliza unos parámetros que sirven para evaluar dicha funcionalidad. Estos parámetros son:

- Número de entradas: Datos que el usuario aporta al sistema (nombre de ficheros, menús de selección, etc.).
- Número de salidas: Datos que el sistema aporta al usuario (informes, mensajes).
- Número de ficheros lógicos internos: Ficheros o bases de datos internos al sistema (es decir sólo los utiliza el sistema, ficheros maestros).
- Número de ficheros externos: Ficheros o bases de datos externos al sistema, es decir, que pueden ser “vistos” por otras aplicaciones.
- Número de consultas externas: Entradas que requieren de una respuesta por parte del sistema.

Los pasos a seguir son:

1. Se debe contar el número de elementos de cada clase.
2. Cada elemento debe ser clasificado según su grado de complejidad (alta, media o baja).
3. Por último se obtienen los PFNA (Puntos de Función No Ajustados) mediante una suma ponderada de esas cantidades con los pesos que aparecen a continuación:

Parámetro significativo	Complejidad baja	Complejidad media	Complejidad alta
Entradas	x3	x4	x6
Salidas	x4	x5	x7
Ficheros internos	x7	x10	x15
Ficheros externos	x5	x7	x10
Consultas externas	x3	x4	x6

Ilustración 17. Multiplicadores de complejidad de la estimación por puntos de función.

Los criterios para evaluar la complejidad de los elementos de cálculo son los siguientes:

<b>Ficheros lógicos externos e internos.</b>	<b>Registros elementales</b>	<b>Datos elementales</b>		
	1	1-9	20-50	>51
	2-5	Baja	Baja	Media
	>6	Baja	Media	Alta
<b>Salidas y consultas</b>	<b>Tipos de ficheros</b>	<b>Datos elementales</b>		
	0-1	1-5	6-19	>20
	2-3	Baja	Baja	Media
	>4	Baja	Media	Alta
<b>Entradas</b>	<b>Tipos de ficheros</b>	<b>Datos elementales</b>		
	0-1	1-4	5-15	>16
	2-3	Baja	Baja	Media
	>3	Baja	Media	Alta

Ilustración 18. Tabla para valorar la complejidad de las características del proyecto.

Una vez calculados los parámetros y su complejidad los sumamos mediante la siguiente suma ponderada:

$$\begin{aligned}
 \text{PFNA} = & (\text{N}^\circ \text{ Entradas} \times \text{multiplicador (complejidad)}) + \\
 & (\text{N}^\circ \text{ Salidas} \times \text{multiplicador (complejidad)}) + \\
 & (\text{N}^\circ \text{ Fichero internos} \times \text{multiplicador (complejidad)}) + \\
 & (\text{N}^\circ \text{ Ficheros externos} \times \text{multiplicador (complejidad)}) + \\
 & (\text{N}^\circ \text{ Consultas externas} \times \text{multiplicador (complejidad)}).
 \end{aligned}$$



Los PFNA deben ser ajustados mediante un Factor de Ajuste (FA).

El factor de ajuste se obtiene de la suma de los 14 factores de complejidad (FC), que se describen más adelante, mediante la siguiente ecuación:

$$FA = (0,01 \times \Sigma FC) + 0,65$$

Los puntos de función ajustados (PF) se obtienen de la siguiente forma:

$$PF = PFNA \times FA$$

Existen 14 factores que contribuyen a la complejidad de una aplicación, cada uno de ellos valorados dentro de una escala de 0 a 5.

## **FACTORES DE COMPLEJIDAD:**

### **1. Comunicación de Datos**

- 0 Aplicación es batch exclusivamente.
- 1-2 Impresión o entrada de datos remota.
- 3-5 Teleproceso (TP) interactivo.
- 3 TP interface a un proceso batch.
- 5 La aplicación se interactiva.

### **2. Rendimiento (referido a la importancia de respuesta dentro de todo el sistema)**

- 0-3 Análisis y diseño de las consideraciones del rendimiento son estándar. No se requieren requerimientos especiales por parte del usuario.
- 4 En la fase de diseño se incluyen tareas del análisis del rendimiento para cumplir los requerimientos del usuario.
- 5 Además se utilizan herramientas de análisis del rendimiento en el diseño, desarrollo e instalación.

### **3. Frecuencia de Transacciones**

- 0-3 Las tasas son tales que las consideraciones de análisis de rendimiento son estándares.
- 4 En la fase de diseño se incluyen tareas de análisis de rendimiento para verificar las altas tasas de transacciones.
- 5 Además se utilizan herramientas de análisis del rendimiento.

### **4. Requisitos de manejo del usuario final.**

- 0 Sistema batch.
- 1-3 No se especifican requerimientos especiales.
- 4 Se incluyen tareas de diseño para la consideración de factores humanos.
- 5 Además se utilizan herramientas especiales o de prototipado para promover la eficiencia.

## 5. Procesos complejos

¿Qué características tiene la aplicación?

Mucho procesamiento matemático y/o lógico

Muchas excepciones de procesamiento, muchas transacciones incompletas y mucho reprocesamiento de las transacciones

Procesamiento de seguridad y/o control sensitivo

- 0 No se aplica nada de esto.
- 1-3 Se aplica alguna cosa.
- 4 Se aplican dos cosas.
- 5 Se aplica todo.

## 6. Facilidad de mantenimiento e instalación

- 0-1 No se requieren por parte del usuario facilidades especiales de conversión e instalación.
- 2-3 Los requerimientos de conversión e instalación fueron descritos por el usuario y se proporcionaron guías de conversión e instalación.
- 4-5 Además se proporcionaron y probaron herramientas de conversión e instalación.

## 7. Instalación en múltiples lugares. Añadir puntos por cada uno de los siguientes factores:

- 0 El usuario no requiere la consideración de más de un puesto
- 1 De uno a cuatro puestos.
- 2 Cinco o más puestos.
- 1 Se proporciona documentación y plan de apoyo para soportar la aplicación en varios lugares.
- 2 Los puestos están en países diferentes.

## 8. Funciones distribuidas. "Distribuida" significa que los componentes de la aplicación están distribuidos en dos o más procesadores diferentes.

- 0 La aplicación no ayuda a la transferencia de datos o a la función de procesamiento entre los componentes del sistema.
- 1 La aplicación prepara datos para el usuario final de otro procesador.
- 2-3 Los datos se preparan para transferencia, se transfieren y se procesan en otro componente del sistema.
- 4 Igual que 2-3, pero con realimentación al sistema inicial.
- 5 Las funciones de procesamiento se realizan dinámicamente en el componente más apropiado del sistema.

### **9. Gran carga de trabajo (referente a la importancia del entorno)**

- 0-3 La aplicación corre en una máquina estándar sin restricciones de operación.
- 4 Restricciones de operación requieren características específicas de la aplicación en el procesador central.
- 5 Además hay restricciones específicas a la aplicación en los componentes distribuidos del sistema.

### **10. Entrada interactiva de datos**

- 0-2 Hasta el 15% de las transacciones tienen entrada interactiva.
- 3-4 15% al 30% tienen entrada interactiva.
- 5 30% al 50% tienen entrada interactiva.

### **11. Actualizaciones On-Line**

- 0 Nada.
- 1-2 Actualización on-line de los ficheros de control. El volumen de actualización es bajo y la recuperación fácil.
- 3 Actualización on-line de la mayoría de los ficheros internos lógicos.
- 4 Además es esencial la protección contra la pérdida de datos.
- 5 Además se considera el coste de recuperación de volúmenes elevados.

### **12. Utilización con otros sistemas (el código se diseña para que sea compartido o utilizable por otras aplicaciones. No confundir con 13).**

- 0-1 Una aplicación local que responde a las necesidades de una organización usuaria.
- 2-3 La aplicación utiliza o produce módulos comunes que consideran más necesidades que las del usuario.
- 4-5 Además, la aplicación se "empaquetó" y documentó con el propósito de fácil reutilización.

### **13. Facilidad de Operación**

- 0 No se especifican por parte del usuario consideraciones específicas de operación.
- 1-2 Se requieren, proporcionan y prueban procesos específicos de arranque, backup y recuperación.
- 3-4 Además la aplicación minimiza la necesidad de actividades manuales, tales como instalación de cintas y papel.
- 5 La aplicación se diseña para operación sin atención.

**14. Facilidad de Cambio (esfuerzo específico de diseño para facilitar cambios futuros). Añadir puntos por cada uno de los siguientes factores:**

- 0-2 No hay requerimientos especiales del usuario para minimizar o facilitar el cambio.
- 3-4 Se proporciona capacidad de consulta flexible.
- 5 Datos importantes de control se mantienen en tablas que son actualizadas por el usuario a través de procesos on-line interactivos.

Por último, podemos estimar las líneas de código en función de los puntos de función. Como cada lenguaje tiene sus particularidades, las líneas de código equivalentes a un punto de función no pueden ser las mismas para todos los lenguajes. Por ello Casper Jones realizó una tabla con las equivalencias que vemos a continuación.

Lenguaje	LDC/PF
Ensamblador	320
C	150
Cobol	106
Pascal	91
Basic	64
TCL	64
Java	53
C++	29

Ilustración 19. Tabla de correspondencia entre puntos de función y Líneas De Código (LDC).

En nuestro caso, como también tenemos programación en PHP, tenemos que buscar la equivalencia de LDC/PF, después de una búsqueda por internet, encuentro que, según un documento al que se hace referencia en ["http://www.slideshare.net/cancerf8/bitacoraelectronica"](http://www.slideshare.net/cancerf8/bitacoraelectronica):

La equivalencia dice:	12 líneas de código PHP = 1 Punto de Función
-----------------------	--

Ilustración 20. Correspondencia entre puntos de función y Líneas de Código en PHP.

### 4.1.1.2 Estimación por puntos de función

Tanto este apartado como la aplicación del COCOMO se hará por duplicado, puesto que tenemos que hacer para cada una de ellas, dos estimaciones, la de la App Web y la de la App Android.

Comenzamos obteniendo los parámetros que nos permitirán evaluar la funcionalidad del desarrollo software:

#### APLICACIÓN WEB

Entradas	Complejidad	Salidas	Complejidad
Nuevo Archivo PDF	Media	Listado de asignaturas	Media
Borrar Archivo PDF	Media	Listado de archivos PDF	Media
Envío datos formulario	Alta	Mostrar formulario	Media
Login	Media	Errores	Media

Consultas	Complejidad
Asignaturas	Simple
Archivos	Simple
Usuarios	Simple

Ficheros Lógicos Internos	Complejidad	Ficheros Lógicos Externos	Complejidad
Base de datos	Alta	Mostrar PDF	Medio

Para obtener los PFNA se realiza la suma de los productos del número de parámetros de cada tipo por su multiplicador de complejidad.

$$PFNA = (3_{in} \cdot 4) + (1_{in} \cdot 6) + (4_{out} \cdot 5) + (3_q \cdot 3) + (1_{fii} \cdot 15) + (1_{fle} \cdot 7) = 69$$

Obtenemos el Factor de Ajuste valorando los factores de complejidad.

<b>Factores de complejidad</b>	<b>0-5</b>	<b>Factores de complejidad</b>	<b>0-5</b>
Comunicación de datos	5	Funciones distribuidas	0
Rendimiento	1	Gran carga de trabajo	3
Frecuencia de transacciones	1	Entrada on-line de datos	5
Requisitos manejo usuario	3	Actualización on-line	5
Procesos complejos	2	Utilización con otros sist.	4
Facilidad de mantenimiento	1	Facilidad de operación	0
Instalación múltiples lugares	0	Facilidad de cambio	2

$$\Sigma FC = 32$$

$$FA = (0.01 \times 32) + 0.65 = 0.97$$

Los puntos de función ajustados (PF) se obtienen de la siguiente forma:

$$PF = PFNA \times FA = 69 \times 0.97 = 66.93$$

Ahora calculamos las líneas de código PHP para cada punto de función. Para ello sacamos la equivalencia de la ilustración 20.

$$1PF = 12 \text{ Líneas de código en PHP.}$$

$$LDC = 66.93 \times 12 = 803.16 \cong 803 \text{ Líneas de código PHP.}$$

## APLICACIÓN ANDROID

Entradas	Complejidad	Salidas	Complejidad
Login	Alta	Listado de asignaturas	Alta
		Listado de archivos PDF	Alta
		Errores	Media

Consultas	Complejidad
Asignaturas	Alta
Archivos	Alta
Usuarios	Alta

Ficheros Lógicos Internos	Complejidad	Ficheros Lógicos Externos	Complejidad
Base de datos	Alta	Mostrar PDF	Alto

Para obtener los PFNA se realiza la suma de los productos del número de parámetros de cada tipo por su multiplicador de complejidad.

$$PFNA = (1_{in} \cdot 6) + (2_{out} \cdot 7) + (1_{out} \cdot 5) + (3_q \cdot 6) + (1_{fii} \cdot 15) + (1_{fle} \cdot 10) = 68$$

Obtenemos el Factor de Ajuste valorando los factores de complejidad.

Factores de complejidad	0-5	Factores de complejidad	0-5
Comunicación de datos	5	Funciones distribuidas	0
Rendimiento	1	Gran carga de trabajo	3
Frecuencia de transacciones	1	Entrada on-line de datos	5
Requisitos manejo usuario	3	Actualización on-line	5
Procesos complejos	3	Utilización con otros sist.	4
Facilidad de mantenimiento	1	Facilidad de operación	0
Instalación múltiples lugares	0	Facilidad de cambio	2

$$\Sigma FC = 33$$

$$FA = (0.01 \times 33) + 0.65 = 0.98$$

Los puntos de función ajustados (PF) se obtienen de la siguiente forma:

$$PF = PFNA \times FA = 68 \times 0.98 = 66.64$$

Ahora calculamos las líneas de código Java (que es muy parecido al lenguaje de ANDROID) por cada punto de función. Para ello, sacamos la equivalencia de la tabla que realizó Casper Jones.

$$1PF = 53 \text{ Líneas de código en Java}$$

$$LDC = 66.64 \times 53 = 3.521,92 \cong 3.522 \text{ Líneas de código Java.}$$

## 4.1.2 Estimación de costes por COCOMO

En este apartado se realiza una estimación del esfuerzo y tiempo que supondrá realizar el proyecto software. Para ello, este método se basa en una estimación previa del tamaño del software en líneas de código (LDC).

### 4.1.2.1 Explicación del algoritmo COCOMO

El algoritmo de COCOMO varía en función de las características del sistema que se va a desarrollar. En concreto, este modelo de estimación diferencia entre sistemas orgánicos, empotrados y semi-libres.

La Tabla 5 muestra los tipos de desarrollos y sus correspondientes valores para los parámetros A, B y C de las expresiones utilizadas para estimar el esfuerzo y tiempo de desarrollo.

Modo de desarrollo	A	B	C
Orgánico	3,20	1,05	0,38
Empotrado	3,00	1,12	0,35
Semi-Libre	2,80	1,20	0,32

Ilustración 21. Modos de desarrollo contemplados por COCOMO

Fórmulas:

- Esfuerzo nominal [personas x mes] =  $A \times (KLDC)^B$
- Esfuerzo [personas x mes] = Esfuerzo Nominal x  $\prod$  Factores de coste
- Tiempo de desarrollo [meses] =  $2,5 \times \text{Esfuerzo}^C$
- Nº medio de personas [personas] = Esfuerzo / Tiempo de desarrollo



Los factores de coste para el cálculo del esfuerzo se obtienen de la siguiente tabla:

Factores	Valor de los factores					
	Muy bajo	Bajo	Medio	Alto	Muy alto	Extra
Fiabilidad requerida	0,75	0,88	1	1,15	1,4	
Tamaño de la base de datos		0,94	1	1,08	1,16	
Complejidad del software	0,70	0,85	1	1,15	1,30	1,65
Restricciones de tiempo de ejecución			1	1,11	1,30	1,66
Restricciones de memoria			1	1,06	1,21	1,56
Volatilidad del hardware		0,87	1	1,15	1,30	
Restricciones de tiempo de respuesta		0,87	1	1,07	1,15	
Calidad de los analistas	1,46	1,19	1	0,86	0,71	
Experiencia con el tipo de aplicación	1,29	1,13	1	0,91	0,82	
Experiencia con el hardware	1,21	1,10	1	0,90		
Experiencia con el lenguaje de programación.	1,14	1,07	1	0,95		
Calidad de los programadores	1,42	1,17	1	0,86	0,70	
Técnicas modernas de programación	1,24	1,10	1	0,91	0,82	
Empleo de herramientas	1,24	1,10	1	0,91	0,83	
Restricciones a la duración del proyecto	1,23	1,08	1	1,04	1,10	

Ilustración 22. Tabla con los valores correspondientes a los factores de coste que intervienen en el cálculo del esfuerzo.

#### 4.1.2.2 Aplicación de COCOMO

Al igual que en el apartado anterior, realizaremos COCOMO por duplicado.

Para este proyecto se elige el modo orgánico puesto que este modo esta aconsejado en:

- Desarrollos con entorno estable.
- No demasiada innovación técnica.
- Escasas presiones de tiempo.
- Tamaño relativamente pequeño (<50 KLDC, Kilo Líneas De Código).

## APLICACIÓN WEB

La estimación de KLDC para este proyecto es de 0,803 KLDC (estimación obtenida en los puntos de función).

$$\text{Esfuerzo nominal} = 3,2 \times (0,803)^{1,05} = 2,54 \text{ personas} \bullet \text{mes}$$

$$\begin{aligned} \text{Esfuerzo} &= 2,54 \times 1,15 \text{ (Fiabilidad requerida)} \times 1,08 \text{ (tamaño de la base de datos)} \\ &\times 0,85 \text{ (Complejidad software)} \times 0,86 \text{ (calidad de los analistas)} \times 0,91 \\ &\text{(experiencia con la aplicación)} \times 1,10 \text{ (Experiencia con el hardware)} \times \\ &0,95 \text{(experiencia con el lenguaje de programación)} \times 1,07 \text{ (Restricciones de} \\ &\text{tiempo de respuesta)} \times 0,86 \text{ (calidad de los programadores)} \times 0,83 \text{ (Empleo de} \\ &\text{herramientas)} = 1.67 \text{ personas} \bullet \text{mes} \end{aligned}$$

$$\text{Tiempo de desarrollo} = 2,5 \times 1,67^{0,38} = 3.04 \text{ meses}$$

Mediante el esfuerzo y el tiempo de desarrollo se puede hacer una estimación del número de personas necesarias cada mes para el desarrollo.

Nº de personas = 1,67 personas • mes / 3,04 meses = 0,54 personas al mes para realizar el proyecto en 3,04 meses.

## APLICACIÓN ANDROID

La estimación de KLDC para este proyecto es de 3,522 KLDC (estimación obtenida en los puntos de función)

$$\text{Esfuerzo nominal} = 3,2 \times (3,522)^{1,05} = 12,002 \text{ personas} \bullet \text{mes}$$

$$\begin{aligned} \text{Esfuerzo} &= 12,002 \times 1,15 \text{ (Fiabilidad requerida)} \times 1,08 \text{ (tamaño de la base de} \\ &\text{datos)} \times 1,15 \text{ (Complejidad software)} \times 0,86 \text{ (calidad de los analistas)} \times 0,91 \\ &\text{(experiencia con la aplicación)} \times 1,10 \text{ (Experiencia con el hardware)} \times \\ &0,95 \text{(experiencia con el lenguaje de programación)} \times 1,07 \text{ (Restricciones de} \\ &\text{tiempo de respuesta)} \times 0,86 \text{ (calidad de los programadores)} \times 0,83 \text{ (Empleo de} \\ &\text{herramientas)} = 11,95 \text{ personas} \bullet \text{mes} \end{aligned}$$

$$\text{Tiempo de desarrollo} = 2,5 \times 11,95^{0,38} = 6,42 \text{ meses}$$

Mediante el esfuerzo y el tiempo de desarrollo se puede hacer una estimación del número de personas necesarias cada mes para el desarrollo.

Nº de personas = 11,95 personas • mes / 6,42 meses = 1.86 personas al mes para realizar el proyecto en 3,04 meses.

## 4.2 PLANIFICACIÓN

En este apartado vamos a detallar la planificación temporal del presente proyecto.

Desde la planificación temporal, el trabajo es dividido en una serie de unidades que podemos medir de forma cuantitativa. Estas unidades son las tareas. Cada tarea tiene una duración determinada y su inicio y fin pueden estar condicionados por el resto de tareas del desarrollo. De esta forma, podemos definir que una tarea no pueda ser iniciada hasta que se haya completado otra, o bien, que dos tareas puedan ser iniciadas a la vez.

Adicionalmente, la planificación es usada como base para realizar un seguimiento del estado del proyecto durante su fase de desarrollo. Como consecuencia, si una determinada tarea se extiende más de lo planificado se pueden realizar ajustes que queden reflejados en los plazos y costes reales del proyecto.

Para el proyecto que se describe en este documento, las tareas identificadas y planificadas se pueden observar en la siguiente imagen.

		Nombre	Duración	Inicio	Terminado	Predecesores
1		Documentación	91 days	25/02/13 8:00	1/07/13 17:00	
2		<input checked="" type="checkbox"/> Aplicación web	22 days	<b>26/02/13 8:00</b>	<b>27/03/13 17:00</b>	
3		Análisis de requisitos	2 days	26/02/13 8:00	27/02/13 17:00	
4		Diseño de la base de datos	1 day	28/02/13 8:00	28/02/13 17:00	3
5		Diseño de la interfaz de usuario	2 days	1/03/13 8:00	4/03/13 17:00	4
6		Análisis del modelo de datos	1 day	5/03/13 8:00	5/03/13 17:00	5
7		Implementación	16 days	6/03/13 8:00	27/03/13 17:00	6
8		<input checked="" type="checkbox"/> Aplicación Android	67 days	<b>28/03/13 8:00</b>	<b>28/06/13 17:00</b>	
9		Fase de estudio	67 days	28/03/13 8:00	28/06/13 17:00	
10		Fase de análisis	16 days	28/03/13 8:00	18/04/13 17:00	
11		Fase de diseño	16 days	19/04/13 8:00	10/05/13 17:00	10
12		Fase de implementación	35 days	13/05/13 8:00	28/06/13 17:00	11

Ilustración 23. Tareas planificadas.

Como se puede observar en la ilustración 23, el proyecto está dividido en 2 iteraciones claramente diferenciadas y cada una de ellas, con diferentes fases.

### 1. Aplicación Web:

A lo largo de toda esta iteración, se ha realizado la parte de la aplicación web, teniéndola terminada antes de comenzar con la aplicación Android, puesto que es en esta donde se puebla la BD para poder utilizarla tanto en esta parte de la aplicación como en la de Android.

Esta parte no ha necesitado una fase de estudio previa puesto que ya tenía conocimientos de las tecnologías a emplear.

### 2. Aplicación Android:

Como se puede ver en la ilustración 23, esta iteración supone mucho más tiempo que la parte de la aplicación Web.

Esto es debido al completo desconocimiento por mí parte tanto de las herramientas a utilizar como el lenguaje necesario para poder implementarlo. Por esto, la prima fase (estudio) se alarga a durante toda la iteración puesto que es necesario ir estudiando a lo largo que se diseña e implementa.

La tarea de **documentación**, dado que se extenderá durante todo el desarrollo, se ha planificado para que tenga una duración igual a la del proyecto completo.

En la siguiente imagen, se muestra el diagrama de GANTT donde se observan las dependencias entre las tareas.

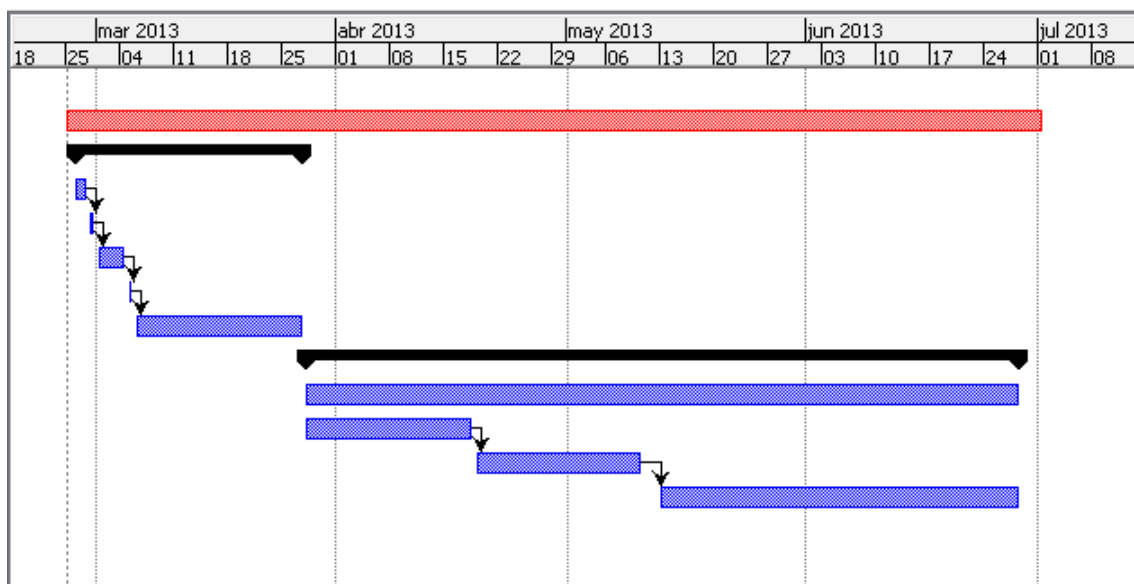


Ilustración 24. Diagrama de GANTT

### 4.3 PRESUPUESTO

En este apartado se va a cuantificar económicamente el coste total del proyecto. Para ello, hay que sumar los recursos materiales y humanos que intervienen en el mismo.

Los cálculos se hacen en función del tiempo empleado para la realización del proyecto, especificado en el anterior apartado.

#### Recursos materiales

Nombre	Uso (%)	Coste total (€)	Coste para proyecto (€)
Ordenador personal	30	900,00	270,00
Impresora HP	10	60,00	6,00
Internet Fibra Óptica	30	240,00	72,00
Microsoft 8 (incluido en el coste del ordenador personal)	100	0,00	0,00
Eclipse IDE	100	0,00	0,00
SDK Android	100	0,00	0,00
Microsoft Office 2010	25	495,00	123,75
StarUML 5.0	100	0,00	0,00
OpenProj	100	0,00	0,00
Material de oficina	100	100,00	100,00
EasyPHP	100	0,00	0,00
Dreamweaver CS5	50	488,00	244,00

#### Recursos humanos

Todo el proyecto se ha de realizar por personal cualificado, es decir, necesitamos un Ingeniero Técnico en Informática.

Nombre	Sueldo bruto anual (€)	Sueldo para el proyecto (€)
Ingeniero Técnico	24.000,00	9.000,00

#### Coste total

Especificación	Coste
Recursos materiales	815,75
Recursos humanos	9.000,00
<b>TOTAL</b>	<b>9.815,75</b>



## 5. CUESTIONES DE DISEÑO RESEÑABLES

### MODELO DE DATOS

La elección del sistema gestor de la base de datos, tiene una repercusión determinante, tanto en el rendimiento de nuestra aplicación como en su mantenimiento. La elección es, por consiguiente, una decisión crítica en la estrategia cliente/servidor global.

El motor más potente como gestor de base de datos, puede ser el más difícil de administrar o puede ser simplemente excesivo para nuestra aplicación.

El modelo de datos de CV\_UVA es bastante simple, únicamente es necesario almacenar la información relativa a los usuarios de las aplicaciones (Web/Android) y los archivos PDF. Por ello, se ha diseñado la aplicación para usar una base de datos MySQL.

El principal motivo por el que seleccioné este Gestor de Base de Datos fue que es gratuito y de libre distribución.

Otras características importantes es que consume MUY POCOS RECURSOS, tanto de CPU como de memoria y su VELOCIDAD.

Ventajas:

- ✚ Mayor rendimiento. Mayor velocidad tanto al conectar con el servidor como al servir selects y demás.
- ✚ Utilidades de administración (backup, recuperación de errores, etc.)
- ✚ Aunque se cuelgue, no suele perder información ni corromper los datos.
- ✚ Buena integración con PHP.
- ✚ No hay límites en el tamaño de los registros.
- ✚ Buen control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- ✚ MySQL se comporta bien a la hora de modificar o añadir campos a una tabla “en caliente”.

Inconvenientes:

- ✚ No soporta transacciones, “*roll-backs*” ni subselects.
- ✚ No considera las claves ajenas. Ignoras la integridad referencial, dejándola en manos del programador de la aplicación.

## PHP (*Hypertext Pre-processor*).

El lenguaje PHP es un lenguaje de programación de estilo clásico, un lenguaje de programación con variables, sentencias condicionales, bucles, funciones, etc. No es un lenguaje de marcas como podría ser HTML, XML o WML. Está más cercano a JAVASCRIPT o a C, para aquellos que conocen estos lenguajes.

Pero a diferencia de JAVA o JAVASCRIPT que se ejecutan en el navegador, PHP se ejecuta en el servidor, por eso nos permite acceder a los recursos que tenga el servidor, en nuestro caso, una base de datos. El programa PHP es ejecutado en el servidor y el resultado enviado al navegador. PHP, es lo que se llama un lenguaje de servidor.

Al ser PHP un lenguaje que se ejecuta en el servidor no es necesario que su navegador lo soporte, es independiente del navegador, pero sin embargo para que sus páginas PHP funcionen, el servidor donde están alojadas, debe soportar PHP.

Las cuatro características a tener en cuenta en un lenguaje de servidor son:

- ✦ **Velocidad:** No sólo la velocidad de ejecución, la cual es importante, sino además no crear demoras en la máquina. Por esta razón no debe requerir demasiados recursos de sistema. PHP se integra muy bien junto a otro software, especialmente bajo ambientes UNIX, cuando se configura como módulo de APACHE, está listo para ser utilizado.
- ✦ **Estabilidad:** La velocidad no sirve de mucho si el sistema se cae cada cierta cantidad de ejecuciones. Ninguna aplicación es 100% libre de *bugs*, pero teniendo de respaldo una increíble comunidad de programadores y usuarios es mucho más difícil para los bugs sobrevivir. PHP utiliza su propio sistema de administración de recursos y dispone de un sofisticado método de manejo de variables, conformando un sistema robusto y estable.
- ✦ **Seguridad:** El sistema debe poseer protecciones contra ataques. PHP provee diferentes niveles de seguridad, que pueden ser configurados desde el archivo `.ini`.
- ✦ **Simplicidad:** Se les debe permitir a los programadores generar código productivamente en el menor tiempo posible. Usuarios con experiencia en C y C++ podrán utilizar PHP rápidamente.

Otra característica a tener en cuenta sería la conectividad. PHP dispone de una amplia gama de librerías, y agregarle extensiones es muy fácil.



Las ventajas adicionales de PHP son:

- ▶ PHP corre en cualquier plataforma utilizando el mismo código fuente, pudiendo ser compilado y ejecutado en algo así como 25 plataformas, incluyendo diferentes versiones de UNIX, WINDOWS y MACS. Como en todos los sistemas se utiliza el mismo código base, los scripts pueden ser ejecutados de manera independiente al sistema operativo.
- ▶ La sintaxis de PHP es similar a la de C. Por esto, cualquiera con experiencia en lenguajes del estilo C, podrá entender rápidamente PHP.
- ▶ PHP es completamente expandible. Está compuesto de un sistema principal, un conjunto de módulos y una variedad de extensiones de código.
- ▶ Muchas interfaces distintas para cada tipo de servidor. PHP actualmente se puede ejecutar bajo APACHE, IIS, AOLSERVER, ROXEN y THTTPD. Otra alternativa es configurarlo como módulo
- ▶ Puede interactuar con muchos motores de bases de datos tales como MYSQL, MS SQL, ORACLE, INFORMIX, POSTGRESQL, y otros muchos.
- ▶ Una gran variedad de módulo cuando un programador PHP necesite una interface para una librería en particular, fácilmente podrá crear una API para esta. Algunas de las que ya vienen implementadas permiten manejo de gráficos, archivos PDF, FLASH, XML, IMAP, POP, etc.
- ▶ Rapidez. PHP generalmente es utilizado como módulo de APACHE, lo que lo hace extremadamente veloz. Está completamente escrito en C, así que se ejecuta rápidamente utilizando poca memoria.
- ▶ PHP es *Open Source*, lo cual significa que el usuario no depende de una compañía específica para arreglar cosas que no funcionan, además no estás forzado a pagar actualizaciones anuales para tener una versión que funcione.

Algunas desventajas de PHP son:

- ▶ El manejo de errores no es tan sofisticado como COLD FUSION o ASP.
- ▶ No existe IDE o *Debugger*.

Las aplicaciones Web escritas en PHP constan de un conjunto de scripts que interactúan con las fuentes que les proveen el contenido (bases de datos, archivos de disco, ficheros XML, etc.), procesan estos contenidos y generan una salida HTML que es la que se envía al navegador.

Asimismo, los datos que el usuario envía al sitio web, pueden ser procesados y almacenados en estas fuentes de datos, proporcionando dinamismo en ambos sentidos:

aplicación web → usuario y usuario → aplicación web.

El funcionamiento de las páginas dinámicas se basa en la solicitud de realización de una operación en el lado del cliente, atender esa solicitud (esta acción se lleva a cabo en el lado del servidor) y enviar al cliente desde el servidor los datos resultantes de dicha operación.

El funcionamiento de las páginas web es básicamente el siguiente. Al introducir una dirección web lo que estamos haciendo es pedir un fichero localizado en un ordenador (que actuará de servidor). El servidor nos enviará este fichero y nuestro navegador (el programa cliente) se encargará de interpretarlo para que nos aparezca la página web (que será más o menos vistosa) en pantalla.

A partir de los diagramas obtenidos en la etapa de diseño y la información persistente que se almacena en la base de datos diseñada, implementaremos los scripts php que crearán las páginas Web del portal.

En general, cada módulo obtenido en la etapa de diseño está representado por un script php, y además se incluyen unos scripts que crean la conexión con la base de datos para el intercambio de la información entre ésta y los *scripts* php.

En casos concretos, hay scripts que implementan más de un módulo de los obtenidos en la fase de diseño (por ejemplo, los módulos de validación de los datos se incluyen junto a los módulos donde se tratan esos datos). Esto lo hacemos para facilitar la conectividad de la información que vamos a tratar, y evitar la creación de demasiados scripts de tamaño muy reducido que complicarían el entendimiento del entramado formado por los scripts que constituyen la aplicación web.

### Lenguaje SQL

Otro estándar, que lleva utilizándose más tiempo y de vital importancia para el desarrollo de aplicaciones cliente-servidor relacionadas con la base de datos, es el lenguaje estructurado de consulta SQL (*Structure Query Language*). Se desarrolló como estándar para la programación de bases de datos y para devolver los conjuntos de resultados a partir de ellas.

### Lenguaje de modelado UML

Es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema.

Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo.

El lenguaje de modelado pretende unificar la experiencia pasada sobre técnicas de modelado e incorporar las mejores prácticas actuales en un acercamiento estándar.

UML no es un lenguaje de programación. Las herramientas pueden ofrecer generadores de código de UML para una gran variedad de lenguaje de programación, así como construir modelos por ingeniería inversa a partir de programas existentes.

Es un lenguaje de propósito general para el modelado orientado a objetos. UML es también un lenguaje de modelado gráfico que permite una abstracción del sistema y sus componentes.

Existían diversos métodos y técnicas Orientadas a Objetos, con muchos aspectos en común pero utilizando distintas notaciones, se presentaban inconvenientes para el aprendizaje, aplicación, construcción y uso de herramientas, etc., además de pugnas entre enfoques, lo que generó la creación del UML como estándar para el modelado de sistemas de software principalmente, pero con posibilidades de ser aplicado a todo tipo de proyectos.

### Objetivos del UML

UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.

UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.

Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.

- ▶ Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- ▶ Imponer un estándar mundial
- ▶ Arquitectura del UML
- ▶ Arquitectura de cuatro capas, definida a fin de cumplir con la especificación *Meta Object Facility* del OMG.
- ▶ Meta-metamodelo: define el lenguaje para especificar metamodelos.
- ▶ Metamodelo: define el lenguaje para especificar modelos.

- Modelo: define el lenguaje para describir un dominio de información.
- Objetos de usuario: define un dominio de información específico.

## Lenguaje JavaScript

En el caso de que las páginas cliente requieran código JavaScript para la realización de alguna tarea en el lado del cliente deberán declararse funciones PHP que generen una o varias de esas funciones JavaScript. Un ejemplo son las funciones JavaScript de validación de formularios.

- Paso de información entre las páginas

El paso de información se realiza principalmente a través de los formularios, en los que tendremos una variable por cada campo del formulario, cuyo contenido será el contenido del campo.

También pueden incluirse campos ocultos que también serán enviados.

Otra forma de paso de la información es en la propia URL, menos recomendada para operaciones de inserción, modificación y borrado de datos ya que los valores enviados son visibles en la url. La notación utilizada para ello es la siguiente

“programa\_destino.php?variable\_1=valor&variable\_2=valor&...&variable\_N=valor”

## 6. CUESTIONES DE IMPLEMENTACIÓN RESEÑABLES

La aplicación desarrollada está condicionada por la arquitectura de ANDROID que puede ser observada en la Ilustración 13. Dentro de la arquitectura de ANDROID, la aplicación desarrollada estaría incluida en la sección “Applications”, junto al resto de aplicaciones del dispositivo. En un nivel inferior se situaría el framework de aplicaciones que contiene elementos que facilitan la gestión de recursos y el uso de las librerías del sistema por parte de la aplicación.

El siguiente componente es “*ANDROID RUNTIME*”, que está formado por:

- *CORE LIBRARIES*

Este elemento contiene librerías que incluyen casi la totalidad de la funcionalidad disponible en las librerías del lenguaje de programación Java.

- *DALVIK VIRTUAL MACHINE*

Al igual que en Java que utiliza las “*JAVA VIRTUAL MACHINES*”, las aplicaciones de la plataforma ANDROID se ejecutan en un contenedor independiente del resto de aplicaciones. Este contenedor no es otra cosa que una máquina virtual llamada *DALVIK*.



Ilustración 25. Arquitectura de ANDROID.

La última capa es la encargada de gestionar el hardware del dispositivo y permitir su uso a los niveles superiores.

## 6.1 CICLO DE VIDA DE LAS ACTIVITIES

En el framework de aplicaciones se dispone de un elemento llamado “Activity Manager”, este bloque es el encargado de controlar el ciclo de vida de las actividades y proporcionar un método común de navegación entre ellas.

Debido a la limitación de memoria por la que actualmente se caracterizan los dispositivos móviles, es necesario implementar un riguroso ciclo de vida que garantice al usuario poder ejecutar cualquier activity, sea cual sea el nivel de ocupación de memoria.

Para ello, ANDROID ha implementado el ciclo de vida de las actividades que se puede observar en la Ilustración 14.

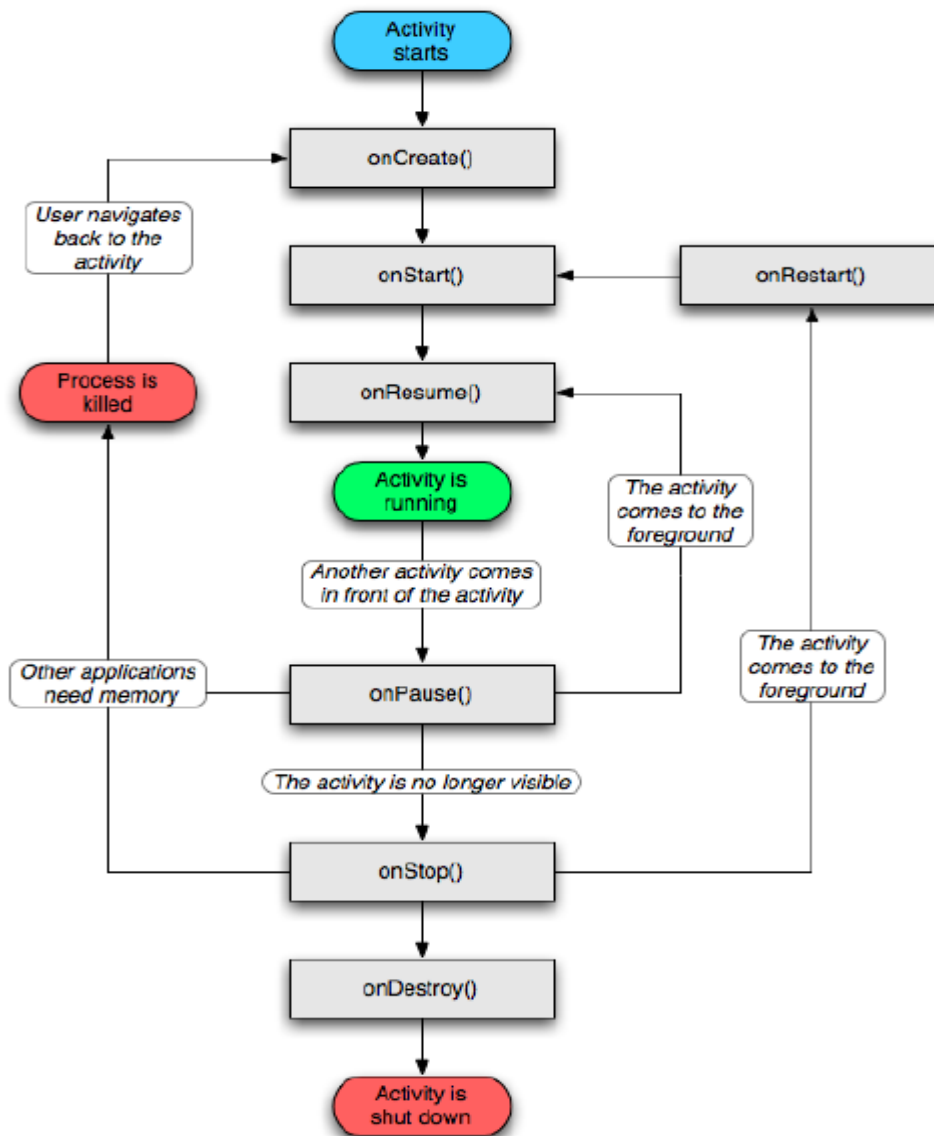


Ilustración 26. Ciclo de vida de las Activities.

Una activity, cuando va a dejar de estar visible, es decir, se va a superponer una nueva activity, pasa a un estado de pausa. En este estado se debe parar cualquier proceso que pueda consumir memoria o capacidad de proceso ya que es la antesala de la destrucción de la activity.

Desde este estado se puede pasar a:

- `OnResume` → Si la activity es invocada para volver a estar visible.
- `OnCreate` → Si la activity es destruida por el sistema por falta de recursos y vuelve a ser invocada por el usuario.
- `OnStop` → Si la activity deja de estar visible para el usuario (superposición de otra activity).

Al estado `OnStop` se llega cuando la activity deja de estar visible porque va a ser destruida por código (destrucción planificada por el desarrollador) o porque otra actividad ha sido reanudada y se superpone a la existente. Desde este estado se puede pasar a:

- `OnRestart` → Si la activity vuelve a estar visible para que el usuario pueda interactuar con ella.
- `OnDestroy` → Si se llama al método `finish()`.

El estado `OnDestroy` es el último estado por el que pasa una activity antes de su completa destrucción.

Para el caso concreto de la aplicación desarrollada, se ha hecho uso de algunos estados como `OnCreate`, `OnPause`, `OnResume` y `OnDestroy` para obtener el resultado deseado en la navegación entre las actividades y garantizar una apertura, reanudación y cierre correctos de la aplicación.

### 6.1.1 Guardar estado de una activity.

Como ya se ha visto, en ocasiones una *activity* puede ser eliminada por el sistema por falta de recursos, y en estos casos, la aplicación pierde el estado en el que se encontraba la *activity* antes de ser destruida y puede interesarnos que esto no ocurra. En ese caso es necesario guardar su estado para que cuando se reanude la *activity* pueda ser recargado su estado, y el usuario no note si la aplicación ha sido destruida o no (transparencia para el usuario).

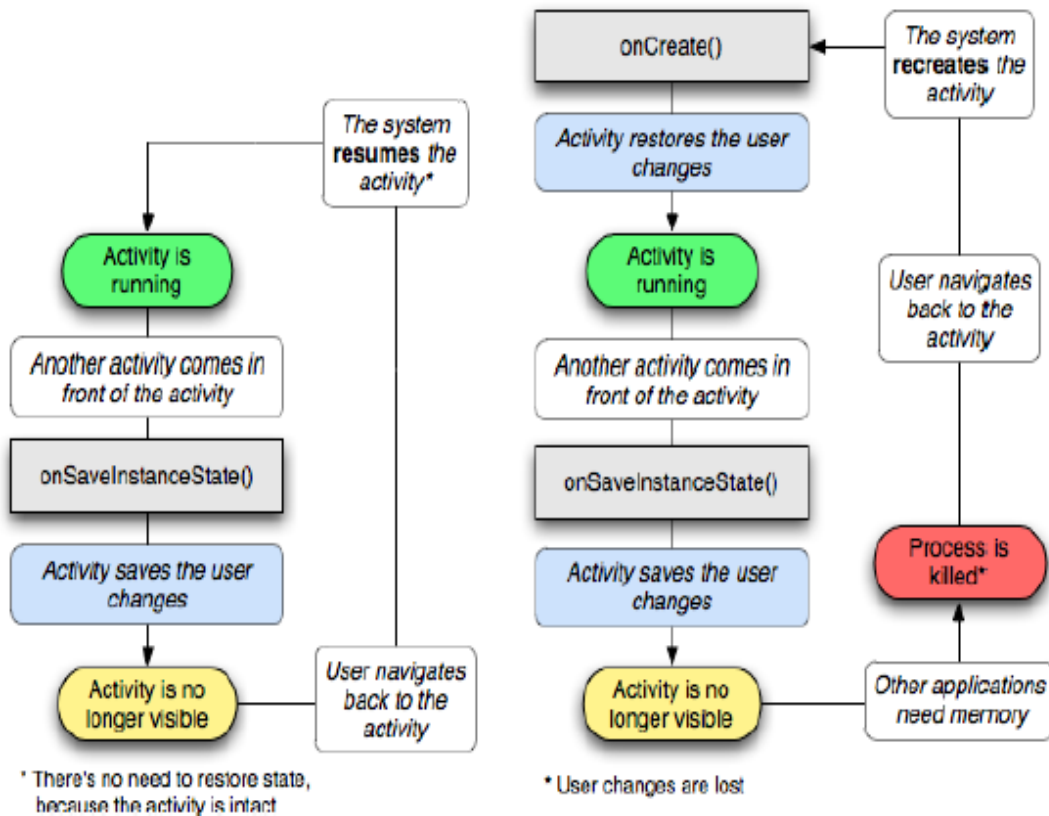


Ilustración 27. Guardar estado de una activity.

Como se observa en la Figura 20, existe un método llamado `onSaveInstanceState()` que es invocado cuando la actividad pasa al estado `onPause` y donde se puede guardar el estado para luego recuperarlo en el estado `onCreate`, cuando se vuelva a iniciar la *activity*.

En el caso concreto de este proyecto, no ha sido necesario almacenar el estado de la aplicación, puesto que los datos con los que trabaja (archivos y directorios) se obtienen en tiempo real en cada momento.



## 6.2 UTILIZACION DE “listviews”

Una vista *ListView* visualiza una lista deslizable verticalmente de varios elementos, donde cada elemento puede definirse como un *Layout*. Su utilización es algo compleja, pero muy potente.

El uso de un *ListView* conlleva los siguientes cuatro pasos:

- Diseñar un *Layout* que lo contenga al *ListView*

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical"
    android:background="#FF8C00">

    <ListView
        android:id="@android:id/list"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:divider="#b5b5b5"
        android:dividerHeight="1dp"
        android:cacheColorHint="#00000000" />

</LinearLayout>
```

• Diseñar un *Layout* individual que se repetirá en la lista

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:orientation="vertical">

    <!-- Album id / Hidden by default -->
    <TextView
        android:id="@+id/album_id"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:visibility="gone" />

    <!-- Album Name -->
    <TextView
        android:id="@+id/album_name"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:textSize="16dip"
        android:textColor="#000000"
        android:paddingTop="15dip"
        android:paddingBottom="15dip"
        android:paddingLeft="10dip"
        android:textStyle="bold" />

    <!-- Album Songs count -->
    <TextView android:id="@+id/songs_count"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_alignParentRight="true"
        android:textColor="#ffffff"
        android:textStyle="bold"
        android:background="#0000cd"
        android:paddingRight="3dip"
        android:paddingLeft="3dip" />

</RelativeLayout>
```

- Implementar una actividad que lo visualice el *Layout* con el *ListView*

```

package com.example.cv_uva;

import java.util.ArrayList;

public class AlbumsActivity extends ListActivity {

    // Connection detector
    ConnectionDetector cd;

    // Alert dialog manager
    AlertDialogManager alert = new AlertDialogManager();

    // Progress Dialog
    private ProgressDialog pDialog;

    // Creating JSON Parser object
    JSONParser jsonParser = new JSONParser();

    ArrayList<HashMap<String, String>> albumsList;

    // albums JSONArray
    JSONArray albums = null;

    // albums JSON url
    // private static final String URL_ALBUMS =
    // "http://api.androidhive.info/songs/albums.php";
    private static final String URL_ALBUMS = "http://10.0.2.2:8887/lis/albums.php";

    // ALL JSON node names
    private static final String TAG_ID = "id";
    private static final String TAG_NAME = "name";
    private static final String TAG_SONGS_COUNT = "songs_count";

    public void onCreate(Bundle savedInstanceState) {}

    // esto es para que actualice los numero de archivo por asig. con las
    // notificaciones
    BroadcastReceiver receiver = new BroadcastReceiver() {

        public void onReceive(Context context, Intent intent) {}

    };

    protected void onPause() {}

    protected void onResume() {}

    // hasta aqui

    /**
     * Background Async Task to Load all Albums by making http request
     * */
    class LoadAlbums extends AsyncTask<String, String, String> {}
}

```

- Personalizar cada una de los *Layouts* individuales según nuestros datos

## 6.3 NOTIFICACIONES

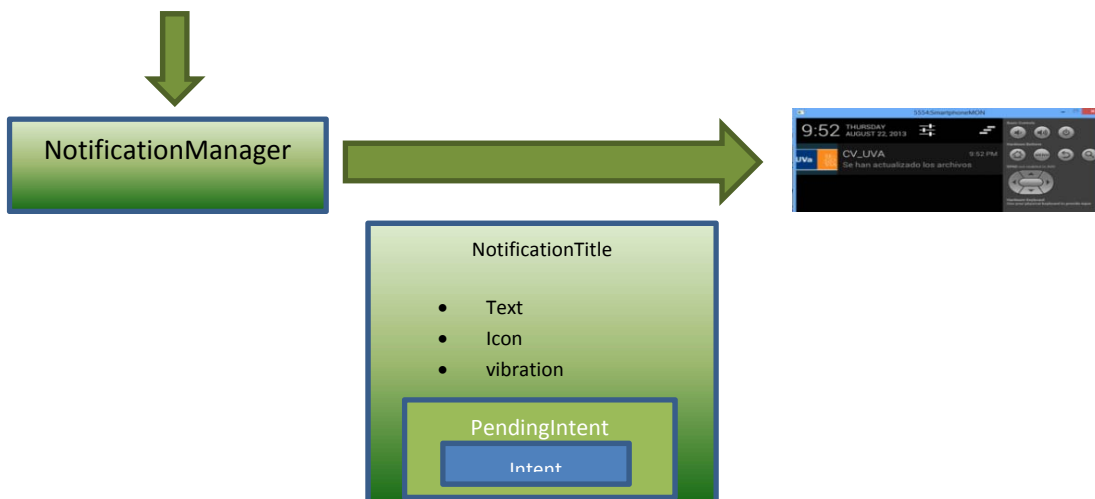
El sistema de notificaciones de Android es un mecanismo por el cual se le pueden enviar comunicaciones al usuario sin interrumpir la aplicación que esté ejecutando en ese momento. Las notificaciones son los componentes ideales para enviar mensajes al usuario desde servicios en segundo plano o desde los broadcast receivers. En nuestro caso, enviar un mensaje al usuario de CV\_UVA cuando se produce un cambio en la base de datos referente a los archivos PDF.

Las notificaciones se presentan a través de la barra de estado localizada en la parte superior de la pantalla y en la zona de notificaciones.

Para lanzar notificaciones disponemos de una serie de clases que simplificarán notablemente este proceso:

- **android.app.NotificationManager:** clase que gestiona las notificaciones. Se trata de un servicio del sistema, por lo tanto obtendremos una instancia de la misma a través del método `getSystemService(Context.NOTIFICACION_SERVICE)`.
- **android.app.Notification:** encapsula todo lo referente a la notificación que será lanzada a través del método `notify` de la clase `NotificationManager`. Alguno de sus parámetros de configuración son el título y texto de la notificación, así como el icono que la acompañara. También podremos establecer el sonido de aviso de la notificación o la vibración. No debemos olvidar que esta clase envolverá a la actividad que se lanzara al pulsar sobre la notificación.
- **android.app.PendingIntent:** contiene la descripción del intent que se va a lanzar al pulsar sobre la notificación.

En la siguiente se refleja cómo se relacionan entre sí las clases enumeradas anteriormente:



El presente trabajo fin de grado está dedicado a desarrollar una aplicación que permita visualizar en un dispositivo móvil (cliente) basado en Android, archivos PDF que se encuentren en un servidor. Así mismo, recibir notificaciones en el dispositivo cada vez que se añadan y/o modifiquen archivos en el servidor.

Este desarrollo está enfocado para usuarios que quieran consultar en cualquier lugar y a cualquier hora del día archivos que estén en un servidor. Por ejemplo, el caso que nos ocupa en este TFG, alumnos de la Universidad de Valladolid que quieran consultar los apuntes de sus asignaturas o cualquier otro archivo que el profesor haya subido al Campus Virtual. Aunque de igual forma puede ser empleado por trabajadores de una empresa que necesiten consultar archivos...

Para poder consultar estos archivos PDF, previamente han tenido que ser almacenados en el servidor por el profesor. Para ello, se ha desarrollado también una aplicación web, utilizando APACHE+PHP+MYSQL.

Esta última parte, cobra menos protagonismo, puesto que la instancia principal del TFG es la creación de la aplicación Android para dispositivos móviles. Aunque nadie duda de su necesidad, puesto que es el motor que hace poblar nuestra BBDD de archivos para poder ser visitados.

Al realizar el análisis de los requisitos se obtuvo el modelo de negocio al que se debe ajustar este desarrollo. Este modelo se analiza y explica mediante un conjunto de gráficos que se presentan a continuación.

Puesto que este TFG consta de dos aplicaciones claramente diferenciadas, en cuanto al lenguaje, entorno de ejercicios etc..

Esta sección la vamos a separar en dos partes, para poder realizar un análisis y diseño del sistema de cada aplicación lo más precisa posible. De esta forma (al no mezclarlas) se pretende ofrecer al lector de esta memoria una visión más concreta y sin confusión de cada una de las dos aplicaciones.

## **7. ANÁLISIS DEL SISTEMA**

Análisis del sistema es el proceso que, en reunión con el cliente, nos sirve para identificar las metas globales, analizar las perspectivas del cliente, sus necesidades y requisitos, realizar una planificación temporal y presupuestaria, y tratar otros puntos que puedan ayudar a la identificación y desarrollo del proyecto

El modelo de análisis es una abstracción resumida y precisa de lo que debe hacer el sistema deseado y no de la forma en que se hará. Gracias a que no incluye detalles de implementación, un buen modelo puede ser criticado por expertos de la aplicación que no sean programadores.

## 7.1 OBJETIVOS DEL SISTEMA

El objetivo principal de este TFG, es el desarrollo de la aplicación para dispositivos móviles basado en el sistema operativo Android, del Campus Virtual de la Universidad de Valladolid (Segovia).

Consecuencia de este objetivo tan general, se van a detallar, objetivos más concretos. Los cuales vamos a perseguir a lo largo de todo el proceso de desarrollo, para poder llegar a su consecución.

### APLICACIÓN WEB

<b>OBJ-1</b>	<b>Gestion de usuario</b>
<b>Descripción</b>	El sistema deberá gestionar la información relativa a los usuarios
<b>Subobjetivos</b>	OBJ-1.1 Gestión del profesor

<b>OBJ-1.1</b>	<b>Gestion del profesor</b>
<b>Descripción</b>	El sistema deberá gestionar la información relativa a los datos del usuario profesor.

<b>OBJ-2</b>	<b>Gestion de Archivos PDF</b>
<b>Descripción</b>	El sistema deberá gestionar la información relativa a los archivos PDF.
<b>Subobjetivos</b>	OBJ-1.1 Subir archivo PDF OBJ-1.2 Mostrar archivo PDF OBJ-1.3 Borrar archivo PDF

<b>OBJ-2.1</b>	<b>Subir Archivo PDF</b>
<b>Descripción</b>	El sistema deberá almacenar en la BBDD los archivos PDF que suba el profesor.

<b>OBJ-2.2</b>	<b>Mostrar Archivo PDF</b>
<b>Descripción</b>	El sistema deberá mostrar los archivos PDF que se encuentren almacenados en la BBDD.

<b>OBJ-2.3</b>	<b>Borrar Archivo PDF</b>
<b>Descripción</b>	El sistema deberá borrar los archivos PDF que se encuentren almacenados en la BBDD.

<b>OBJ-4</b>	<b>Ingerfaz</b>
<b>Descripción</b>	Diseño de una interfaz sencilla y atractiva adaptada a las particularidades del navegador.

## APLICACIÓN ANDROID

<b>OBJ-1</b>	<b>Gestion de usuario</b>
<b>Descripción</b>	El sistema deberá gestionar la información relativa a los usuarios
<b>Subobjetivos</b>	OBJ-1.1 Gestión de profesor OBJ-1.2 Gestión de alumno

<b>OBJ-1.1</b>	<b>Gestion del profesor</b>
<b>Descripción</b>	El sistema deberá gestionar la información relativa a los datos del usuario profesor.

<b>OBJ-1.2</b>	<b>Gestion de alumno</b>
<b>Descripción</b>	El sistema deberá gestionar la información relativa a los datos del usuario alumno.

<b>OBJ-2</b>	<b>Mostrar Archivo PDF</b>
<b>Descripción</b>	El sistema deberá mostrar los archivos PDF que se encuentren almacenados en la BBDD.

<b>OBJ-3</b>	<b>Envío de notificaciones</b>
<b>Descripción</b>	El sistema deberá ser capaz de mandar notificaciones con vibración cada vez que se realice una subida de archivos a la BBDD.

<b>OBJ-4</b>	<b>Ingerfaz</b>
<b>Descripción</b>	Diseño de una interfaz sencilla y atractiva adaptada a las particularidades de un dispositivo móvil (rotaciones de pantalla, tamaño, uso de las distintas resoluciones, etc)

## 7.2 CATÁLOGO DE REQUISITOS DEL SISTEMA

### 7.2.1 Requisitos de información

Los requisitos generales de nuestro sistema los podemos diferenciar en dos grupos:

- Mantener información acerca de archivos PDF.
- Mantener información acerca de los usuarios.

En las siguientes tablas, se muestra información más detallada sobre todos y cada uno de los requisitos de información.

### APLICACIÓN WEB

<b>IRQ-01</b>	<b>Información Usuario</b>
<b>Objetivo Asociado</b>	OBJ-1 Gestión de usuario OBJ-1.1 Gestión del profesor
<b>Requisitos asociados</b>	CU-001 Iniciar Sesión CU-002 Seleccionar Asignatura CU-003 Subir Archivo CU-004 Mostrar Archivo CU-005 Borrar Archivo
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente al usuario. En concreto
<b>Datos específicos</b>	dni pass nombre asig1 asig2 asig3 asig4 asig5 asig6 asig7 asig8 asig9 asig10 profesor
<b>Comentarios</b>	

<b>IRQ-03</b>	<b>Información Archivo</b>
<b>Objetivo Asociado</b>	OBJ-2 Gestión de Archivos PDF OBJ-2.1 Subir Archivo OBJ-2.2 Mostrar Archivo OBJ-2.3 Borrar Archivo
<b>Requisitos asociados</b>	CU-002 Seleccionar Asignatura CU-003 Subir Archivo CU-004 Borrar Archivo CU-005 Ver Archivo
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente al archivo. En concreto:
<b>Datos específicos</b>	id título archivo comentario
<b>Comentarios</b>	



## APLICACIÓN ANDROID

<b>IRQ-01</b>	<b>Información Usuario</b>
<b>Objetivo Asociado</b>	OBJ-1 Gestión de usuario OBJ-1.1 Gestión del profesor OBJ-1.2 Gestión del alumno
<b>Requisitos asociados</b>	CU-001 Iniciar Sesión CU-002 Seleccionar Asignatura CU-003 Seleccionar Archivo CU-004 Ver Archivo CU-005 Notificaciones
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente al usuario. En concreto
<b>Datos específicos</b>	dni pass nombre asig1 asig2 asig3 asig4 asig5 asig6 asig7 asig8 asig9 asig10 profesor
<b>Comentarios</b>	

<b>IRQ-03</b>	<b>Información Archivo</b>
<b>Objetivo Asociado</b>	OBJ-2 Mostrar Archivo OBJ-3 Enviar Notificaciones
<b>Requisitos asociados</b>	CU-003 Seleccionar Archivo CU-004 Ver Archivo CU-005 Notificaciones
<b>Descripción</b>	El sistema deberá almacenar la información correspondiente al archivo. En concreto:
<b>Datos específicos</b>	id título archivo comentario
<b>Comentarios</b>	

## 7.2.2 Requisitos no funcionales

El conjunto de requisitos no funcionales se divide en las siguientes categorías:

- Requisitos sobre factores relacionados con el usuario (Alta prioridad):
  - ✚ La formación del usuario genérico que interactúe con la aplicación será la básica de cualquier usuario de las aplicaciones Web habituales en la red Internet.
  - ✚ Existen varios tipos de usuarios y cada uno de estos tendrá acceso a unas pantallas y funciones.
- Requisitos referidos a los datos (Alta prioridad):
  - ✚ Se tomará como referencia para el desarrollo y explotación el SGBDs (Sistema Gestor de Base de Datos), MYSQL. Las aplicaciones deberán permitir configurar fácilmente los parámetros del SGBD y de la base de datos a utilizar. Se pretende que sólo haya que cambiar unos pocos datos para poder utilizar otro SGBD.
- Requisitos de funcionamiento (Prioridad media):
  - ✚ Se busca reducir la posibilidad de que el usuario cometa errores, para lo cual habrá una validación de todos los datos que el usuario introduce y las acciones que éste realice.
  - ✚ El tiempo de espera de cada acción debe de ser menor de 10 segundos para un acceso estándar a la red.
- Requisitos de capacidad de uso (Prioridad media):
  - ✚ La aplicación debe ser visible por la mayoría de los usuarios de la red Internet, por lo tanto debe al menos ser diseñada para MICROSOFT INTERNET EXPLORER 9, en la parte de la aplicación Android tiene que estar desarrollada en una versión del SO no muy actual para que puedan utilizarla usuarios con versiones anteriores.
- Requisitos de robustez, fiabilidad y seguridad (Prioridad media-alta):
  - ✚ El sistema se encontrará disponible 24x7 puesto que se trata de una aplicación Web/Dispositivos móviles.
  - ✚ Al menos, cada año se necesitará de una actualización para realizar actualizaciones en la base de datos y eliminar entradas no utilizadas.
  - ✚ La seguridad del sistema vendrá impuesta por la autenticación/autorización.

- Requisitos hardware (Prioridad media):
  - ▮ Los requisitos hardware para el servidor donde se mantenga el sistema serán los propios del sistema operativo, el servidor Web, el servidor Android y el soporte de la base de datos.
- Requisitos de la interfaz del usuario
  - ▮ El diseño de la interfaz de usuario tanto en la App Web como en la App Android debe ser sencilla, atractiva y adaptada a las particularidades del navegado o dispositivo móvil respectivamente.

### 7.2.3 Requisitos funcionales

Este proyecto solo tiene parte privada, esto es, que ningún usuario que no esté dado de alta en el sistema puede entrar, no hay nada que se pueda visitar si no es iniciando sesión.

Dentro de que es todo privado, si es interesante y que ya hemos mencionado en ocasiones anteriores, recordar que la parte de la aplicación Web solo puede ser usada por el usuario profesor.

#### 7.2.3.1 Identificación de Actores

El actor se refiere a cualquier persona que interactúe con el sistema. El usuario, es persistente durante el ciclo de duración del sistema y no un mero iniciador de éste. Él inicia y es el receptor de la salida de la aplicación, pero influye en ésta en diversos pasos intermedios.

#### App Web

Solo tiene acceso a esta parte de CV\_UVA, los usuarios que sean profesores. Estos serán los que suban/borren archivos PDF para que puedan ser mostrados tanto en la aplicación Web como en la aplicación Android.

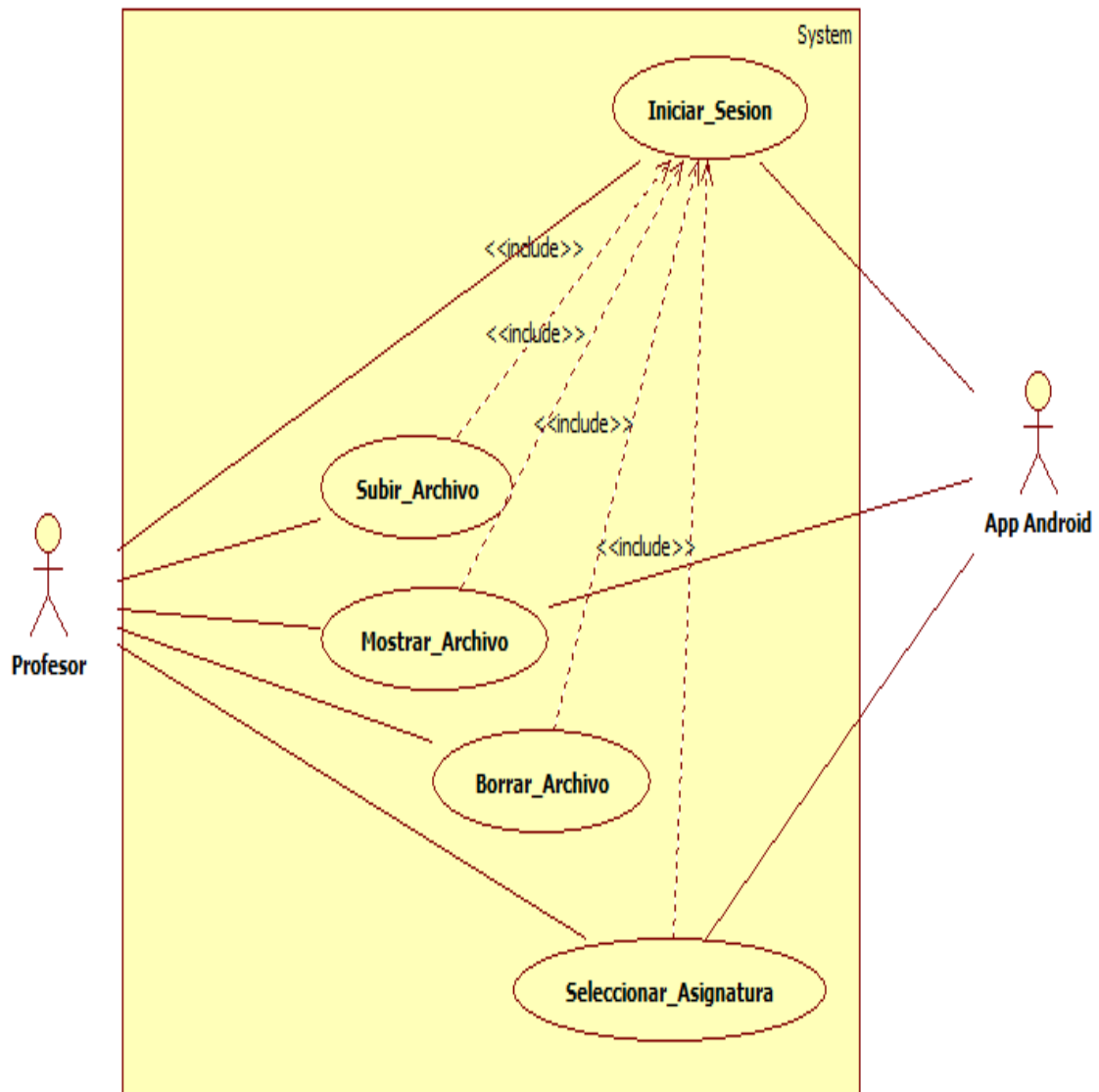
#### App Android

Al contrario que en la parte Web de la aplicación, en ésta, podrán acceder de forma indistinta tanto profesores como alumno. Pudiendo navegar por las asignaturas de cada uno, sus archivos y poder visualizar los archivos que previamente se subieron desde la parte Web de CV\_UVA.

### 7.2.3.2 Casos de uso

Los casos de uso, se van a mostrar según estemos hablando de la parte Web o Android de CV\_UVA.

## APLICACIÓN WEB



<b>CU-001</b>	<b>Iniciar Sesión</b>	
<b>Descripción</b>	El sistema deberá permitir al usuario identificarse como tal y poder acceder a las labores de gestión asociadas a cada categoría	
<b>Precondición</b>	El usuario debe ser profesor.	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor introduce su nombre de usuario y contraseña
	2	El sistema comprueba la validez de los datos
	3	Si los datos son correctos el sistema muestra la pantalla con las asignaturas propias de ese profesor
<b>Postcondición</b>	El usuario ha accedido al sistema	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3'	Si los datos introducidos son incorrectos, el sistema muestra un mensaje de error y vuelve al paso 1

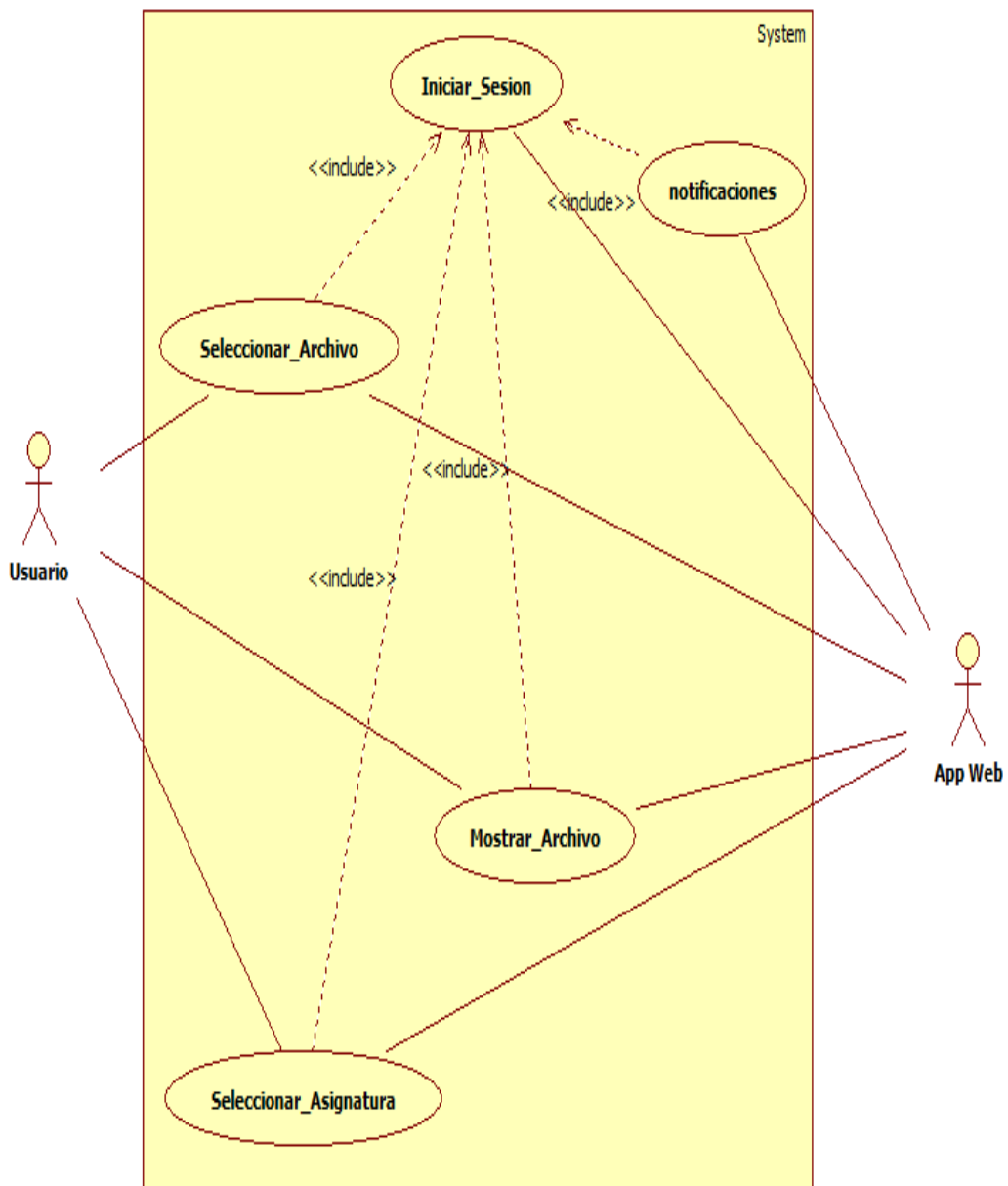
<b>CU-002</b>	<b>Seleccionar Asignatura</b>	
<b>Descripción</b>	El sistema permite al profesor seleccionar una de las asignaturas que imparte.	
<b>Precondición</b>	El usuario debe ser profesor	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor debe realizar el "CU-001 Iniciar Sesión"
	2	El sistema muestra la pantalla personal del profesor
	3	El profesor selecciona una asignatura de las que están en la lista.
	4	El sistema el sistema buscar los datos propios de la asignatura y abre una nueva pantalla con ellos.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

<b>CU-003</b>	<b>Subir Archivo</b>	
<b>Descripción</b>	El sistema debe permitir a los profesores subir nuevos archivos a una asignatura concreta	
<b>Precondición</b>	El usuario debe de ser profesor	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor debe realizar el “CU-001 Iniciar Sesión”
	2	El sistema muestra la pantalla personal del profesor
	3	El profesor rellena los campos existentes en el formulario de subida de archivos y pincha el botón de “Subir Archivo”
	4	El sistema comprueba que no tenga campos vacíos y que exista archivo para subir
	5	El sistema guarda el archivo en su correspondiente tabla de la base de datos
	6	El sistema muestra un mensaje de confirmación
	7	El administrador muestra su conformidad
	8	El profesor puede ver el archivo subido en la lista, actualizando la asignatura donde lo subió.
<b>Postcondición</b>	El nuevo archivo ha sido subido al servidor	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	4'	Si los datos introducidos son incorrectos, el sistema muestra un mensaje de error y vuelve al paso 3

<b>CU-004</b>	<b>Ver Archivo</b>	
<b>Descripción</b>	El sistema permite al profesor ver los archivos activos en ese momento, dentro de la asignatura seleccionada.	
<b>Precondición</b>	El usuario debe estar registrado en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor debe realizar el “CU-001 Iniciar Sesión”
	2	El sistema muestra la pantalla personal del profesor
	3	El profesor selecciona la asignatura donde está el archivo que quiere ver, mediante el “CU-002 Selecciona Asignatura”
	3	El sistema muestra los archivos activos para esa asignatura
	4	El profesor selecciona el archivo
	5	El sistema abre una nueva pestaña, visualizando de esta forma el archivo.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

<b>CU-005</b>	<b>Borrar Archivo</b>	
<b>Descripción</b>	El sistema permite al profesor borrar los archivos que quiera de una asignatura concreta	
<b>Precondición</b>	El usuario debe ser profesor	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor debe realizar el “CU-001 Iniciar Sesión”
	2	El sistema muestra la pantalla personal del profesor
	3	El administrador pulsa “Editar/Borrar Evento”
	4	El sistema muestra un mensaje de confirmación
	5	El administrador muestra su conformidad y vuelve al paso 2
<b>Postcondición</b>	El evento ha sido eliminado de la base de datos	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

APLICACIÓN ANDROID





<b>CU-001</b>	<b>Iniciar Sesión</b>	
<b>Descripción</b>	El sistema deberá permitir al usuario (profesor/alumno) identificarse como tal y poder acceder a sus asignaturas.	
<b>Precondición</b>	El usuario debe estar registrado en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor introduce su nombre de usuario y contraseña
	2	El sistema comprueba la validez de los datos
	3	Si los datos son correctos el sistema muestra la pantalla con las asignaturas propias de ese profesor
<b>Postcondición</b>	El usuario ha accedido al sistema	
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>
	3'	Si los datos introducidos son incorrectos, el sistema muestra un mensaje de error y vuelve al paso 1

<b>CU-002</b>	<b>Seleccionar Asignatura</b>	
<b>Descripción</b>	El sistema permite al profesor/alumno seleccionar una de las asignaturas que imparte.	
<b>Precondición</b>	El usuario debe estar registrado en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El usuario debe realizar el "CU-001 Iniciar Sesión"
	2	El sistema muestra la pantalla personal del usuario, con las asignaturas que cursa o imparte según el tipo de usuario que sea
	3	El usuario selecciona una asignatura de las que están en la lista.
	4	El sistema el sistema buscar los datos propios de la asignatura y abre una nueva pantalla con ellos.
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

<b>CU-003</b>	<b>Seleccionar Archivo</b>	
<b>Descripción</b>	El sistema permite al profesor seleccionar una de las asignaturas que imparte.	
<b>Precondición</b>	El usuario debe estar registrado en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El usuario realiza el “CU-002 Seleccionar Asignatura”
	2	El sistema muestra en la pantalla todos los archivos que contiene la asignatura previamente seleccionada
	3	El profesor selecciona un archivo de los que están en la lista.
	4	El sistema el sistema buscar los datos propios de la archivo y abre una nueva pantalla con ellos.
	5	El usuario pincha el botón “Ver Archivo”
	6	El sistema abre el archivo con la aplicación adecuada en función de la extensión del archivo
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

<b>CU-004</b>	<b>Ver Archivo</b>	
<b>Descripción</b>	El sistema permite al profesor ver los archivos activos en ese momento, dentro de la asignatura seleccionada.	
<b>Precondición</b>	El usuario debe estar registrado en el sistema	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El profesor debe realizar el “CU-003 Seleccionar Archivo”
	2	El usuario pincha el botón “Ver Archivo”
	3	El sistema abre el archivo con la aplicación adecuada en función de la extensión del archivo
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

<b>CU-005</b>	<b>Recibir notificación</b>	
<b>Descripción</b>	El sistema debe enviar notificaciones a los usuarios cuando se produzcan cambios en los archivos que estén en el sistema	
<b>Precondición</b>	El usuario ha debido de iniciar sesión anteriormente	
<b>Secuencia</b>	<b>Paso</b>	<b>Acción</b>
<b>Normal</b>	1	El sistema manda al usuario una notificación
	2	El usuario entra en CV_UVA para comprobar los cambios
	3	El sistema debe actualizar el número de archivos por asignatura y borrar el icono de notificación en la barra del dispositivo móvil
<b>Excepciones</b>	<b>Paso</b>	<b>Acción</b>

### 7.3 MODELO DE OBJETOS

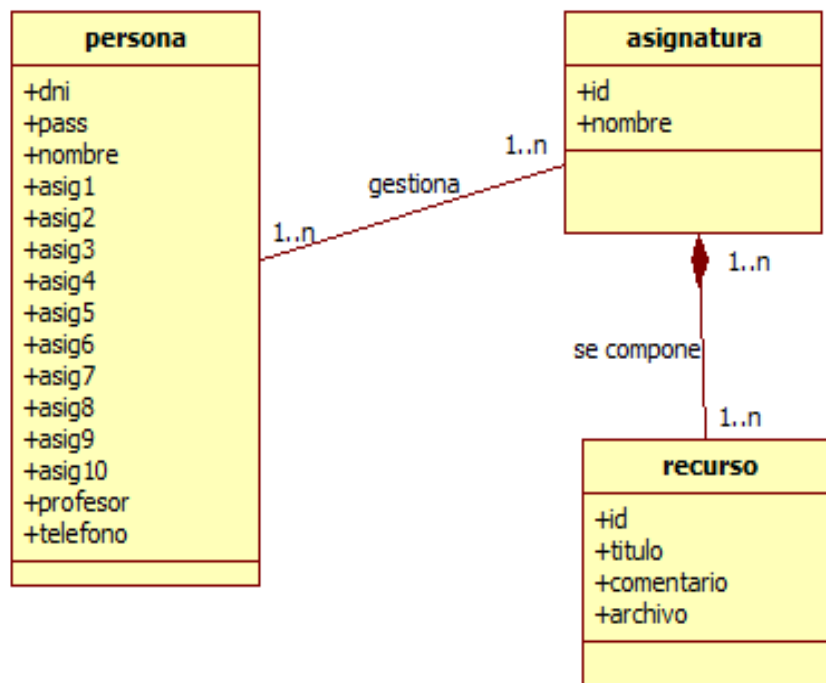
El Modelo de Objetos define qué atributos son asociados con cada objeto y cómo los objetos y los atributos pueden ser manipulados.

#### 7.3.1 Diagrama de clases de análisis

Un diagrama de clases es un tipo de diagrama estático que describe la estructura de un sistema mostrando sus clases, atributos y las relaciones entre ellos. Los diagramas de clases son utilizados durante el proceso de análisis y diseño de los sistemas, donde se crea el diseño conceptual de la información que se manejará en el sistema, y los componentes que se encargaran del funcionamiento y la relación entre uno y otro.

#### 7.3.2 Catálogo de clases de análisis

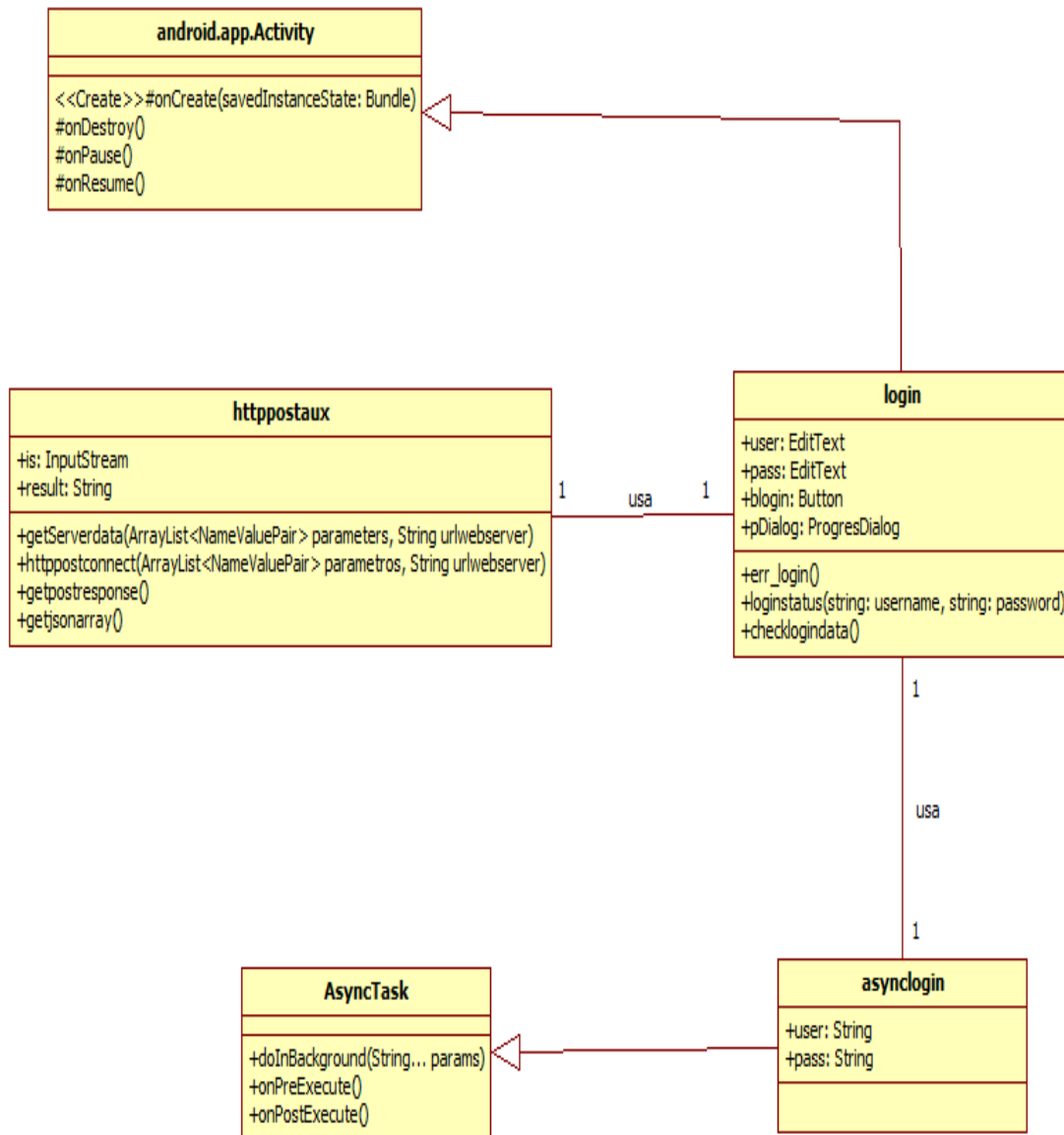
APLICACIÓN WEB



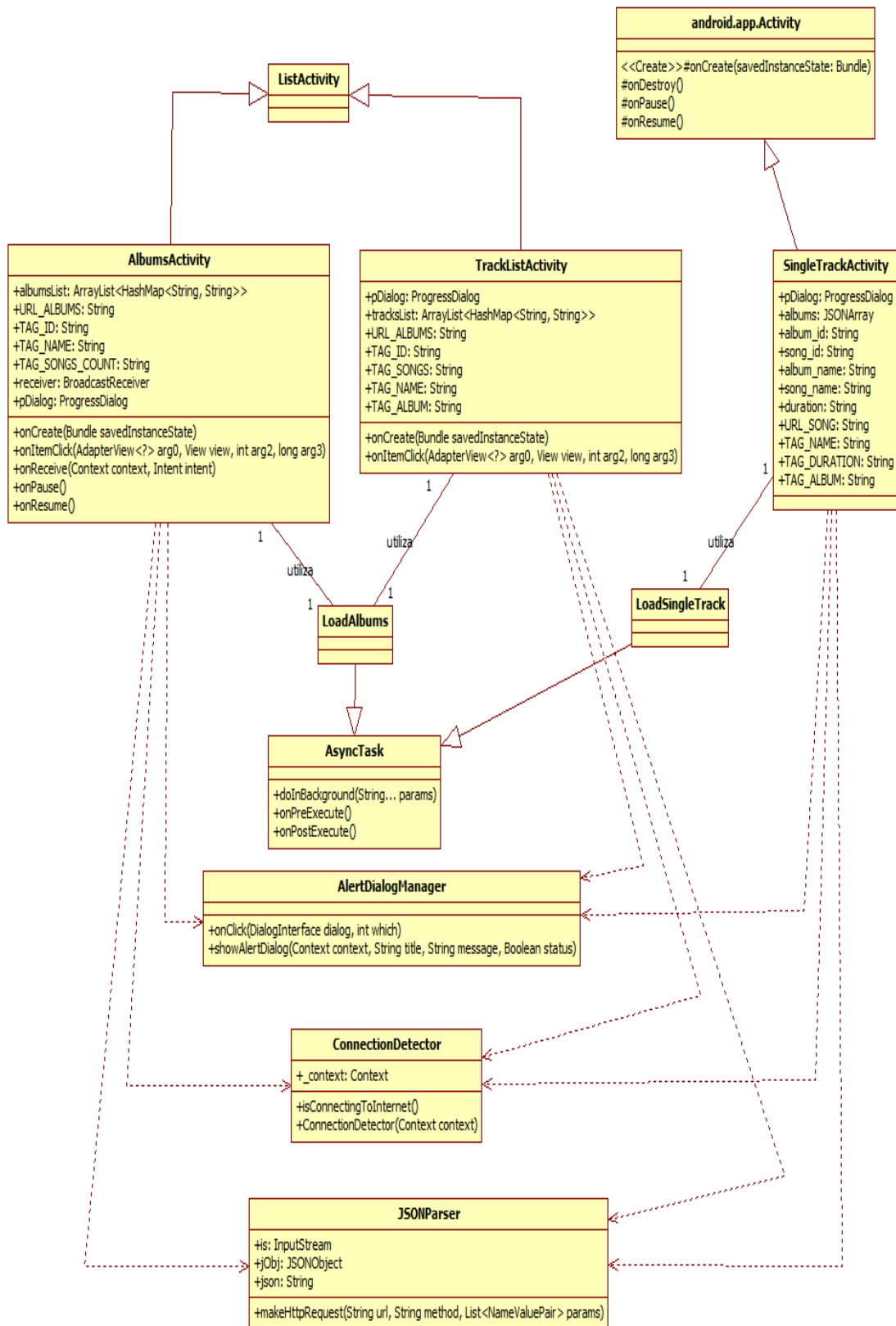
APLICACIÓN ANDROID

En este apartado se va a indicar el diseño de clases de la aplicación Android. Se ha creído oportuno, por el tamaño y detalle del diagrama, dividirlo en varias imágenes.

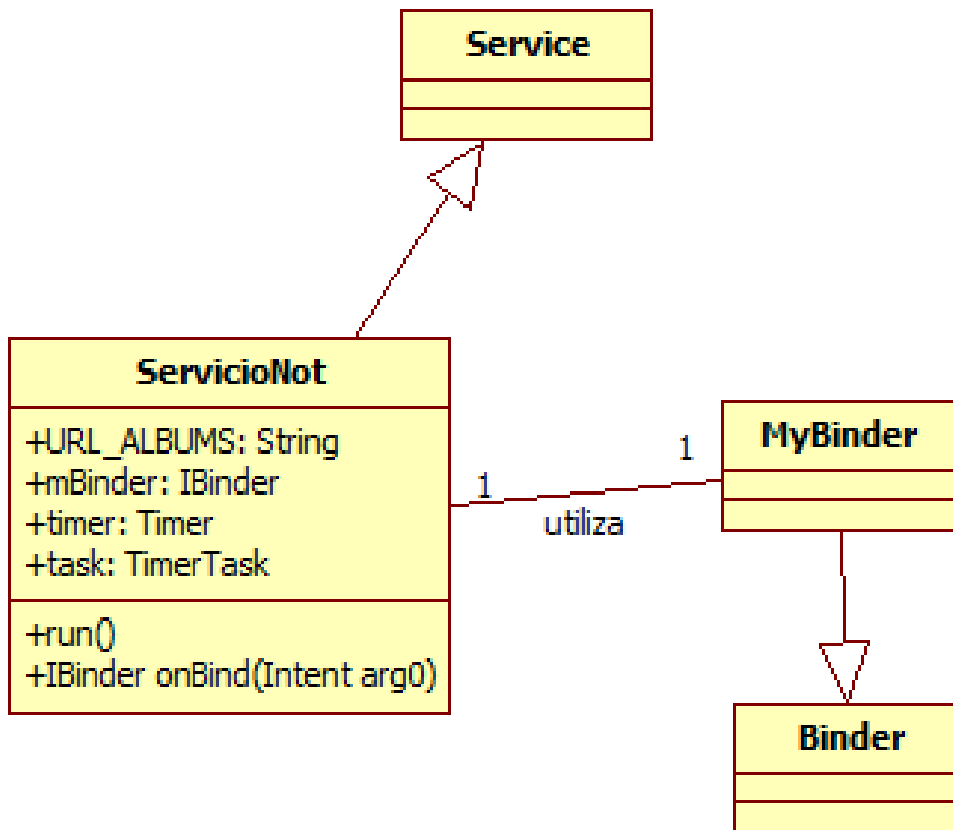
Login



● Crear listas (asignaturas/archivos)



Notificaciones



## 7.4 MODELO DINÁMICO

En cualquier sistema los objetos interactúan entre sí pasándose mensajes. Una interacción es un comportamiento que incluye un conjunto de mensajes intercambiados por un conjunto de objetos dentro de un contexto para lograr un propósito.

Las interacciones se utilizan para modelar aspectos dinámicos de las colaboraciones, que representan sociedades de objetos que juegan papeles específicos, y colaboran entre sí para llevar a cabo un comportamiento mayor que la suma de sus elementos.

Los principales diagramas que utiliza UML para la representación de las interacciones son los diagramas de secuencia.

En este apartado se va a definir el comportamiento más representativo del sistema. Este comportamiento está definido en los siguientes diagramas de secuencia y diagramas de estados.

### 7.4.1 Lista de escenarios

Vamos a definir un conjunto de escenarios con los que puede encontrarse nuestro sistema. Cada uno de estos escenarios estará asociado a un caso de uso, por lo que tendremos, al menos, tantos escenarios como casos de uso identificados.

Vamos a diferenciar los escenarios y casos de uso, entre los tres tipos de usuarios que tenemos en nuestras dos aplicaciones.



**APLICACIÓN WEB**

Asociado al Caso de Uso	Escenario
CU-001. Iniciar Sesión	El profesor inicia sesión en la aplicación con su identificador y contraseña.

Asociado al Caso de Uso	Escenario
CU-002. Seleccionar Asignatura	El profesor, una vez iniciada la sesión, selecciona una asignatura para poder ver los archivos que tiene, borrar alguno o subir nuevos archivos.

Asociado al Caso de Uso	Escenario
CU-003. Subir Archivo	El profesor, rellena los datos pertinentes en el formulario de subida de archivos

Asociado al Caso de Uso	Escenario
CU-004. Subir Archivo	El profesor, selecciona el archivo que quiere que se muestre en otra pantalla para poder verlo.

Asociado al Caso de Uso	Escenario
CU-005. Subir Archivo	El profesor, pincha sobre el botón “borrar” representado con una papela para poder borrar un archivo.

## APLICACIÓN ANDROID

Asociado al Caso de Uso	Escenario
CU-001. Iniciar Sesión	El profesor/alumno inicia sesión en la aplicación con su identificador y contraseña.

Asociado al Caso de Uso	Escenario
CU-002. Seleccionar Asignatura	El profesor/alumno, una vez iniciada la sesión, selecciona una asignatura para poder ver los archivos que tiene.

Asociado al Caso de Uso	Escenario
CU-002. Seleccionar Asignatura	El profesor/alumno, selecciona un archivo para poder ver los datos concretos de ese archivo.

Asociado al Caso de Uso	Escenario
CU-002. Seleccionar Asignatura	El profesor/alumno, pincha el botón “Ver Archivo” para poder visualizarlo en otra aplicación.

Asociado al Caso de Uso	Escenario
CU-002. Seleccionar Asignatura	El profesor/alumno, recibe notificaciones cada vez que algún archivo haya sido subido o borrado desde la aplicación web (por parte del profesor).

### 7.4.2 Diagramas de Secuencia

En un diagrama de secuencia destaca la ordenación temporal de los mensajes. Se construye colocando los objetos que participan en la interacción en la parte superior del diagrama. A continuación, se colocan los mensajes que los objetos envían y reciben a lo largo del tiempo.

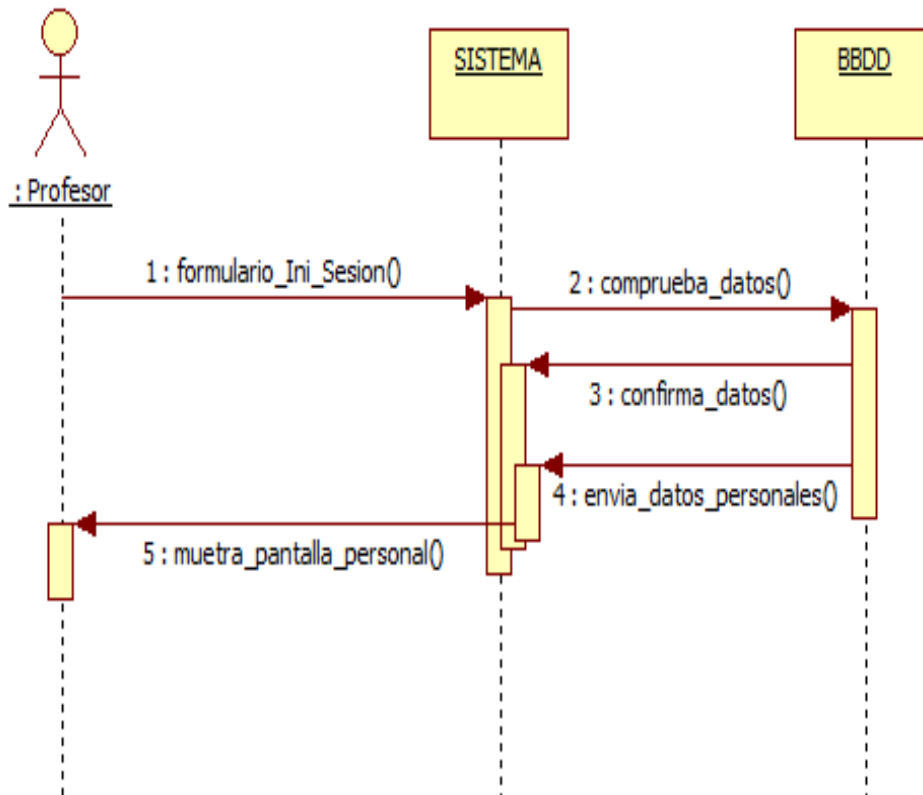
Los diagramas de secuencia asociados a los escenarios descritos anteriormente son:

### APLICACIÓN WEB

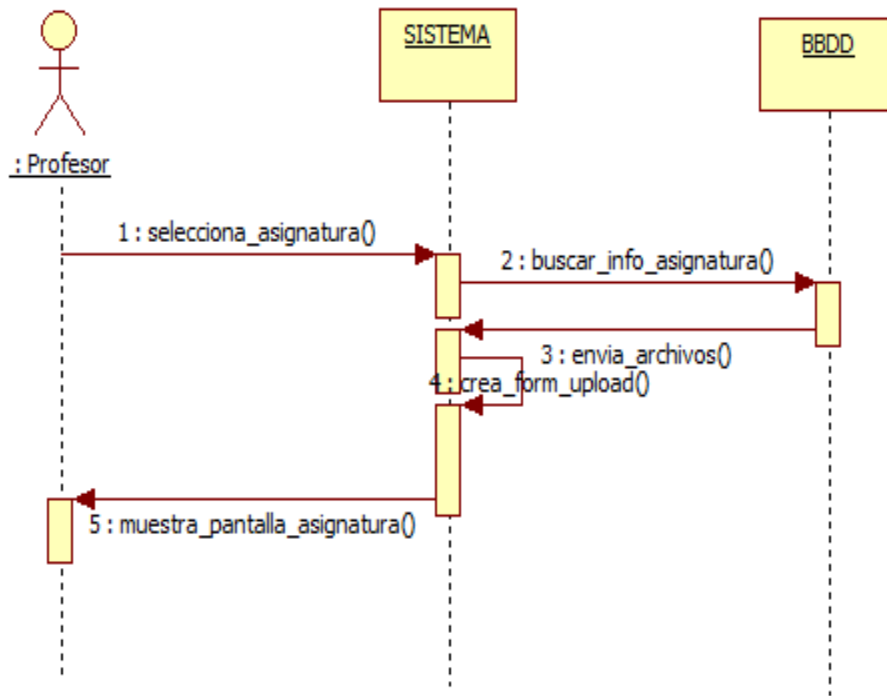
Puesto que hemos dicho a lo largo de toda esta memoria, que los únicos usuarios que pueden acceder a la aplicación Web, son los profesores, el actor que mostraremos será :Profesor.

Sin embargo, es importante decir, que en los casos de uso “CU-001: Iniciar Sesión”, “CU-002: Seleccionar Asignatura” y “CU-004: Ver Archivo” pueden ser invocados desde la Aplicación Android, pero, en vez de hacerlo a través de la interfaz gráfica de la Aplicación Web lo hará mediante una petición http de una página PHP.

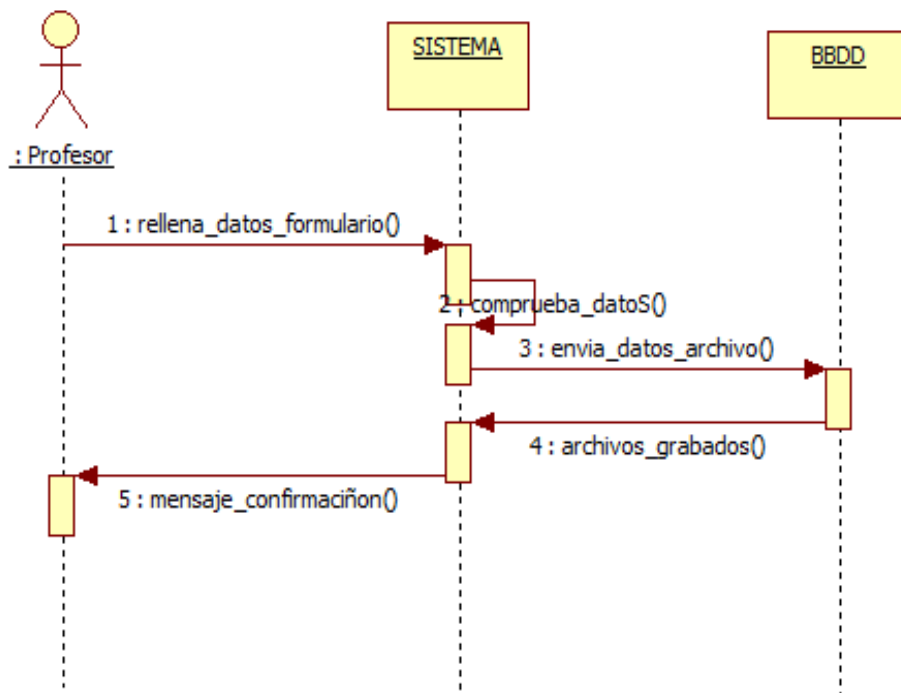
Iniciar Sesión



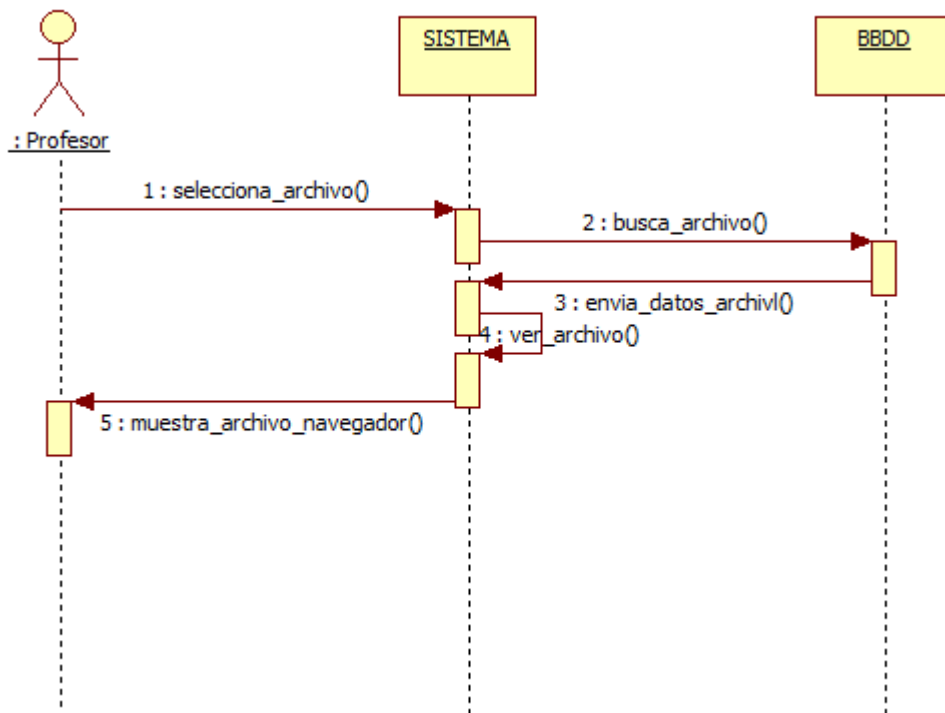
● Seleccionar asignatura



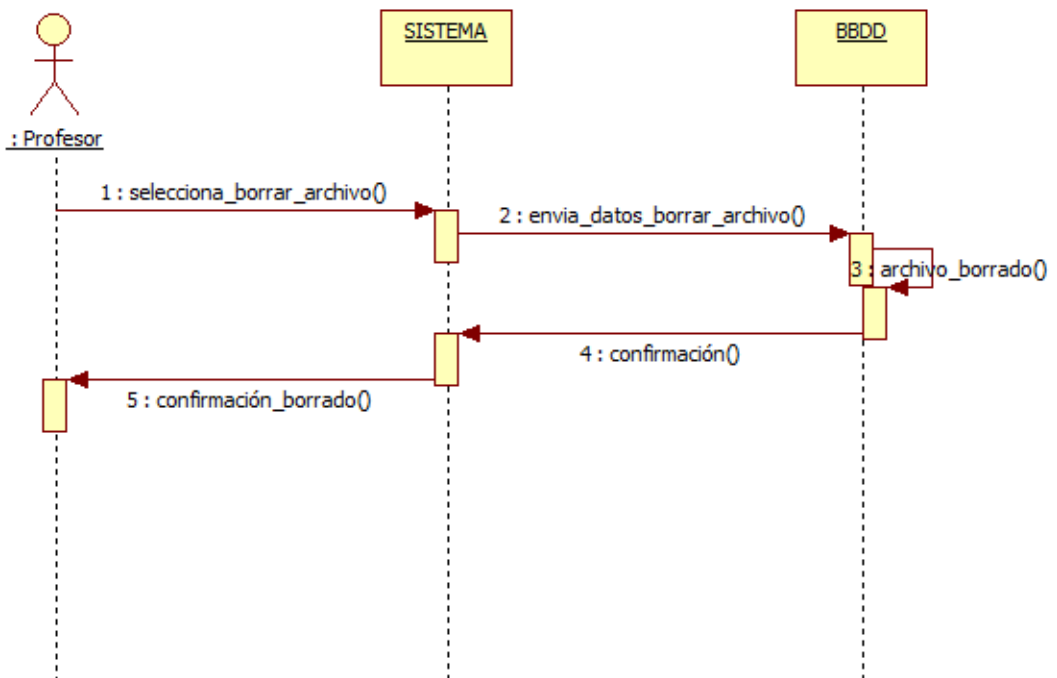
● Subir archivo



• Ver archivo



• Borrar archivo

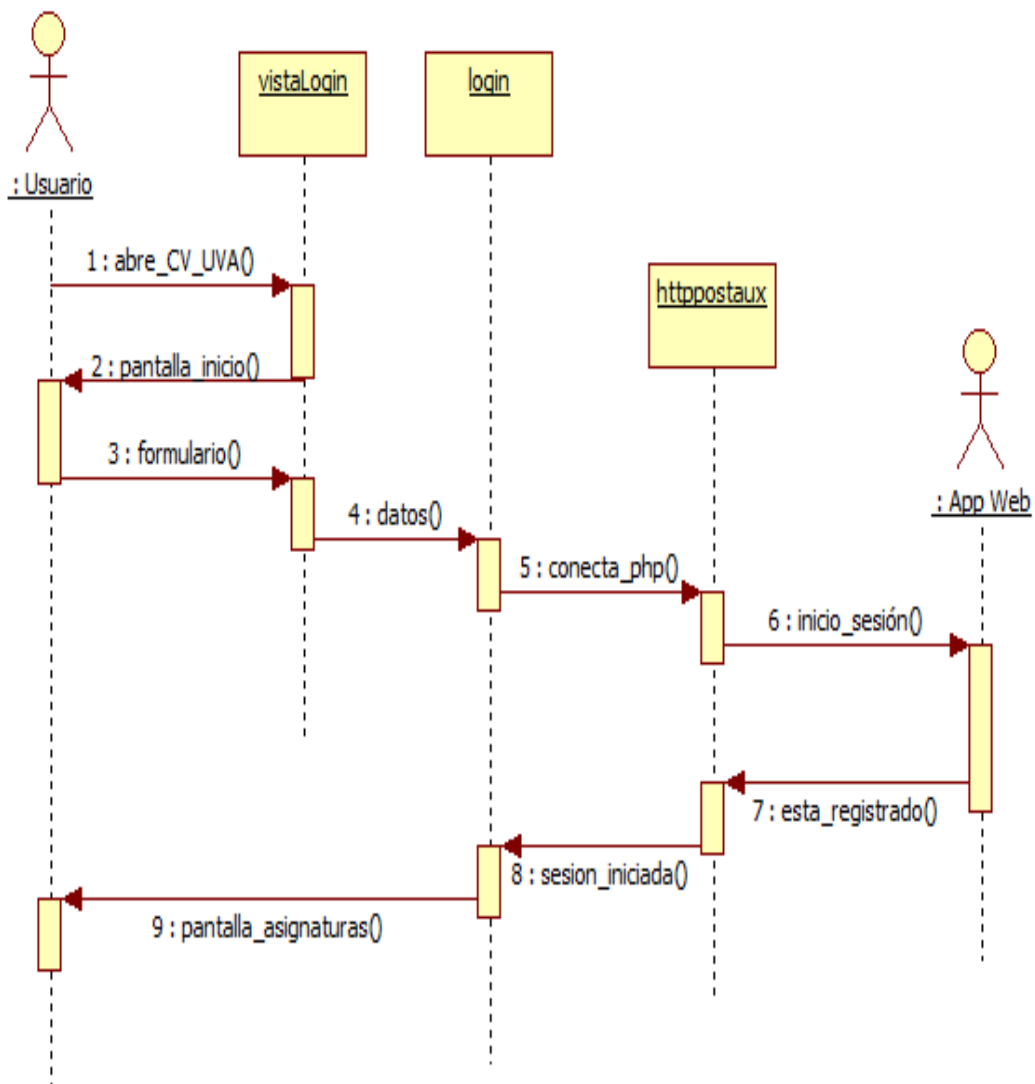


APLICACIÓN ANDROID

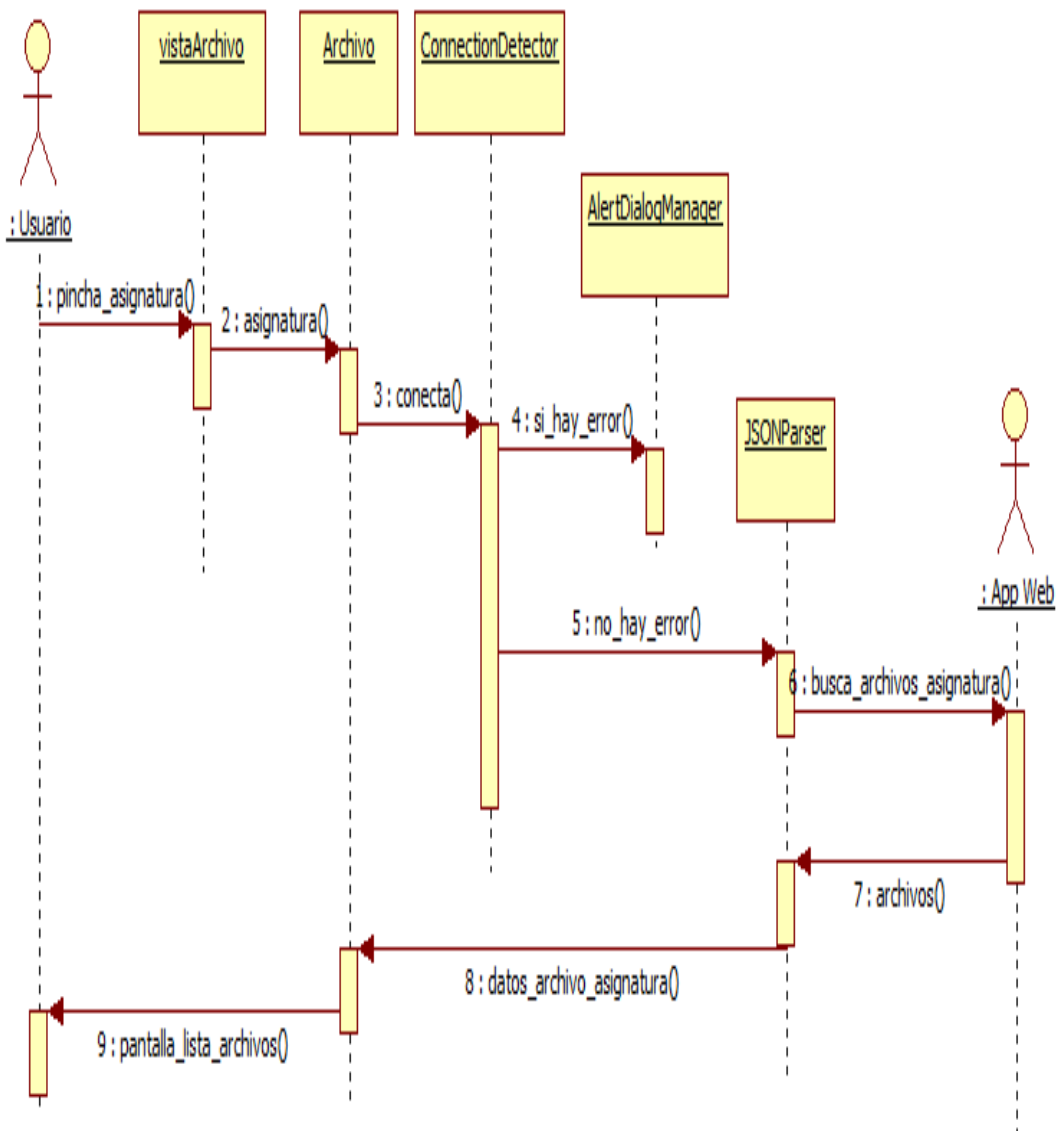
Al contrario que en la aplicación Web, en ésta (Android), pueden acceder cualquier usuario que esté registrado en el sistema, en nuestro caso alumno o profesores, por esto, y para no hacer diagramas repetitivos de secuencia (uno para alumno y otro para profesor), se ha optado por la opción de poner :Usuario como actor de los diagramas.

En estos diagramas de secuencia, podemos ver al final de cada uno de ellos, que interviene el actor App Web, esto es, que mediante una petición http invocaremos funcionalidades de la App Web.

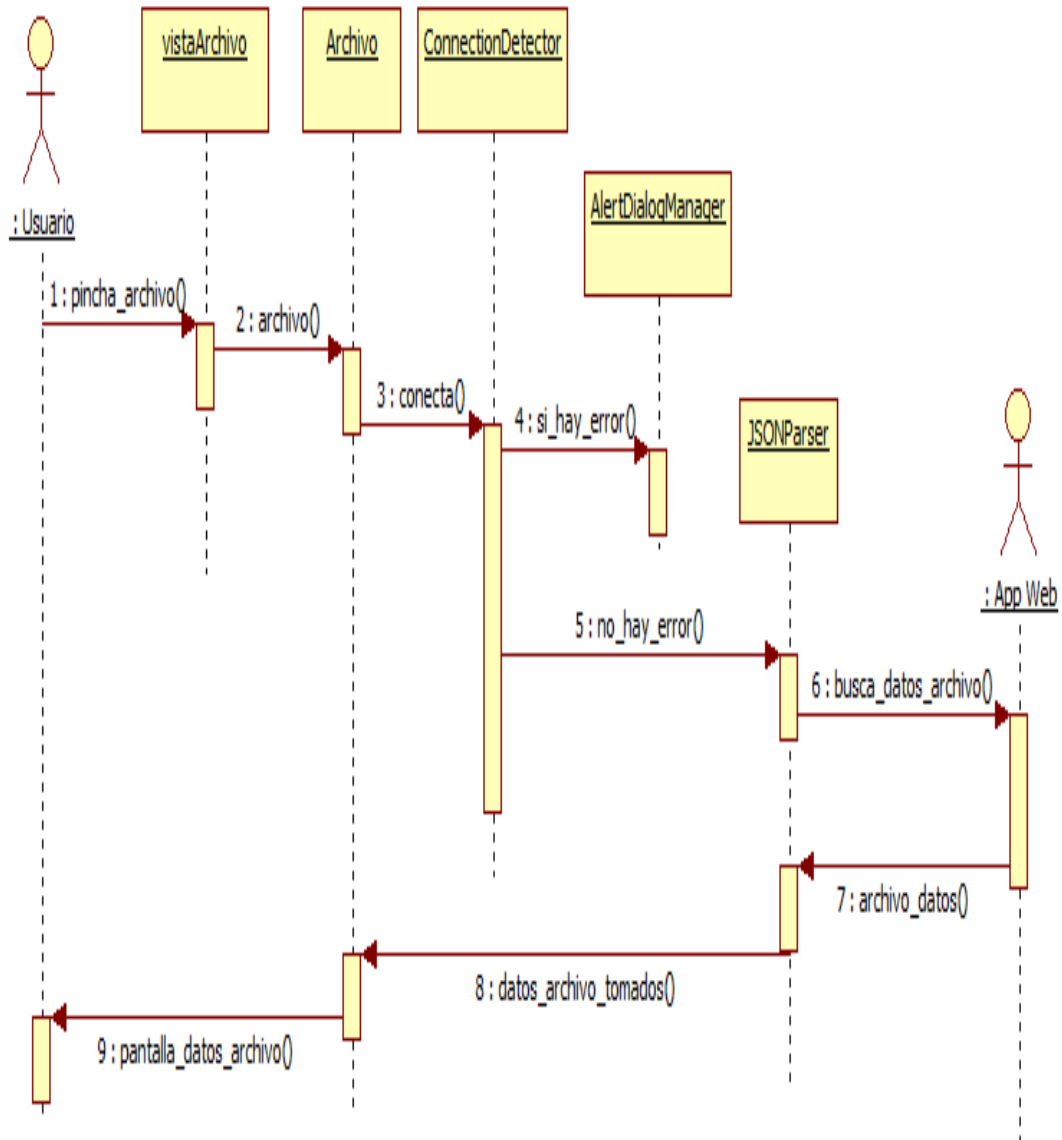
Iniciar Sesión



● Seleccionar asignatura

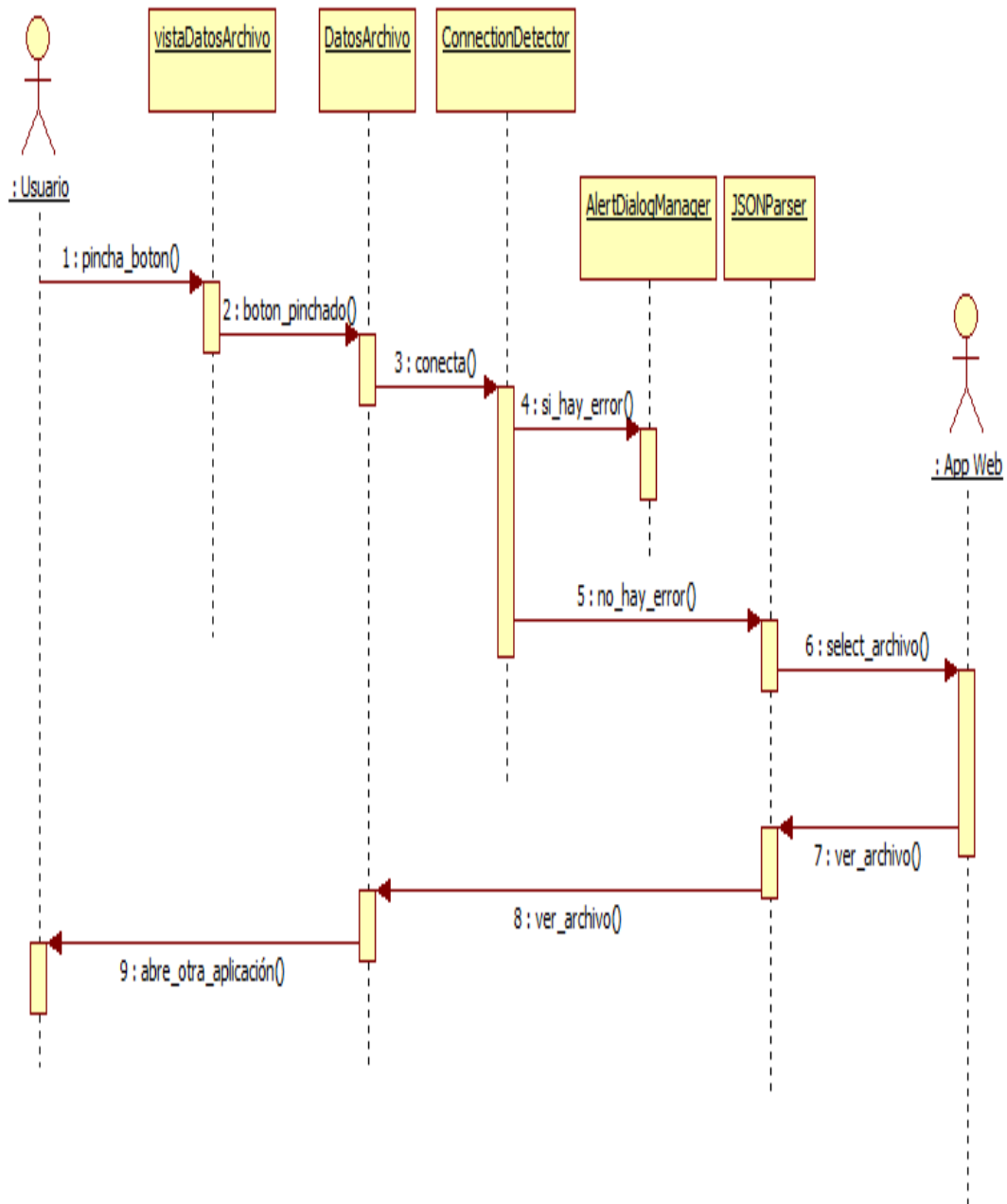


● Seleccionar archivo

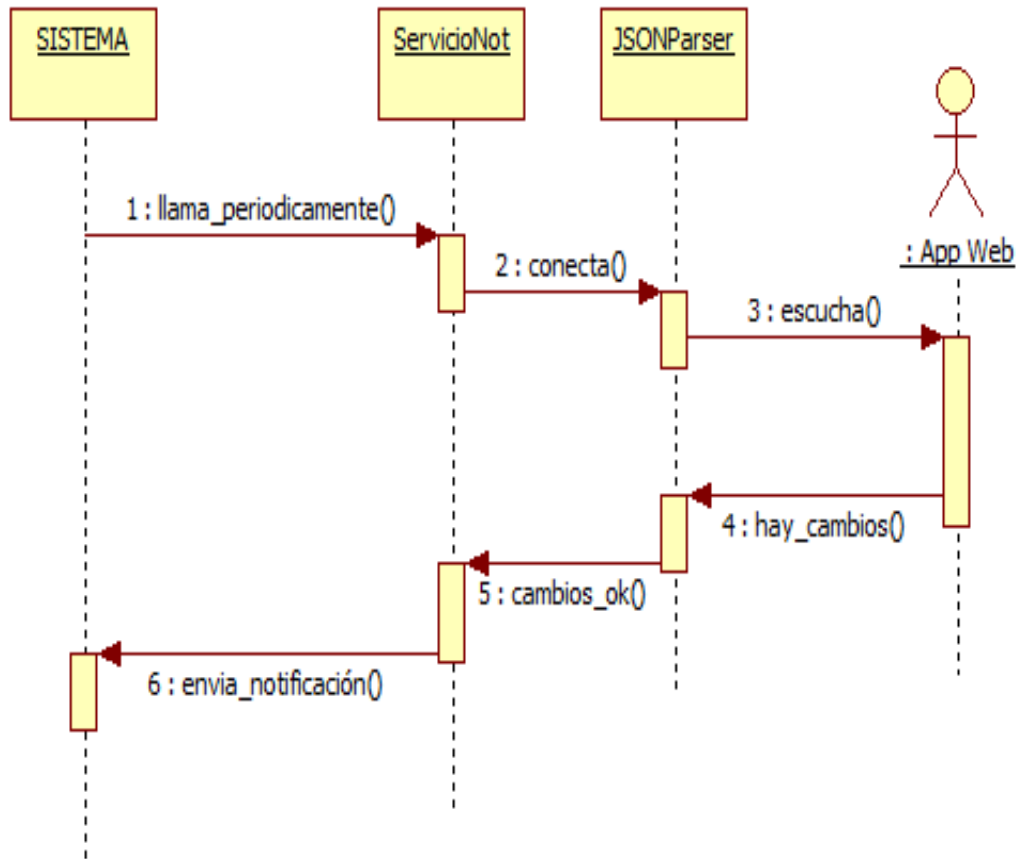




Ver archivo



Notificaciones

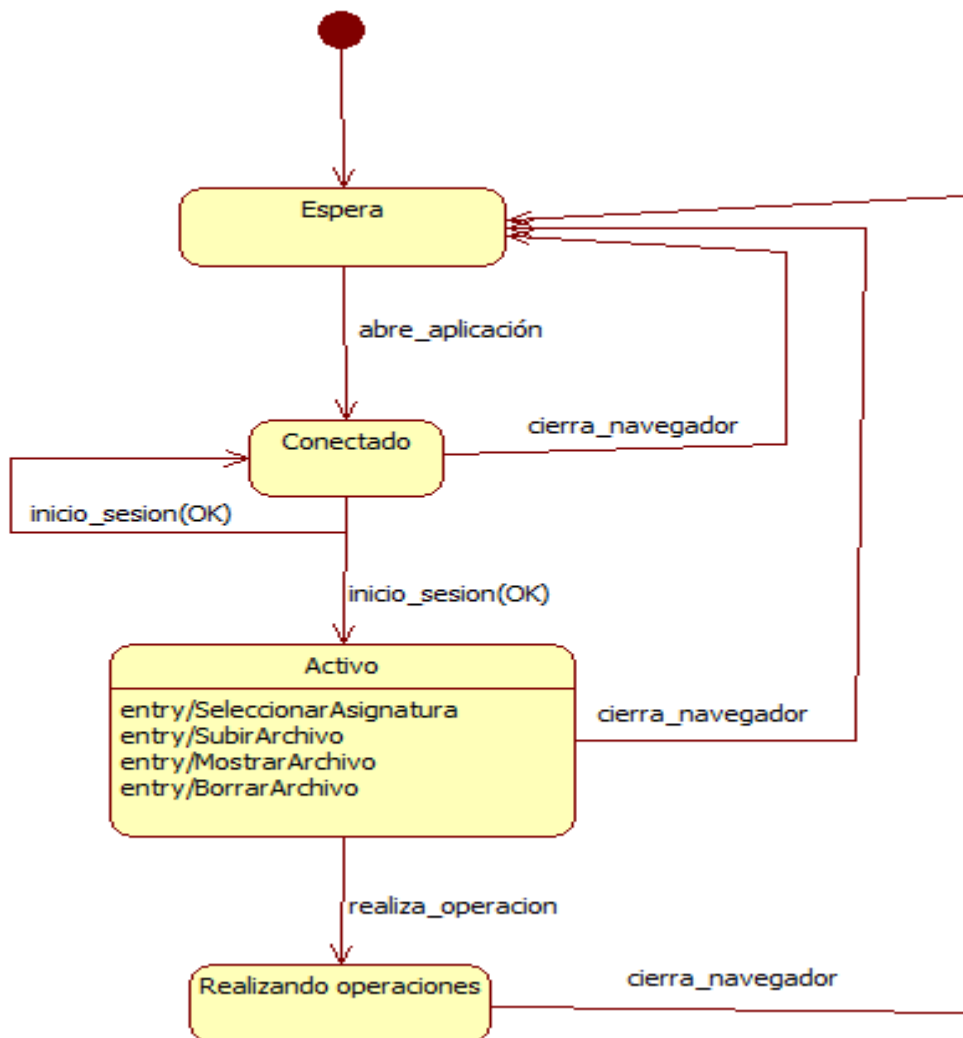


### 7.4.3 Diagrama de estados del sistema

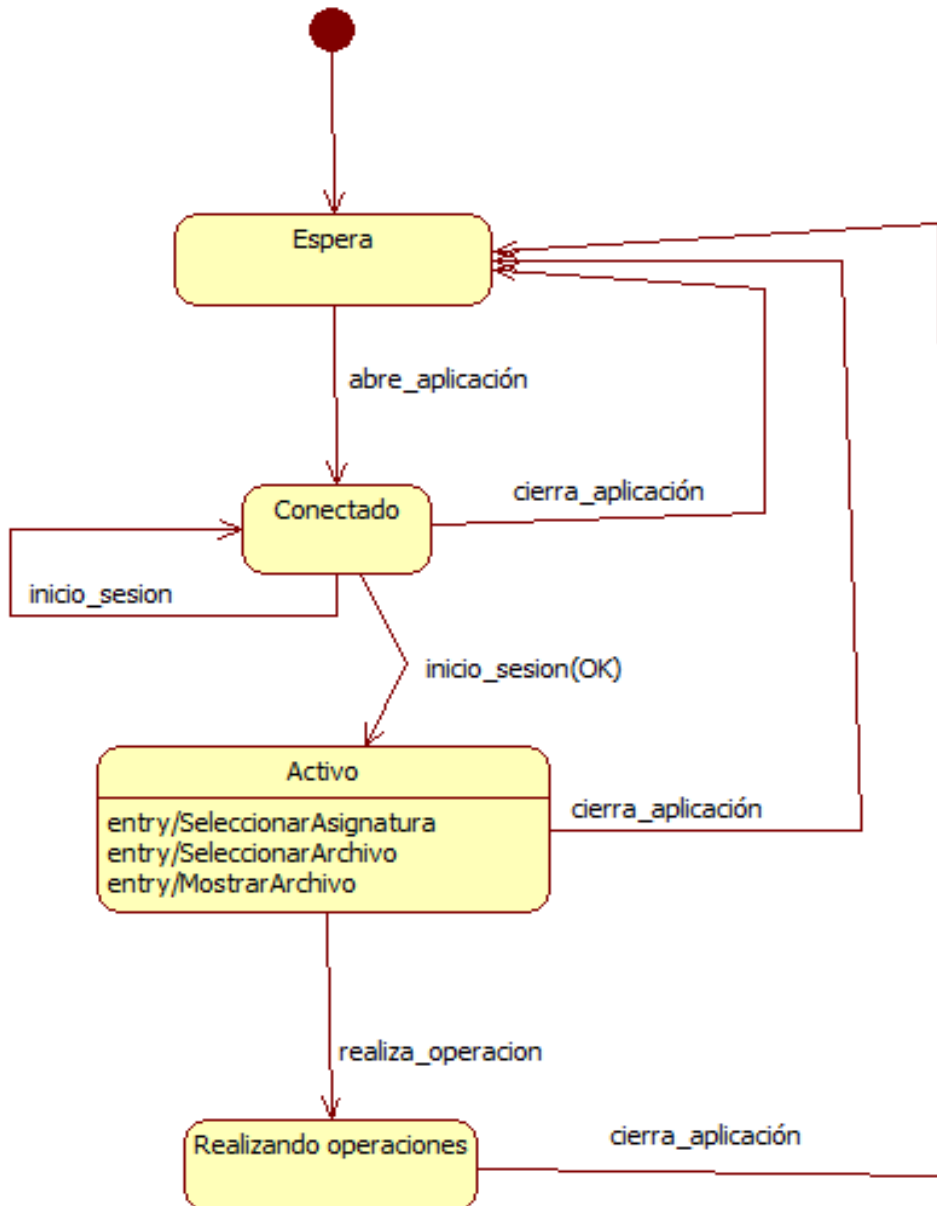
Para entender el funcionamiento del sistema, he considerado necesario incluir el diagrama de estados que, a mi juicio, es el más representativo.

Es un diagrama de estados general. En él, podemos observar los estados más generales del sistema y nos permite hacernos una idea de cómo funciona CV\_UVA a grandes rasgos.

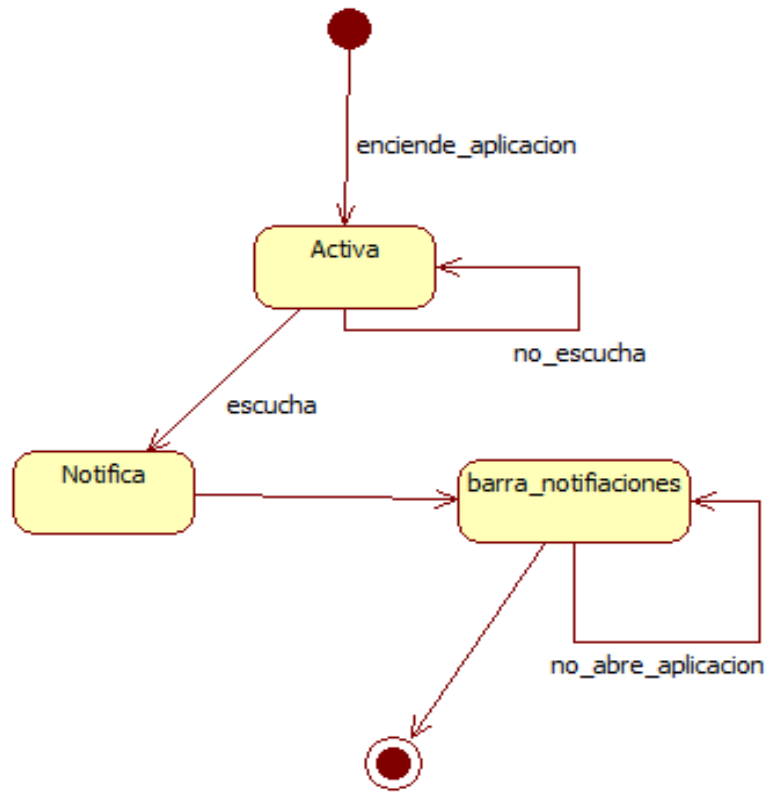
#### APLICACIÓN WEB



APLICACIÓN ANDROID



El siguiente diagrama muestra los estados en los que se puede encontrar una notificación.





## 8. DISEÑO DEL SISTEMA

Tras haber analizado el problema, comienza la fase de diseño. Mientras que en la fase de análisis el objetivo perseguido era qué ha de hacerse sin importar cómo hacerlo, ahora llega el momento de determinar cómo se resolverá el problema.

En las siguientes secciones se van a definir:

- Las clases que forman la solución al problema con sus respectivos métodos y atributos.
- Las interfaces de requeridas y usadas por los tipos de objeto.
- Las asociaciones y dependencias de las clases.
- La visibilidad entre clases.
- La división del sistema en paquetes (Subsistemas).
- El modelo de despliegue del sistema.

Existen dos posibilidades para persistir los objetos, una es el tratamiento por medio de ficheros y la otra es hacerlo a través de una base de datos. La opción elegida para este proyecto es una base de datos relacional.

Las razones para esta elección son que los gestores de bases de datos ya proporcionan:

- Sistema para evitar la redundancia e inconsistencias de los datos.
- Aislamiento de datos.
- Integridad.

El proyecto descrito en este documento seguirá la metodología orientada a objetos (aunque la parte Web no, se ha tomado así para todo su análisis). Esta metodología, guiada por casos de uso, requerirá el uso del lenguaje de modelado UML tanto para el modelado estático, como para el dinámico.

El desarrollo del software se realizará en:

- El lenguaje de programación orientado a objetos propio de la plataforma ANDROID, que en muchos casos es prácticamente idéntico a JAVA lo que hace posible reutilizar librerías de JAVA
- Lenguaje PHP con una base de datos MYSQL y servidor Apache para la parte de aplicación Web.

### **Tipos de arquitectura:**

En este TFG, al tener dos tipos de aplicación tan diferentes entre ellas, tenemos dos tipos de arquitecturas:

- La arquitectura usada será la condicionada por la plataforma ANDROID, la arquitectura Modelo-Vista-Controlador (MVC), que en el *framework* de ANDROID se divide en *layout* (vistas en XML), *activity* (controladores) y las clases java del modelo.
- La App Web está basada en una arquitectura cliente-servidor, modelo de aplicación distribuida en el que las tareas se reparten entre los proveedores de recursos o servicios, llamados servidores, y los demandantes, llamados clientes.

## **8.1 DISEÑO DE LA BASE DE DATOS**

En este apartado se describe el diseño de la base de datos mediante un modelo entidad-relación, un diccionario de datos y el correspondiente modelo relacional.

Debido a la poca cantidad de datos que es necesario almacenar, estos diagramas serán extremadamente simples.

### **8.1.1 Esquema conceptual**

El modelo entidad-relación es el modelo conceptual más utilizado para el diseño conceptual de bases de datos. Estos diagramas expresan entidades relevantes para un sistema de información, sus inter-relaciones y atributos.

Los elementos esenciales del modelo son las entidades, los atributos y las relaciones entre las entidades. Una entidad es un objeto que existe y que es distinguible de otros objetos.

Las entidades tienen atributos. Un atributo de una entidad es una característica interesante sobre ella, es decir, representa alguna propiedad que nos interesa almacenar.

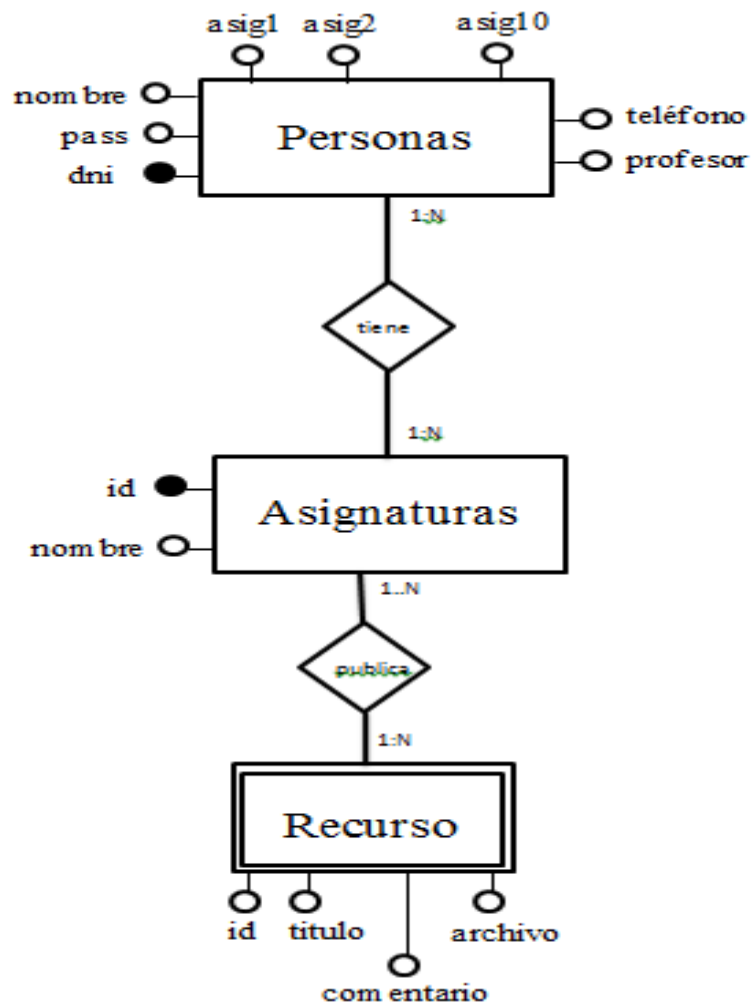
Una relación es una asociación entre entidades, generalmente dos.

En la siguiente imagen se muestra el diagrama de Entidad-Relación, que debería tener la BD de la aplicación CV\_UVA.



Y digo debería, porque como se podrá ver en el siguiente apartado (esquema relacional) no corresponde uno con el otro. Esto es, porque al inicio de la realización del presente TFG pensé que la mejor forma de implementar CV\_UVA es como se muestra en el esquema relacional y cuando termine la aplicación, y comencé a analizar todo de forma detallada, me di cuenta que no era la mejor forma de hacerlo pues limita mucho el poder ampliar esta aplicación sin tener que ampliar la BD.

De esta manera, y siendo consciente del error que supone el modelo relacional inicialmente planteado, se opta por incluir en este apartado este diagrama E/R, pues representa, de una forma clara y concisa, la semántica del modelo de datos que quiere soportarse en el correspondiente modelo relacional. Por falta de tiempo, no ha sido posible corregir el modelo relacional, inicialmente propuesto, en la implementación final.



## 8.1.2 Esquema relacional

Al diseñar una base de datos por un modelo relacional se tienen distintas alternativas, es decir, se pueden obtener diferentes esquemas relacionales a partir del modelo conceptual, los cuales no son todos semánticamente equivalentes y unos se ajustan a la realidad más que otros.

El modelo relacional representa la forma en que se basan la mayor parte de los SGBD comerciales que existen a día de hoy. Este modelo se basa en el concepto matemático de relación, que gráficamente se representa mediante una tabla. Todos los datos están estructurados a un nivel lógico como tablas formadas por filas y columnas.

**Personas** (dni, pass, nombre, asig1, asig2, asig3, asig4, asig5, asig6, asig7, asig8, asig9, asig10, profesor, teléfono)

**Asignatura** (id, nombre)

**Recurso1** (id, título, archivo, comentario)

**Recurso2** (id, título, archivo, comentario)

**Recurso3** (id, título, archivo, comentario)

**Recurso4** (id, título, archivo, comentario)

**Recurso5** (id, título, archivo, comentario)

**Recurso6** (id, título, archivo, comentario)

**Recurso7** (id, título, archivo, comentario)

**Recurso8** (id, título, archivo, comentario)

**Recurso9** (id, título, archivo, comentario)

**Recurso10** (id, título, archivo, comentario)

### 8.1.3 Esquema interno

```
-- Base de datos: `android`  
-----
```

```
--  
--  
Estructura de tabla para la tabla `asignaturas`  
--  
  
CREATE TABLE IF NOT EXISTS `asignaturas` (  
  `id` int(3) NOT NULL,  
  `nombre` varchar(30) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
-----
```

```
--  
--  
Estructura de tabla para la tabla `personas`  
--  
  
CREATE TABLE IF NOT EXISTS `personas` (  
  `dni` varchar(10) NOT NULL,  
  `pass` varchar(10) NOT NULL,  
  `nombre` varchar(20) NOT NULL,  
  `asig1` int(3) NOT NULL,  
  `asig2` int(3) NOT NULL,  
  `asig3` int(3) NOT NULL,  
  `asig4` int(3) NOT NULL,  
  `asig5` int(3) NOT NULL,  
  `asig6` int(3) NOT NULL,  
  `asig7` int(3) NOT NULL,  
  `asig8` int(3) NOT NULL,  
  `asig9` int(3) NOT NULL,  
  `asig10` int(3) NOT NULL,  
  `profesor` int(1) NOT NULL,  
  `telefono` varchar(10) NOT NULL,  
  `activo` varchar(2) NOT NULL,  
  PRIMARY KEY (`dni`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1;  
-----
```

```
--
```

```
--
```

```
Estructura de tabla para la tabla `tabla1`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tabla1` (  
  `id` int(4) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=14 ;
```

```
-----
```

```
--
```

```
-- Estructura de tabla para la tabla `tabla2`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tabla2` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
-----
```

```
--
```

```
--
```

```
Estructura de tabla para la tabla `tabla3`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tabla3` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
-----
```

```
--
```

```
--
```

```
Estructura de tabla para la tabla `tabla4`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tabla4` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
-----
```

```
--
```

```
--
```

```
Estructura de tabla para la tabla `tabla5`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tabla5` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;
```

```
-----
```

```
--
```

```
--
```

```
Estructura de tabla para la tabla `tabla6`
```

```
--
```

```
CREATE TABLE IF NOT EXISTS `tabla6` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;
```

```
-----
```

```
--  
--  
Estructura de tabla para la tabla `tabla7`  
--  
CREATE TABLE IF NOT EXISTS `tabla7` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
-----
```

```
--  
--  
Estructura de tabla para la tabla `tabla8`  
--  
CREATE TABLE IF NOT EXISTS `tabla8` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=4 ;  
-----
```

```
--  
--  
Estructura de tabla para la tabla `tabla9`  
--  
  
CREATE TABLE IF NOT EXISTS `tabla9` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=1 ;  
-----
```

```
--  
--  
Estructura de tabla para la tabla `tabla10`  
--  
  
CREATE TABLE IF NOT EXISTS `tabla10` (  
  `id` int(3) NOT NULL AUTO_INCREMENT,  
  `titulo` varchar(50) NOT NULL,  
  `archivo` longblob NOT NULL,  
  `comentario` varchar(200) NOT NULL,  
  PRIMARY KEY (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 AUTO_INCREMENT=5 ;  
-----
```





## 9. DISEÑO DE LAS PRUEBAS DEL SISTEMA

Hemos utilizado el modelo tradicional de caja negra, basándonos en el modelo de casos de uso visto en el análisis.

### 9.1 Pruebas de instalación

#### APLICACIÓN WEB

##### Prueba ordenador 1

Características: Intel Core i5 2.20 GHz con 4 Gb RAM  
Sistema Operativo: Windows 7 Ultimate  
Servidor: Apache  
Navegador: Internet Explorer 9

Resultado de la instalación de la base de datos: Correcto.  
Resultado de la instalación de la aplicación en el directorio www: Correcto

##### Prueba ordenador 2

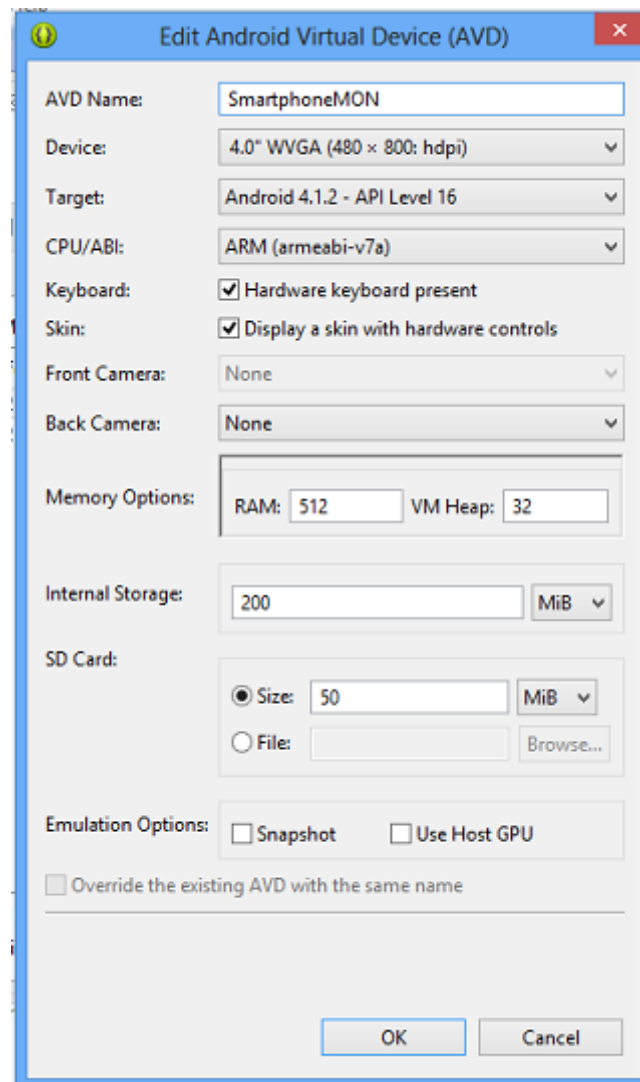
Características: AMD Turion™ X2 Ultra Dual-Core 2.40 GHz con 4 Gb RAM  
Sistema Operativo: Windows 7 Ultimate  
Servidor: Apache  
Navegador: Internet Explorer 9

Resultado de la instalación de la base de datos: Correcto.  
Resultado de la instalación de la aplicación en el directorio www: Correcto.

## APLICACIÓN ANDROID

Puesto que yo no tengo un móvil con Android como SO, se han tenido que hacer todas las pruebas con un AVD (*Andriod Virtual Device*), a lo que hemos llamado varias veces en esta memoria emulador.

Las características del emulador, son las reflejadas en la siguiente imagen:



## 9.2 Pruebas del sistema

### APLICACIÓN WEB

Caso de Prueba	
Objetivo	Comprobar que la pantalla inicial de la aplicación se carga correctamente.
Descripción	Al introducir la dirección donde reside la aplicación ésta se cargará correctamente.
Entrada	Introducimos la dirección en un navegador.
Salida	La pantalla de la aplicación.
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que acepta un usuario y clave válidos
Descripción	Al introducir un identificador y una clave de usuario y pulsar aceptar el sistema debe permitir el acceso al usuario
Entrada	Eventos de ratón o teclado
Salida	Aparece la pantalla de bienvenida y le da paso a su entorno
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que Cerrar Sesión correctamente
Descripción	Al pinchar el icono de la universidad, volvemos a la página principal de la aplicación
Entrada	Eventos de ratón o teclado
Salida	Aparece la pantalla principal
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que rechaza un usuario y clave no validos
Descripción	Al introducir un identificador y una clave de usuario erróneo y pulsar aceptar el sistema debe rechazar el acceso al usuario
Entrada	Eventos de ratón o teclado
Salida	El sistema muestra un error y vuelve a la página principal de la aplicación
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que muestra la pantalla personal usuario profesor que inicio sesión, con las asignaturas que imparte
Descripción	Una vez iniciada sesión se muestra la pantalla personal del profesor
Entrada	Evento de ratón
Salida	Se muestra la pantalla personal del profesor
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que muestra los datos de una asignatura
Descripción	Pinchando sobre alguna de las asignaturas, se muestra los datos, esto son los archivos que contiene.
Entrada	Evento de ratón
Salida	Se muestra por pantalla los archivos perteneciente a la asignatura seleccionada
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que muestra el archivo
Descripción	Pinchando sobre algún de las archivo, se abre otra ventana en el navegador y muestra el archivo seleccionado
Entrada	Evento de ratón
Salida	Se abre otra pantalla con el archivo seleccionado
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que se sube un archivo
Descripción	Rellenando correctamente los datos del formulario de subida, el archivo se graba correctamente en la BD
Entrada	Evento de teclado y ratón
Salida	Muestra un mensaje de confirmación
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que se borra un archivo
Descripción	Pinchando sobre el icono de borrar en un archivo concreto, éste se borra de la BD
Entrada	Evento de ratón
Salida	Muestra un mensaje de confirmación de borrado
Resultado	Se obtiene el resultado esperado.

## APLICACIÓN ANDROID

Caso de Prueba	
Objetivo	Comprobar que la pantalla inicial de la aplicación se carga correctamente.
Descripción	Al introducir la dirección donde reside la aplicación ésta se cargará correctamente.
Entrada	Tocamos sobre el icono de la aplicación
Salida	La pantalla inicial de la aplicación.
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que acepta un usuario y clave válidos
Descripción	Al introducir un identificador y una clave de usuario y pulsar aceptar el sistema debe permitir el acceso al usuario
Entrada	Evento táctil
Salida	Aparece la pantalla de bienvenida y le da paso a su entorno
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que mantiene la sesión abierta correctamente
Descripción	Al salir a la pantalla principal sigue la sesión abierta con el usuario que la inicio
Entrada	Evento táctil
Salida	Mantiene la sesión abierta
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que rechaza un usuario y clave no validos
Descripción	Al introducir un identificador y una clave de usuario erróneo y pulsar aceptar el sistema debe rechazar el acceso al usuario
Entrada	Evento táctil
Salida	El sistema muestra un error y vuelve a la página principal de la aplicación
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que muestra las asignaturas de usuario que inicio sesión
Descripción	Una vez iniciada sesión se muestra la pantalla con todas las asignaturas de ese usuario
Entrada	Evento de táctil
Salida	Se muestra la pantalla de las asignaturas
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que muestra los archivos de la asignatura que hemos seleccionado
Descripción	Cuando tocamos sobre una asignatura se muestran los archivos que perteneces a esa asignatura
Entrada	Evento de táctil
Salida	Se muestra la pantalla de los archivos
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que muestra los datos del archivos que hemos seleccionado
Descripción	Cuando tocamos sobre un archivo se muestran los datos concreto de ese archivo
Entrada	Evento de táctil
Salida	Se muestra la pantalla de los datos del archivo
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que funciona el botón “Ver Archivo”
Descripción	Cuando tocamos sobre el botón “Ver Archivo” se abre el archivo seleccionado en la aplicación que le corresponda en función de la extensión que tenga dicho archivo
Entrada	Evento de táctil
Salida	Se abre el archivo en otra aplicación
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que envía notificaciones
Descripción	Cuando subimos o borramos un archivo desde la aplicación web, se muestra una notificación en la barra de notificaciones de la aplicación Android.
Entrada	Evento de ratón
Salida	Sale el icono de la aplicación en la barra de notificaciones de la aplicación Android
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que se quita el icono de notificación
Descripción	Cuando recibimos una notificación y abrimos de nuevo la aplicación, se borra el icono de la barra de notificaciones
Entrada	Evento de táctil
Salida	El icono se borra de la barra de notificaciones
Resultado	Se obtiene el resultado esperado.

Caso de Prueba	
Objetivo	Comprobar que se actualizan los archivos después de la notificación
Descripción	Cuando recibimos una notificación y abrimos de nuevo la aplicación, se actualizan los archivos de la aplicación, esto es, que cambia el número de archivos por asignatura.
Entrada	Evento de táctil
Salida	El número de archivos por asignatura se actualiza
Resultado	Se obtiene el resultado esperado.

## 10. INTRODUCCIÓN

En este documento se va a describir el funcionamiento de la aplicación para que cualquier usuario pueda utilizarla sin ningún tipo de problema.

El contenido de este manual se va a estructurar en dos bloques. Por un lado, el manual de instalación de todo el sistema y por otro, el manual de usuario de CV\_UVA. Es decir:

### ● **Manual de instalación del sistema**

En este bloque se explicará, de forma detallada y lo más gráficamente posible, la instalación y configuración de la aplicación y los programas necesarios para que la aplicación funcione correctamente, para usarlo en conjunto con la aplicación desarrollada.

### ● **Manual de uso de la aplicación CV\_UVA**

En este bloque se explicarán las funciones disponibles en CV\_UVA, así como su uso y los posibles errores que puedan surgir durante un funcionamiento normal de la aplicación, apoyándose en ilustraciones reales de la aplicación.





## 11. INSTALACIÓN DEL SISTEMA

### 11.1 INSTALACIÓN SERVIDOR APACHE

En primer lugar vamos a distinguir dos tipos de APACHE, que se corresponderán con las versiones 1.3.X y 2.0.X de este programa. El motivo de la distinción es que se usan bastante ambas versiones dependiendo del sistema operativo y la configuración es ligeramente distinta en ambos.

Por ejemplo, ambas versiones son estables en sistemas WINDOWS XP, pero la 2.0.X puede volverse inestable bajo WINDOWS 98. Pero la idea principal de este manual es instalar los software lo más actualizado posible por tanto trabajaremos sobre Apache2.

Deberemos descargar el Apache en su versión para Windows. Para ello podemos acceder por la siguiente URL: <http://httpd.apache.org/download.cgi> En cualquier caso, podemos encontrar cualquier archivo necesario en la carpeta `httpd/binaries/win32` del servidor que usemos para la descarga (pincharíamos en *Other files* y eso nos conduciría al servidor seleccionado). Lo que es importante destacar es que lo que hay que destacar son los Binarios (*Binary*) para WINDOWS (Win32), no los códigos fuente.

Una vez descargado el apache configuremos un poco los directorios donde instalaremos los paquetes, para este manual usaremos los siguientes directorios.

**C:\WebServer\Apache**, donde instalaremos el APACHE

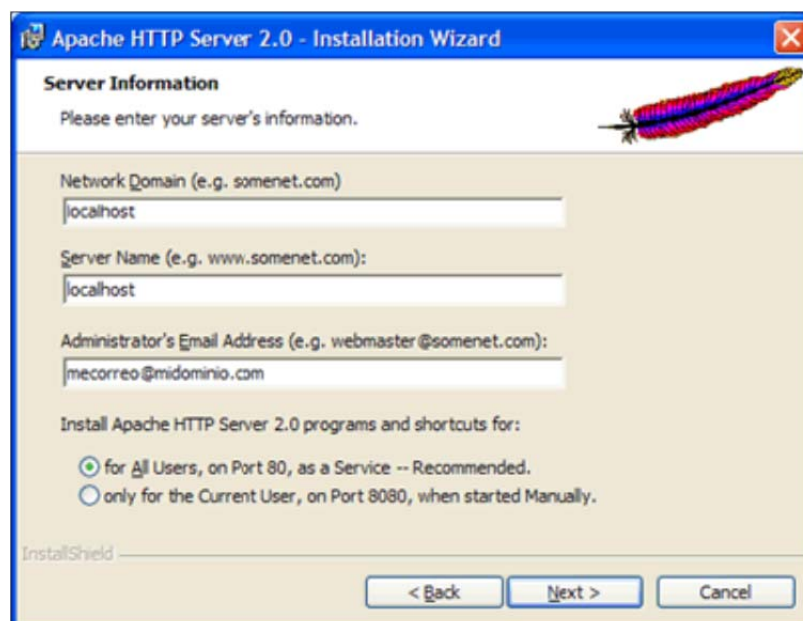
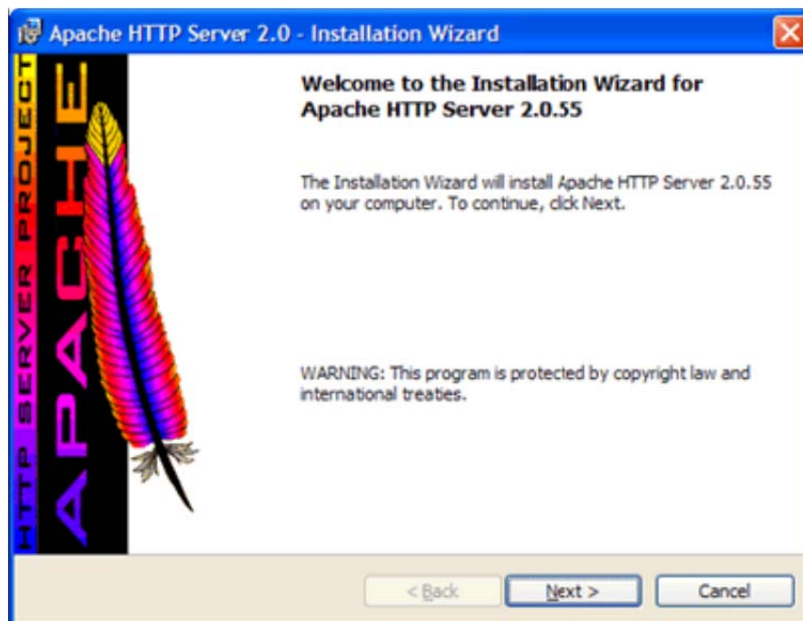
**C:\WebServer\PHP**, donde instalaremos el PHP

**C:\WebServer\MySQL**, donde instalaremos el MYSQL

**C:\WebServer\www**, directorio raíz para nuestros scripts PHP

Para esta versión, en la fecha de actualización del manual la última versión disponible era la 2.0.55, por lo que el archivo que tenemos que bajar puede ser: **apache\_2.0.55-win32-x86-no\_ssl.msi** También puede existir el mismo archivo pero con extensión exe, pero ésta ocupa un poco más de espacio en tu disco duro.

Bien, una vez que lo tengamos ejecutamos el instalador y vamos recorriendo las pantallas...

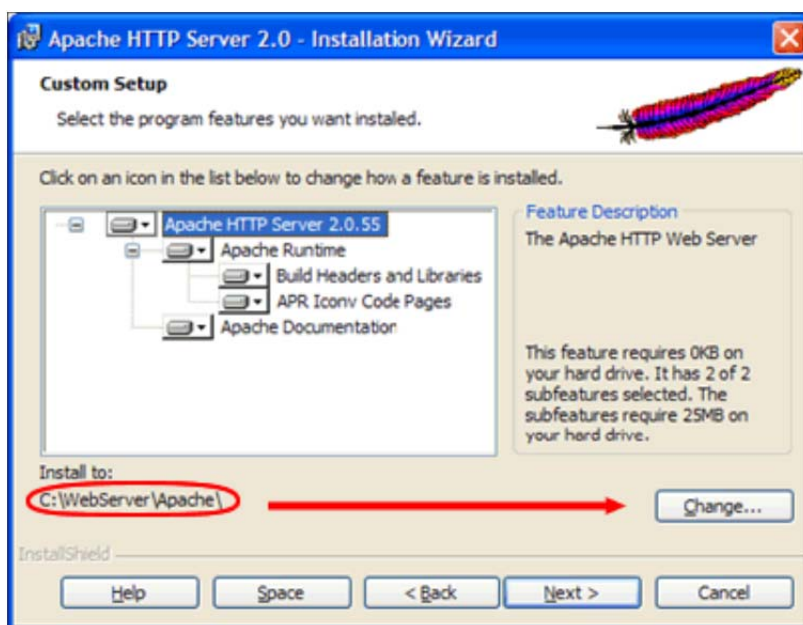
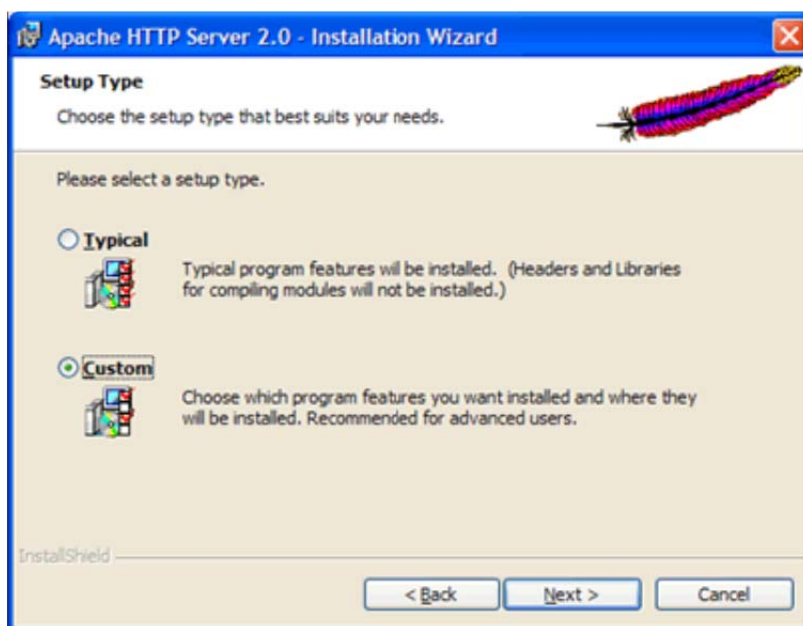


**Network Domain** :localhost ó 127.0.0.1

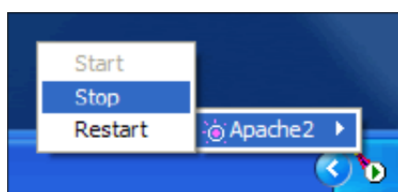
**Server Name**: localhost o 127.0.0.1

**Administrator's Email Address**: nuestro e-mail aunque no es totalmente necesario.

**For All Users, on Port 80, as a Service**: seleccionamos esta opción que básicamente configura apache como un servicio en el puerto 80.



Como se muestra en las 2 últimas imágenes solo debemos estar pendiente de seleccionar una instalación Custom o Personalizada y seleccionar el directorio que creamos con anterioridad para **APACHE** en **C:\WebServer\Apache**. Luego terminamos la instalación de APACHE. Seguido a esto nos aseguramos de cerrar el Apache para instalar el **PHP**.



## 11.2 INSTALACIÓN DE PHP5.

Procedemos a descargar el PHP para Windows. El archivo está localizado en la sección Downloads, apartado Windows Binaries, y es el Zip Package (no el installer, aunque éste ocupe menos luego no nos servirá). Una vez descargado todo el ZIP, extraemos los archivos del ZIP dentro de esa carpeta PHP, tal que los contenidos del ZIP quedarán dentro de la ruta: **C:\WebServer\PHP\**

Y ahora hay que seleccionar todos los archivos DLL localizados en la carpeta principal **C:\WebServer\PHP\** y copiarlos al directorio System (en Windows 9x) o System32 (NT, 2000, XP, 2003) de la carpeta del Windows (los archivos DLL contenidos en la carpeta EXT no hace falta copiarlos).

### 11.2.1 Configuración del archivo php.ini

El siguiente paso es configurar el php.ini. Renombramos o copiamos el archivo

**C:\WebServer\PHP\php.ini-dist** y le ponemos **php.ini**. Ahora lo editamos con el mismo block de notas. Si lo necesitáramos (lee antes la explicación), editamos la línea **register\_globals = Off** y la colocamos el valor: **register\_globals = On**

#### ¿Qué hago con register\_globals? ¿ON u OFF?

Activar esta directiva nos permite asumir que las variables son globales y pueden llegar por cualquier método (POST, GET, COOKIE, SERVER, etc). Así, por ejemplo, si utilizamos una variable global de sesión o cookie se puede suplantar fácilmente mediante una variable por url, con lo cual nuestro script no es seguro. Un buen programador de PHP tendría la directiva en OFF y usaría los arrays globales (\$HTTP\_X\_VARS) o los superglobales \$\_POST, \$\_GET, etc., que están disponibles a partir de la versión 4.1.X de PHP. ¿Y por qué? Pues por esos temas de seguridad en los script y porque debemos acostumbrarnos a no manejar variables globales ya que en un futuro el PHP tendrá la opción en OFF por defecto y no podremos cambiarla. Sin embargo, por temas de compatibilidad con script antiguos o que hagan uso de variables globales, podría interesarnos activar esta característica, pero repito que lo deseable sería tenerla en OFF y hacer uso de los arrays globales o superglobales.

A continuación vamos a indicar a PHP dónde se guardan las extensiones. Dentro del php.ini buscamos **extension\_dir** y le ponemos la carpeta que contiene los archivos **php\_xxx.dll**, que por defecto es la carpeta ext dentro de PHP.

**IMPORTANTE:** durante toda la configuración de directorios, debes utilizar esta barra "/" y no está "\", además de ponerlo entre comillas. O sea, que debe quedar así:

```
; Directory in which the loadable extensions (modules) reside.
```

```
extension_dir = "C:/WebServer/PHP/ext/"
```

Además podemos activar las extensiones que queramos o necesitemos, para lo cual buscamos Windows Extensions y para cargar las extensiones les quitamos el ";" de delante. Por ejemplo, si quisiéramos cargar la extensión gd2.dll (para manejar las funciones PHP relativas a imágenes) deberíamos cambiar ;extension=php\_gd2.dll por extension=php\_gd2.dll Puedes ver para qué sirve cada extensión en el [Manual oficial de PHP](#). Como nota adicional, resaltar que hay algunas extensiones que requieren de librerías extra que no vienen en el paquete completo de PHP, y para hacerlas funcionar tendremos que buscar dichas librerías. No actives todas las librerías a diestro y siniestro porque luego saldrán errores de que no se encuentra tal archivo; antes de instalar una librería, comprueba que en la carpeta de PHP tienes los archivos DLL correspondientes.

**IMPORTANTE:** la librería php\_mysql.dll es la que permite manejar las funciones relacionadas con MySQL, y por defecto en PHP5 viene desactivada, por tanto vamos a activarla de la forma que indicamos antes: buscamos ;extension=php\_mysql.dll y le quitamos el ; de delante.

Ahora, si vamos a hacer pruebas con upload de archivos vía HTTP, debemos indicar el directorio donde los archivos se almacenarán temporalmente. Para ello buscamos upload\_tmp\_dir y le damos el valor de una carpeta que exista. Por ejemplo, en nuestro directorio C:\Servidor\PHP\ creamos una carpeta uploads, por lo que quedará algo así:

```
; Temporary directory for HTTP uploaded files (will use system default if not  
; specified).
```

```
upload_tmp_dir = "C:/Servidor/PHP/uploads/"
```

Si queremos cambiar el tamaño máximo de los archivos que pueden subirse vía HTTP, buscamos **upload\_max\_filesize** y cambiamos el valor por defecto que trae, 2M (2 MB), por el que queramos. No se recomienda poner un valor alto.

```
; Maximum allowed size for uploaded files.
```

```
upload_max_filesize = 2M
```

Para trabajar con sesiones, debemos especificar un directorio donde se guarden los archivos temporales. Al igual que 2 pasos antes, buscamos **session.save\_path** y le damos el valor de un directorio que exista (o lo creamos): `session.save_path = "C:/Servidor/PHP/sessions/"`

A continuación copiamos el **php.ini** a la carpeta **C:\Windows\** o **C:\WinNT\** según nuestra versión de Windows.

¡Ahora vamos a decirle al Apache que tenemos el PHP!

### 11.2.2 Configuración del archivo `httpd.conf` de Apache

Hay que editar el archivo **httpd.conf** que encontramos en la carpeta **Conf** dentro del directorio del Apache.

Buscamos **Dynamic Shared Object (DSO) Support** que es donde se cargan los módulos. Ahí vamos a cargar el módulo de PHP para Apache, dando la dirección del archivo `php4apache2.dll` (PHP4) o `php5apache2.dll` (PHP5) que lo contiene. Encontramos hasta un ejemplo:

```
# Example:  
  
# LoadModule foo_module modules/mod_foo.so  
  
#
```

Y ahora añadimos justo debajo:

```
LoadModule php5_module C:/Servidor/PHP/php5apache2.dll
```

Tiene que quedar así:

```
# Example:
# LoadModule foo_module modules/mod_foo.so
#
LoadModule php5_module C:/Servidor/PHP/php5apache2.dll
```

El directorio predeterminado para guardar nuestras páginas es el htdocs del Apache:

**C:\Archivos de programa\Apache Group\Apache\htdocs\**

Pero podemos cambiarlo fácilmente. Buscamos **DocumentRoot** y nos sale esto:

```
#
# DocumentRoot: The directory out of which you will serve your
# documents. By default, all requests are taken from this directory, but
# symbolic links and aliases may be used to point to other locations.
#
DocumentRoot "C:/Archivos de programa/Apache Group/Apache/htdocs"
```

Nosotros sustituimos la carpeta htdocs del Apache por la que queramos. Todos los archivos que vayamos a probar con el Apache deberán estar localizados en esta carpeta. Por ejemplo nos creamos una carpeta WEB dentro del directorio del servidor:

```
DocumentRoot "C:/Servidor/WEB/"
```

Por lo que construiremos toda nuestra página dentro de ese directorio, que equivale al directorio raíz de nuestro host local.



Ahora buscamos el **DirectoryIndex** y nos sale algo como esto:

```
#
# DirectoryIndex: sets the file that Apache will serve if a directory
# is requested.
#
# The index.html.var file (a type-map) is used to deliver content-
# negotiated documents. The MultiViews Option can be used for the
# same purpose, but it is much slower.
#
DirectoryIndex index.html index.html.var
```

Modificamos la última línea para que si entramos en un directorio tipo **http://127.0.0.1/directorio/** nos autoejecute el index predeterminado (si existe) y no nos salga un mensaje de error. Puedes usar más nombres si lo prefieres. Se ejecutarán por orden: si no encuentra el primero, pasa al segundo; si no está el segundo, pasa al tercero, y así sucesivamente hasta que si no encuentra ninguno entonces da error. En la siguiente línea, si en nuestro directorio tenemos un index.htm y un index.php, por defecto se ejecutará el index.htm ya que está antes.

```
DirectoryIndex index.html index.htm index.php index.php3 index.php4 index.phtml
index.html.var
```

Ahora le añadimos debajo estas líneas (en la primera de ellas, si estamos instalando PHP5 también podemos añadir la extensión .php5 además de las que aparecen):

```
AddType application/x-httpd-php .php .php3 .php4 .phtml
```

```
AddType application/x-httpd-php-source .phps
```

La primera indica las extensiones que serán interpretadas por el Apache. Por ejemplo podemos añadir la extensión .htm o .html para que el Apache ejecute el código PHP contenido en esas páginas (es decir, el uso de código PHP no está limitado exclusivamente a archivos \*.PHP). La segunda sirve para que si entras en una página **loquesea.php** entonces se muestra el código PHP a color, muy útil si queremos por ejemplo mostrar el código fuente a color en una página. De tal forma que al final todo queda así:

```
#  
  
# DirectoryIndex: sets the file that Apache will serve if a directory  
# is requested.  
  
#  
  
# The index.html.var file (a type-map) is used to deliver content-  
# negotiated documents. The MultiViews Option can be used for the  
# same purpose, but it is much slower.  
  
#  
  
DirectoryIndex index.html index.htm index.php index.php3 index.php4 index.phtml  
index.html.var  
  
AddType application/x-httpd-php .php .php3 .php4 .phtml  
  
AddType application/x-httpd-php-source .phps
```

Y ya podemos guardar el archivo httpd.conf

### 11.2.3 Probando nuestro servidor Apache

Ahora vamos a probar el Apache y PHP, para lo cual deberemos arrancar el Apache.

Accederemos desde los accesos directos que se crean en el menú de inicio al acceso directo.

**Monitor Apache Servers** y aparecerá un icono al lado del reloj. Cliqueando en el icono, sale un menú desde donde podremos iniciar, apagar y reiniciar el Apache. Como queremos encenderlo, si no lo hace automáticamente le daremos a Start. Si ya estaba encendido le daremos a Restart (útil si cambiamos alguna configuración de PHP "al vuelo") para que al reiniciarlo nos actualice los cambios que hemos hecho.

Si todo ha ido bien, podemos crear un archivo llamado por ejemplo **index.php** dentro de nuestro directorio raíz en **C:\WebServer\www** cuyo contenido sea exclusivamente el siguiente:

```
<?  
  
phpinfo();  
  
>
```

Si accedemos mediante nuestro navegador a la dirección **http://127.0.0.1/info.php** o **http://localhost/info.php** (recuerda que 127.0.0.1 = localhost) con el Apache activo deberemos visualizar una página de información de PHP. Si no sale, algo ha fallado, revisa los pasos.

PHP Version 5.1.2	
System	Windows NT SERVIDOR 5.1 build 2600
Build Date	Jan 11 2006 16:35:21
Configure Command	cscript inologo configure.js "--enable-snapshot-build" "--with-gd=shared"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\WINDOWS\php.ini
PHP API	20041225
PHP Extension	20050922
Zend Extension	220051025
Debug Build	no
Thread Safety	enabled
Zend Memory Manager	enabled
IPv6 Support	enabled
Registered PHP Streams	php, file, http, ftp, compress.zlib, compress.bzip2
Registered Stream Socket Transports	tcp, udp
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*.zlib.*, bzip2.*
This program makes use of the Zend Scripting Language Engine: Zend Engine v2.1.0, Copyright (c) 1998-2006 Zend Technologies	

Como has podido observar, para acceder a nuestro servidor local y así poder probar nuestros script y las páginas de nuestra web, siempre accederemos con la dirección **127.0.0.1** o **localhost** (sin WWW) en el navegador pues esta es la dirección del host local de nuestro ordenador. En general, para visualizar o acceder a cualquier archivo (PHP, HTML, TXT, ZIP, GIF, JPG, etc.) que tengamos en nuestra carpeta de la web, lo haremos de la siguiente forma:

**http://localhost/CARPETA/ARCHIVO.EXT**

Donde CARPETA es la carpeta (o serie de carpetas) relativas a la raíz en que se encuentra el ARCHIVO.EXT

Por tanto resaltar que no sólo podemos hacer llamadas a script PHP, sino también a páginas HTML, descarga de archivos, imágenes, etc., tan solo poniendo la dirección correcta en el navegador. Es simplemente un servidor.

De esta forma, podemos tener nuestra web funcionando en nuestro propio ordenador para construirla desde cero o bien hacer las pruebas que necesitemos sin tener que estar conectados a Internet.

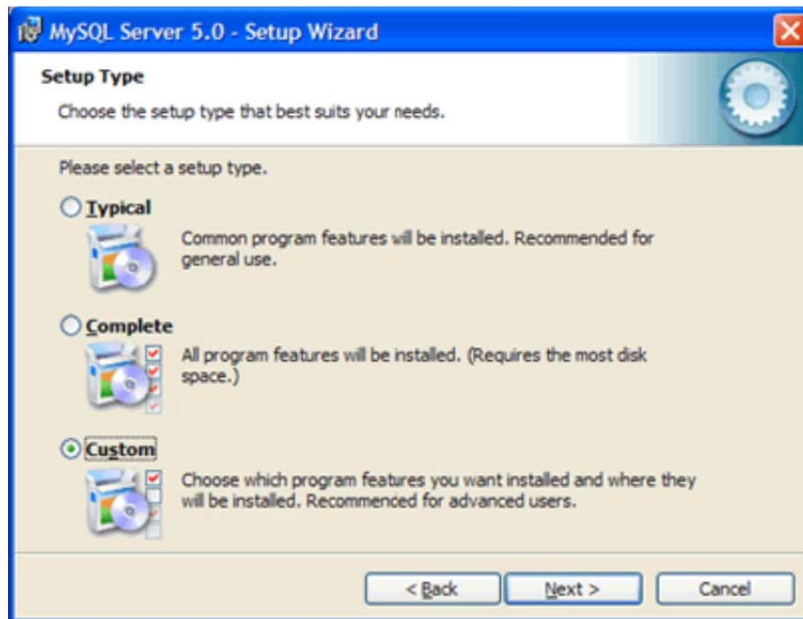
## 11.3 INSTALACIÓN DE MySQL

Web oficial: <http://www.mysql.com>

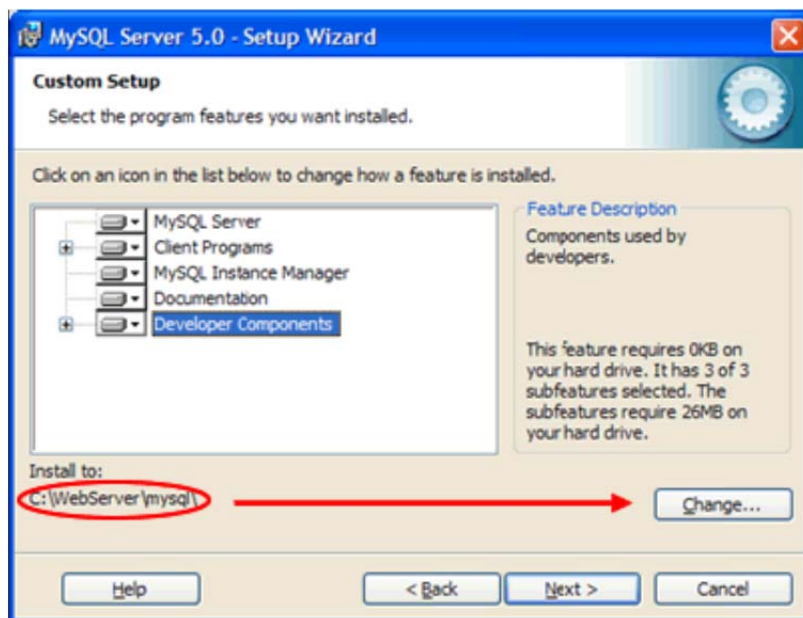
Lo primero por supuesto es bajarnos la última versión estable de MySQL en su sección de downloads de su página principal. <http://dev.mysql.com/downloads/> para el momento de escribir este tutorial la última versión estable es la 5.0.19.

Lo que más me ha llamado la atención de MySQL 5 desde su salida fue su instalador que es realmente sencillo y la herramienta de configuración es simplemente espectacular es por ellos que voy a mostrar las pantallas de las preguntas que hace este instalador para configurarlo a nuestro gusto, recordemos que la misión de este manual es enseñar a montar un servidor local y no un servidor de base de datos dedicado ni nada de eso...

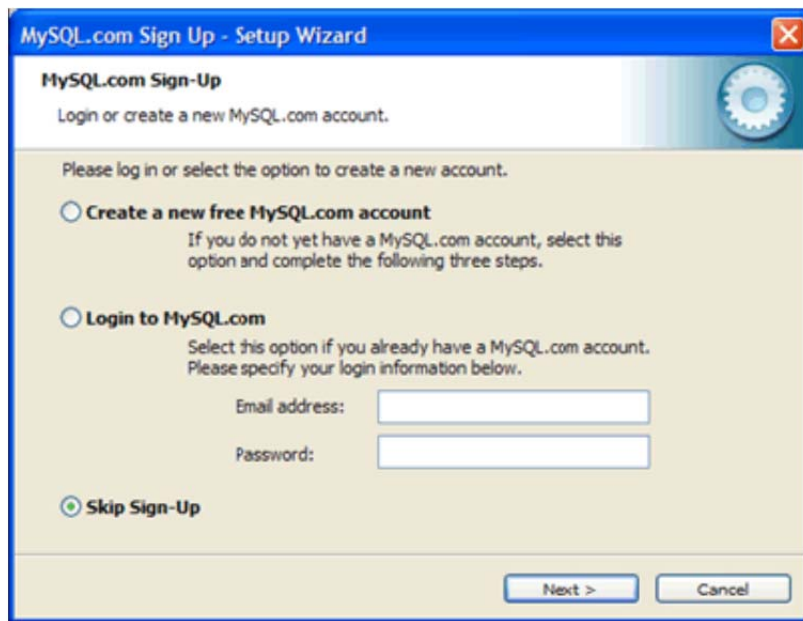




Elegimos la opción Custom para poder tener el control de que vamos a instalar y donde los vamos a instalar...



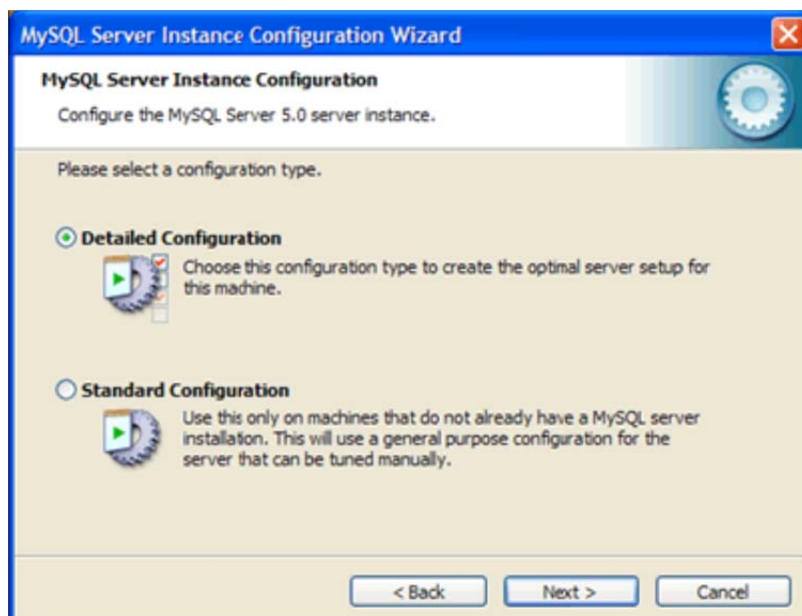
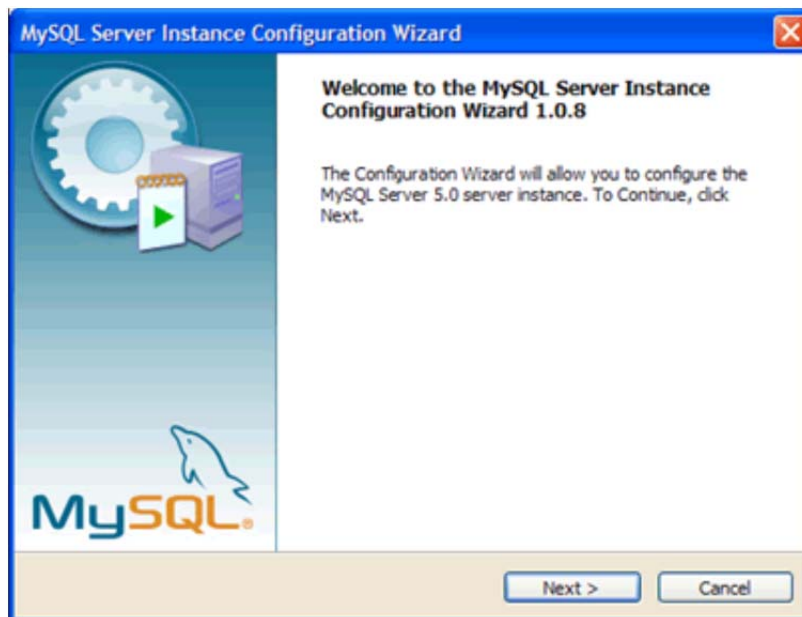
En la pantalla de arriba vemos como configurar el directorio de instalación del MySQL. Previamente habíamos creado uno en C:\WebServer\MySQL



Pues esta opción de registrarnos en MySQL.com es opcional así que la saltamos...



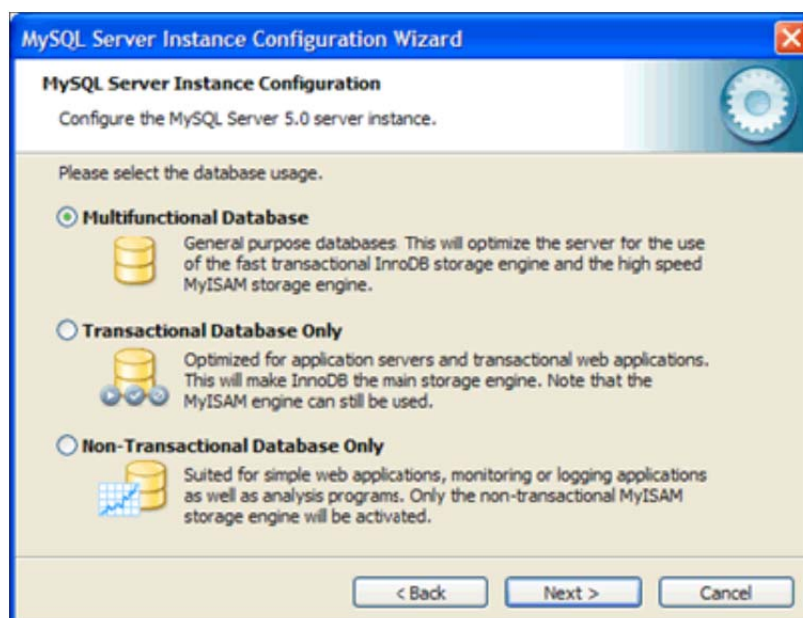
Si deseamos configurar el mysql lo podemos hacer ahora y luego desde el menú Inicio/Programas/MySQL/MySQL Server 5



Seleccionamos una configuración detallada para controlar mejor el proceso de configuración.

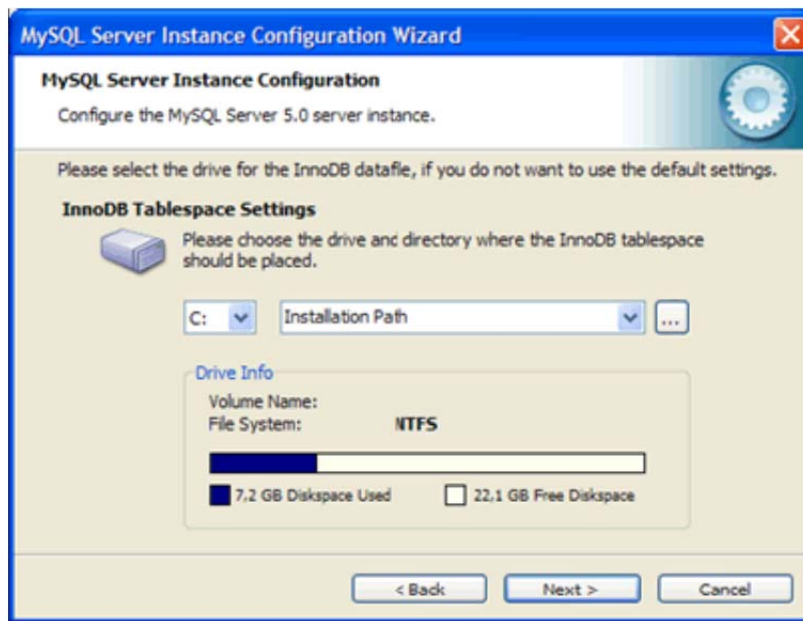


Como nuestro objetivo es configurar MySQL para un servidor local de pruebas seleccionamos la primera opción "Developer Machine" => Máquina de desarrollo.



Seleccionamos "Multifunctional Database" esto permite el uso de tablas MyISAM, InnoDB, entre otras...

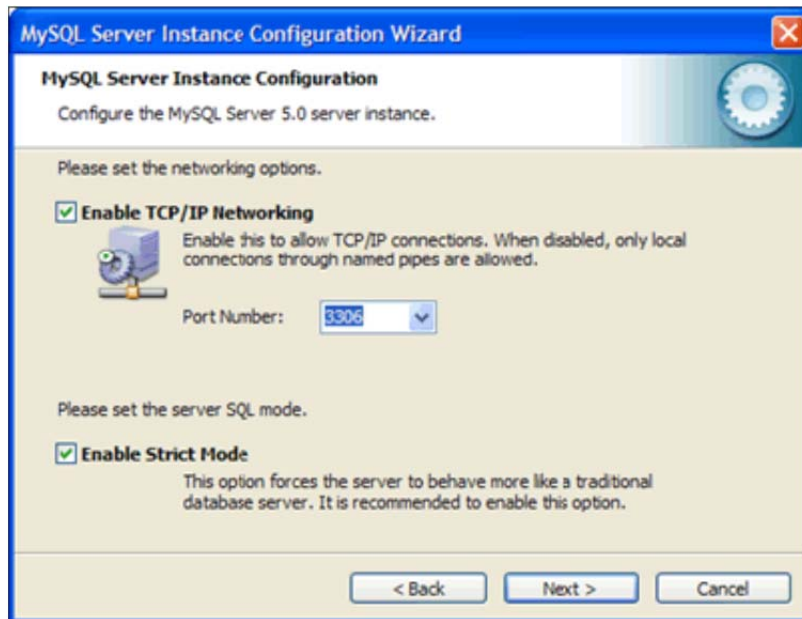




Esta parte la dejamos tal cual.....



Esta opción nos permite especificar el número de conexiones simultáneas que nuestro MySQL va a aceptar... con 10 en nuestro localhost es más que suficiente...

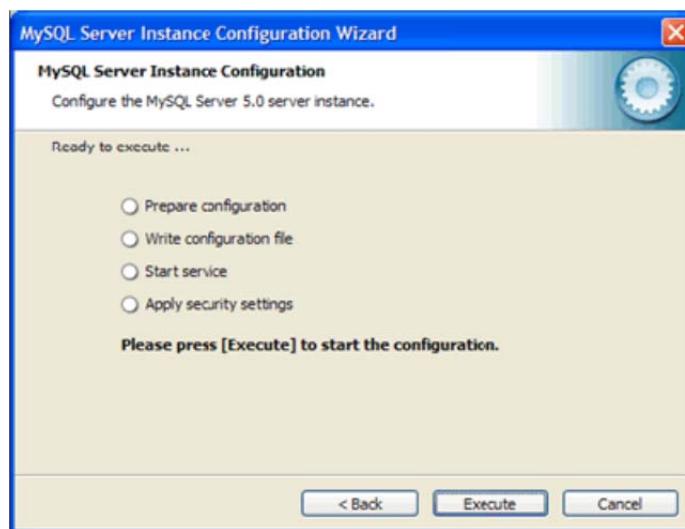


Un punto importante si vamos a trabajar en red habilitar las conexiones TCP/IP...





En este punto establecemos la contraseña del root y habilitamos el acceso remoto del mismo.



Y eso es todo solo faltaría ejecutar todas las configuraciones hechas y Welcome to MySQL 5

## 11.4 INSTALACIÓN DE phpMyAdmin

phpMyAdmin es una herramienta escrita en **PHP** con la intención de manejar la administración de **MySQL** a través de páginas webs, utilizando **Internet**. Actualmente puede crear y eliminar Bases de Datos, crear, eliminar y alterar tablas, borrar, editar y añadir campos, ejecutar cualquier sentencia **SQL**, administrar claves en campos, administrar privilegios, exportar datos en varios formatos y está disponible en 50 idiomas.

Si accedemos a la página oficial [www.phpmyadmin.net](http://www.phpmyadmin.net), nos encontramos en la sección de downloads con una lista de links, y el que tenemos que elegir es el ZIP de la última versión. En nuestro caso vamos a utilizar la versión 2.7 de esta aplicación.

La instalación es relativamente sencilla: extraer todo el ZIP en la carpeta raíz de nuestra web. OJO: en la carpeta de nuestra web, **C:\WebServer\www\** y no dentro de la carpeta del PHP o similares. Como pone un nombre extraño a la carpeta, mejor la renombramos a phpmyadmin, de forma que quedaría algo así: C:\WebServer\www\phpmyadmin\

Ahora vamos a hacer una pequeña configuración del phpMyAdmin. Para ello debemos editar el archivo `config.default.php` y, leyendo de arriba hacia abajo, buscamos la primera aparición de la siguiente línea:

```
$cfg['PmaAbsoluteUri'] = '';
```

Debemos darle la ruta absoluta donde tenemos el phpMyAdmin. En nuestro caso sería así:

```
$cfg['PmaAbsoluteUri'] = 'http://localhost/phpmyadmin/';
```

Recuerda que podemos poner **localhost** o **127.0.0.1** . Ahora buscamos si nos aparece algo como esto:

```
$cfg['blowfish_secret'] = '';
```

Le pondremos una cadena de caracteres cualquiera, que servirá de semilla para la encriptación de contraseñas al usar la autenticación con cookies:

```
$cfg['blowfish_secret'] = 'aqui puedes poner lo que quieras ;) ';
```

Por último buscamos

```
$cfg['Servers'][$i]['password'] = ''; // MySQL password (only needed);
```

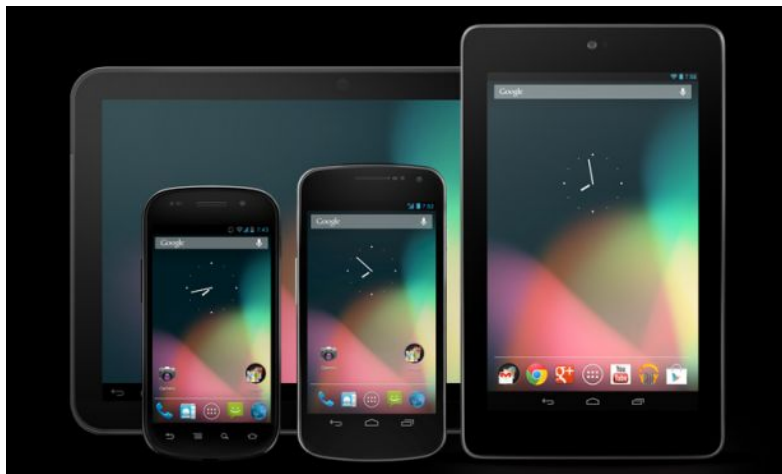
y colocamos la contraseña del root que le asignamos en el configurador de mysql

```
$cfg['Servers'][$i]['password'] = 'MI PASSWORD'; // MySQL password  
(only needed);
```

Y guardaremos el archivo, pero no lo cerraremos. Puedes probar phpMyAdmin para ver si funciona, ya sabes: <http://localhost/phpmyadmin/>

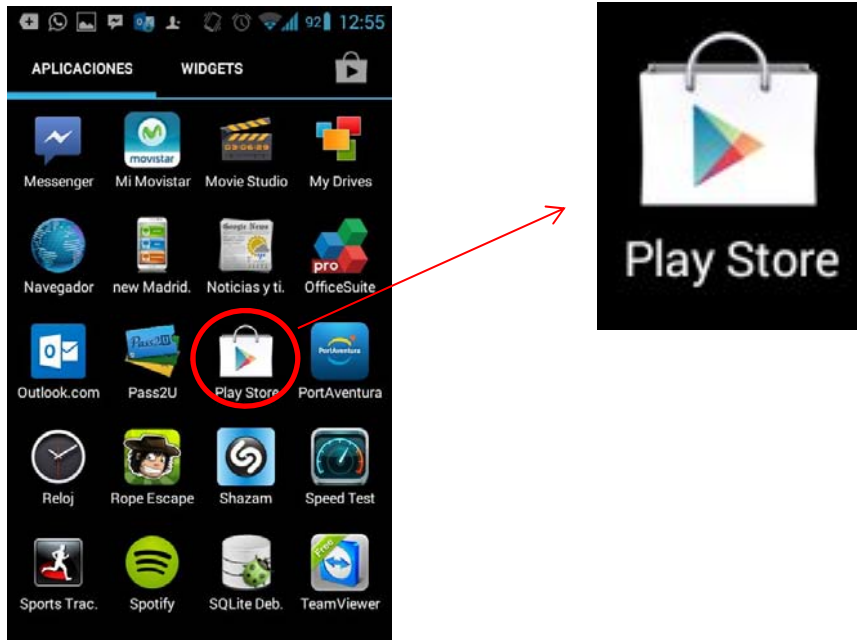
## 11.5 INSTALACIÓN CV\_UVA

La aplicación CV\_UVA se puede visualizar desde cualquier dispositivo que tenga como sistema operativo Android 2.1 o superior.

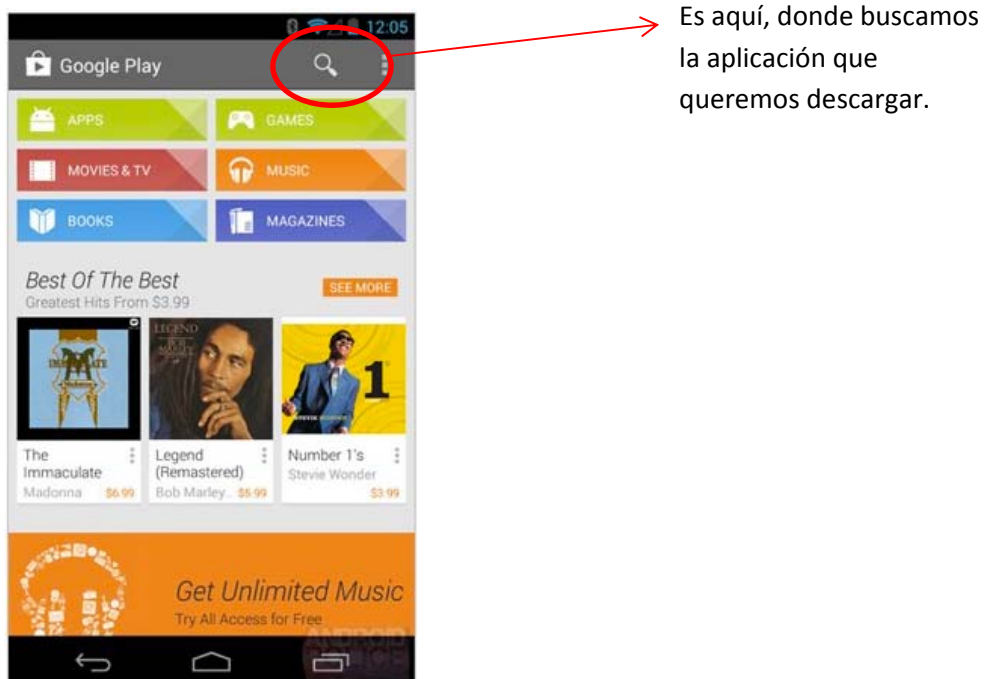


Lo primero que tenemos que hacer, es localizar en nuestro dispositivo móvil el icono de “Play Store” es ahí, donde se encuentran todas las aplicaciones disponibles para los dispositivos móviles Android.

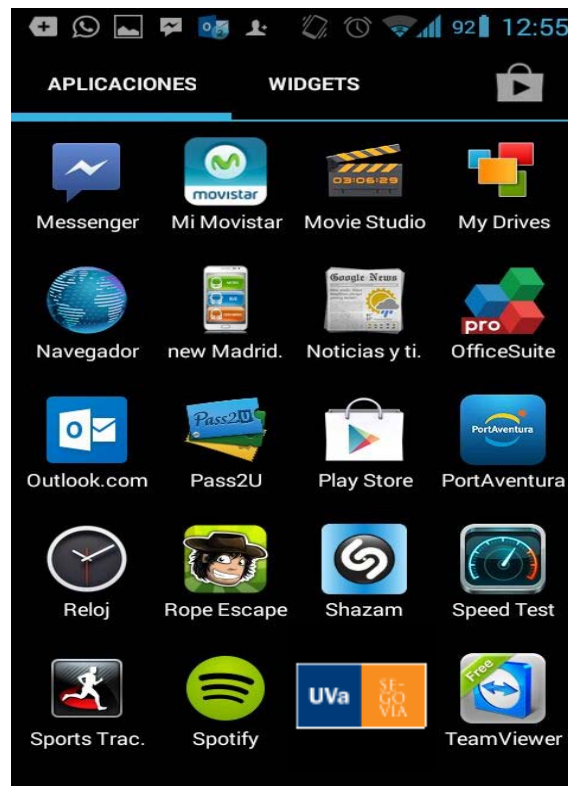
El icono tiene el siguiente aspecto:



Una vez que hemos pinchado en el icono y estamos dentro de la aplicación “Play Store”, buscamos la aplicación que deseamos descargar, en nuestro caso “CV\_UVA”.



Cuando la hemos localizado, y la hemos instalado, aparecerá en nuestro dispositivo móvil en la pantalla donde se encuentren todas las aplicaciones que tenemos instaladas.



Ahora, solo hay que empezar a usarlo!!!!

## 12. MANUAL DE USUARIO DE CV\_UVA

### 12.1 ¿QUÉ ES CV\_UVA?

CV\_UVA proviene de la unión de las siglas CV (Campus Virtual) y UVA (Universidad de Valladolid).

Las funciones disponibles en CV\_UVA son upload, borrado y visualización de archivos PDF en función de la asignatura que visitemos. Según el tipo de usuario (profesor o alumno) se podrá o no, realizar la primera parte de las funciones (upload/borrado).

En los próximos apartados veremos en detalle todas las funcionalidades de este Software.

### 12.2 REQUISITOS MÍNIMOS

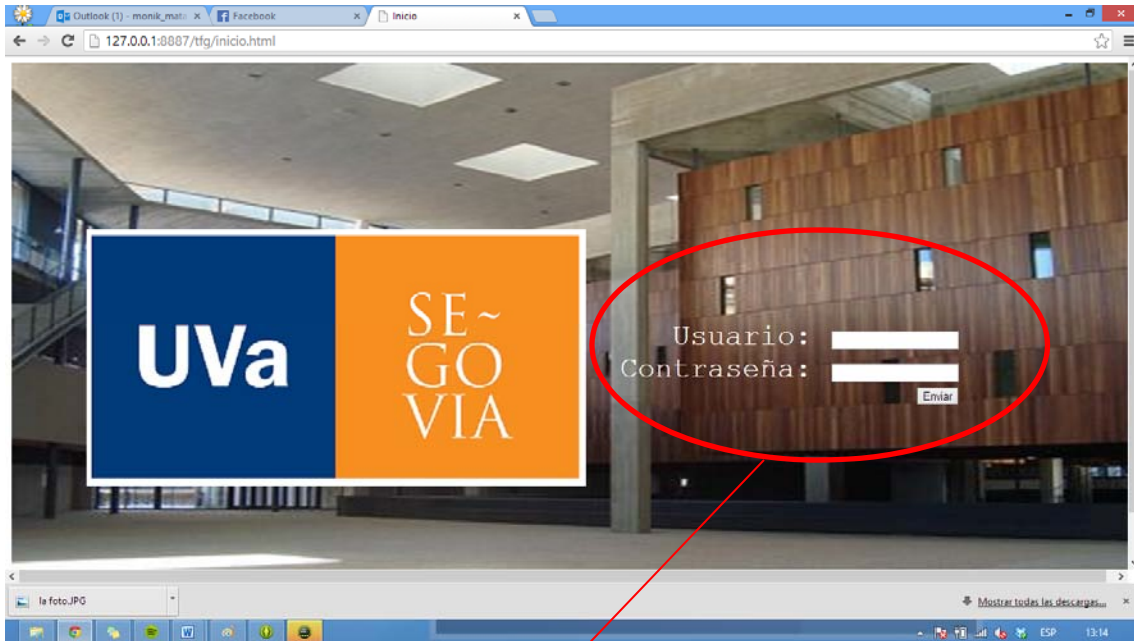
CV_UVA (Web)	CV_UVA (Android)
<b>Hardware</b> <ul style="list-style-type: none"> <li>• PC (sin ningún requisito especial)</li> <li>• Router WiFi</li> </ul>	<b>Hardware</b> <ul style="list-style-type: none"> <li>• Dispositivo móvil (con WiFi/3G, memoria RAM de más de 128mb y más de 1mb libre de memoria interna)</li> </ul>
<b>Software</b> <ul style="list-style-type: none"> <li>• Navegador</li> <li>• Apache</li> <li>• MySQL</li> <li>• PHP</li> </ul>	<b>Software</b> <ul style="list-style-type: none"> <li>• Android 2.1 o siguientes</li> <li>• Lector de PDF</li> </ul>



## 12.3 UTILIZACIÓN CV\_UVA (WEB)

Como venimos diciendo a lo largo de toda esta documentación, la parte de la aplicación Web, solo puede ser usada por el profesorado de la Universidad.

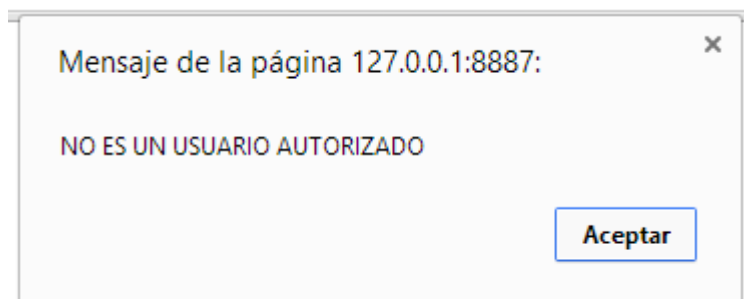
La pantalla principal de la aplicación Web es la siguiente:



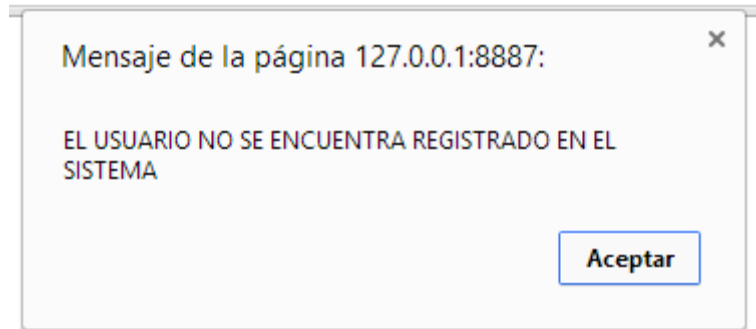
Formulario para iniciar sesión

Los errores que pueden surgir a la hora de iniciar sesión pueden ser los siguientes:

- El usuario no es profesor, no está autorizado, con lo cual no nos deja entrar.



- El usuario no meta bien los datos, o los deje en blanco, el sistema entiende que ese usuario que intenta acceder no está en el sistema.



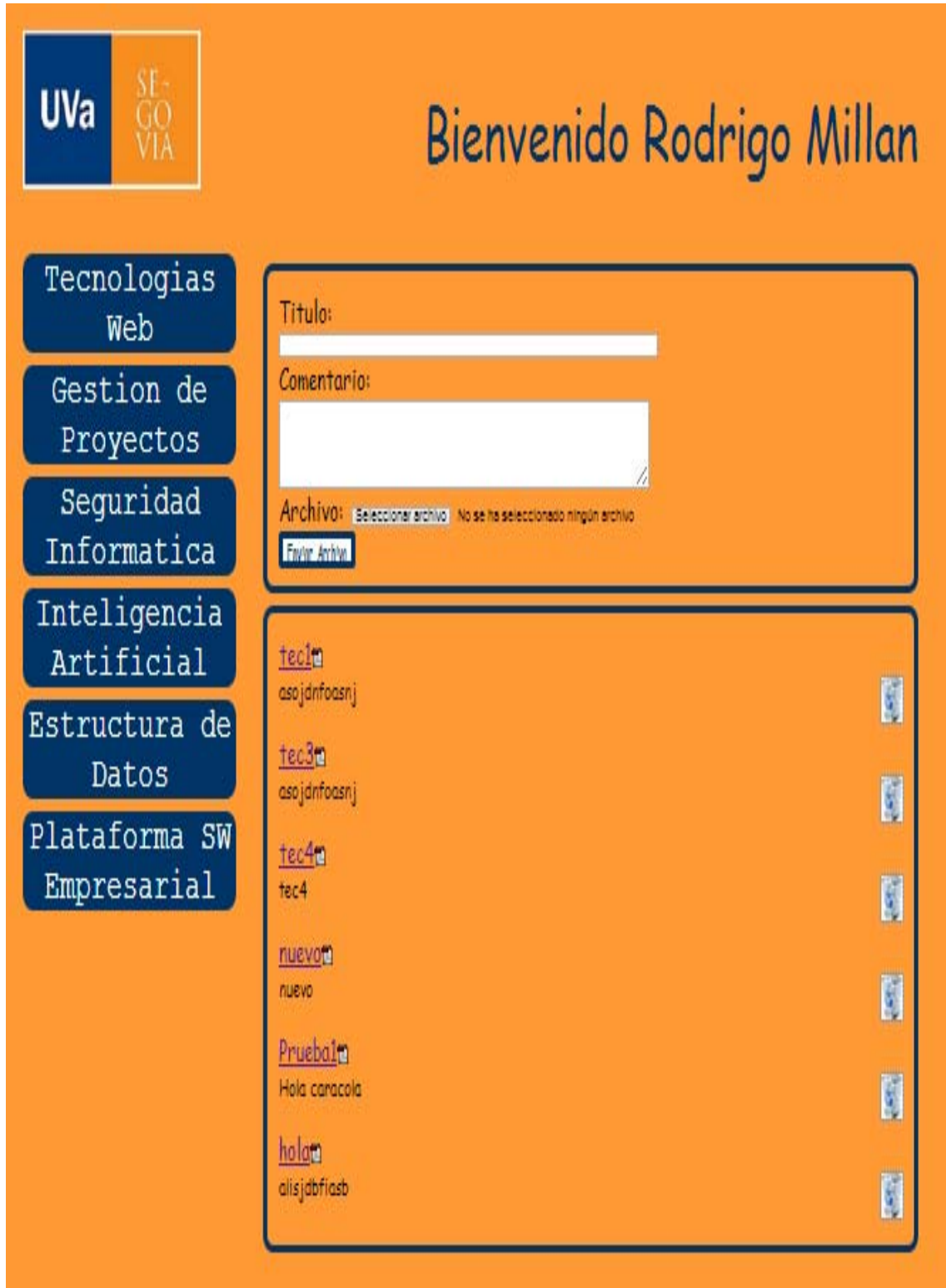
Una vez que hemos iniciado sesión correctamente, la aplicación mostrará las asignaturas que dicho profesor imparte y le dará la bienvenida.

Da la bienvenida al profesor



Muestra las asignaturas que dicho profesor imparte y por lo tanto, donde podrá subir/borrar archivos PDF.

Cuando el profesor selecciona la asignatura sola la cual quiere realizar cambios, ya sea añadir archivo PDF, borrar alguno de los existentes o simplemente consultar los archivos que están ahora mismo en el sistema, se encuentra con esta pantalla.



Esta pantalla, mucho más completa y compleja que la anterior, está dividida en 4 partes claramente diferenciadas, y cada una de ellas, tiene una función.

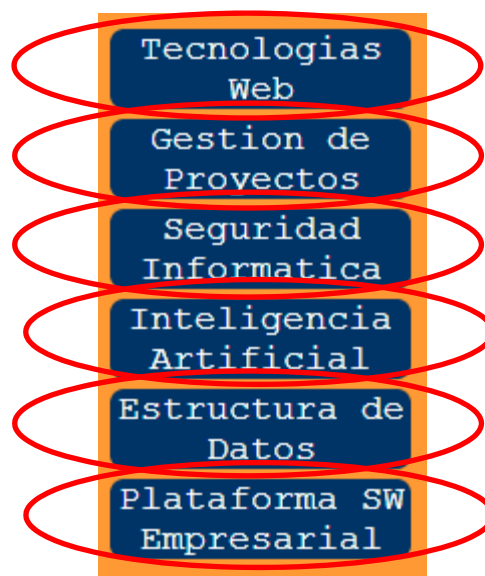
### CERRAR SESIÓN

Si pinchamos sobre el icono de la universidad, la aplicación nos lleva a la pantalla principal de la aplicación, donde tendremos que volver a iniciar sesión para poder acceder a la aplicación.



### VIVITAR OTRAS ASIGNATURAS

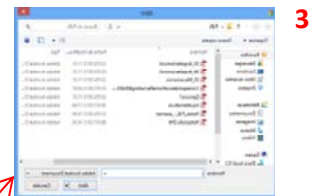
Haciendo click sobre cualquiera de las otras asignaturas, nos llevará a la pantalla con los datos de la asignatura seleccionada. Todas tienen el mismo aspecto y organización, lo único que cambia, son los archivos que cada una de las asignaturas tiene.



## ● AÑADIR ARCHIVOS

Gracias al formulario, podemos añadir nuevos archivos en la asignatura en la que nos encontremos.

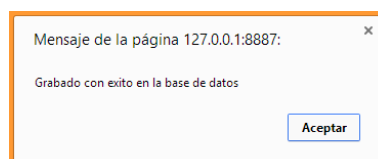
1. Cada archivo contará con un título y un comentario, para que el profesor ponga pequeñas aclaraciones que le puedan venir bien en un futuro.
2. Cuando damos al botón de seleccionar archivo, se nos muestra un explorador de archivos donde localizaremos el archivo que deseamos subir.
3. Una vez que tenemos rellenos los campos, hacemos click sobre enviar archivos para que se almacene en la BD y así podrá estar disponible tanto para esta aplicación, como para la aplicación Android.



Explorador de archivos

4. Si hemos dejado vacío el campo título o el archivo a subir, el sistema nos tendrá que dar un mensaje de error

5. Si todo ha ido bien, y el archivo se ha subido correctamente a la BD para poder visualizado en la aplicación Web o en de Android, el sistema muestra un mensaje de confirmación.



## VISUALIZAR/BORRAR ARCHIVO EXISTENTE

En la parte de abajo/derecha de la pantalla, nos encontramos con la lista de los archivos que tiene la asignatura seleccionada.

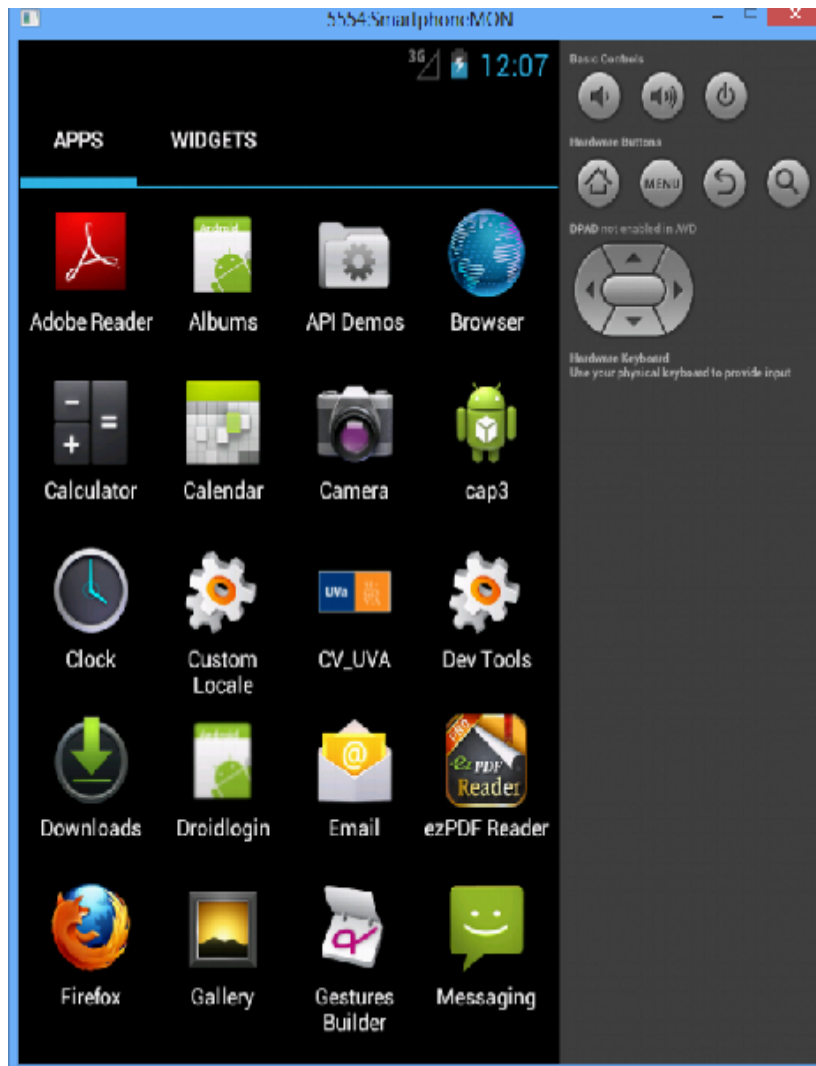
Con cada uno de los archivos, podremos hacer dos cosas:

1. Verlo. Pinchando sobre el título del archivo se abrirá una nueva pestaña en el navegador, donde podremos visualizar el archivo seleccionado.
2. Borrarlo. A la derecha de cada uno de los archivos, tenemos un botón con una papelera de reciclaje, si pinchamos sobre ella, el archivo se borrará tanto de la lista, como de la BD y ya no estará disponible para ser visualizado ni en la aplicación Web ni en la de Android.  
Cuando borramos un archivo, se muestra un mensaje de confirmación de borrado.

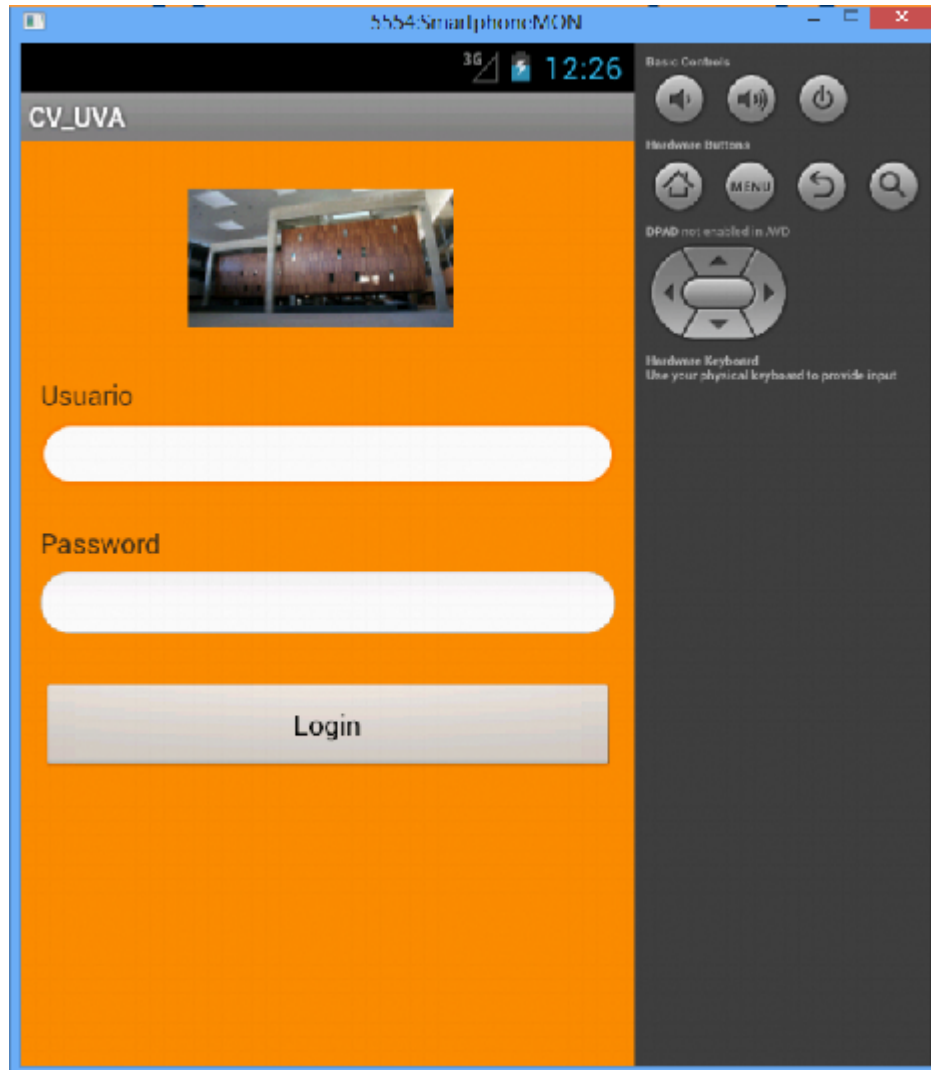


## 12.4 UTILIZACIÓN CV\_UVA (ANDROID)

Como ya hemos explicado en algún apartado anterior, para poder utilizar la aplicación CV\_UVA primero la hemos tenido que instalar en nuestro dispositivo móvil. Una vez ahí, solo tenemos que tocar el icono de la aplicación para que comience a ejecutarse.

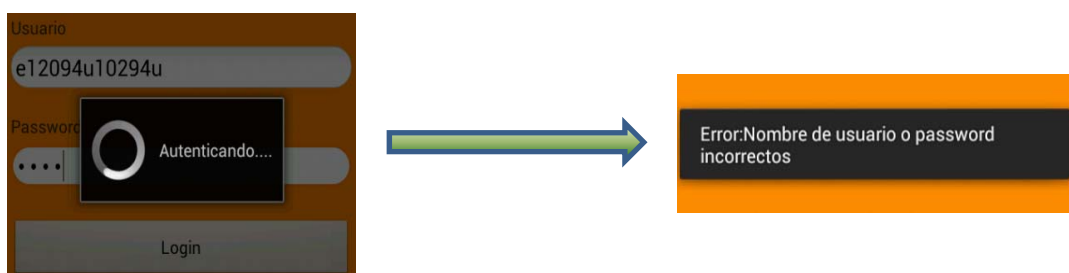


Una vez que se ha iniciado CV\_UVA, la prima pantalla que nos encontramos es la de inicio de sesión, al contrario que en la aplicación web, en ésta, podrá iniciar sesión tanto los profesores como los alumnos.



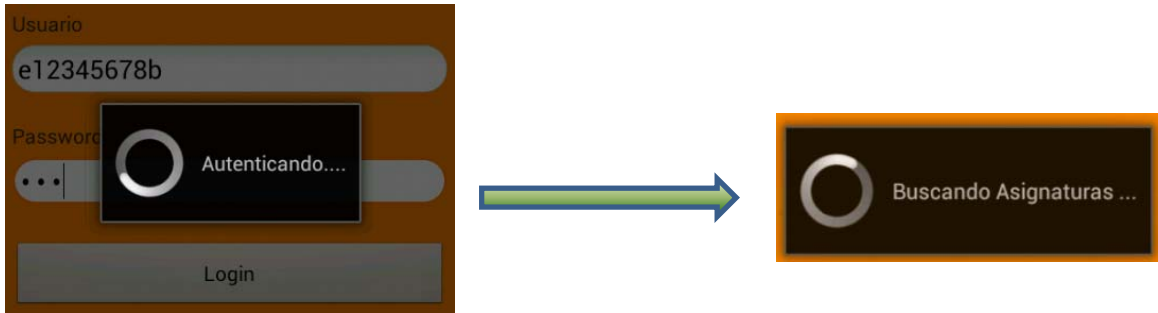
Como se muestra en la imagen, tendremos que poner nuestro nombre y nuestra password para poder acceder los datos que contiene la aplicación.

Si los datos introducidos son incorrectos o se ha dejado en blanco alguno de los dos campos del formulario, la aplicación nos mostrará un mensaje de error impidiendo así, acceder a la aplicación.





Si el usuario y la contraseña que hemos introducido están en un formato correcto y perteneces a un usuario de la aplicación, el sistema buscará las asignaturas para componer la siguiente pantalla..

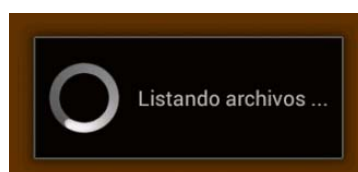


Ésta mostrará en forma de lista:

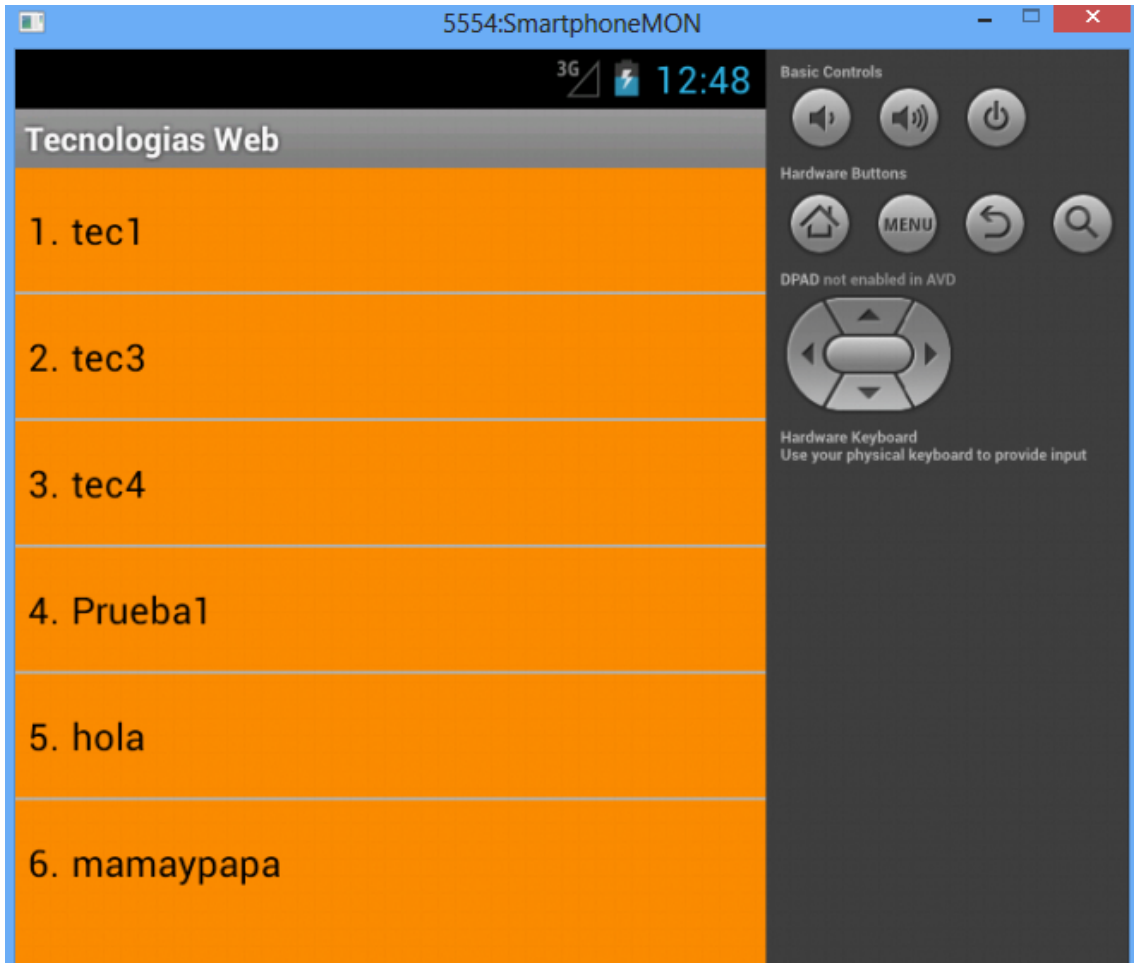
1. Las asignaturas que tiene ese usuario, ya sean matriculadas (alumno) o impartidas (profesor)
2. El número de archivos que contiene esa asignatura



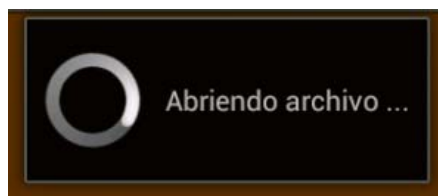
Para poder acceder a una de las asignaturas y poder ver los archivos que contiene, solo hace falta tocar sobre la asignatura y esperar a que se carguen los títulos de los archivos PDF.



Una vez que se han buscado los archivos PDF en la BD, se muestran al igual que las asignaturas (en una lista) de la siguiente forma.



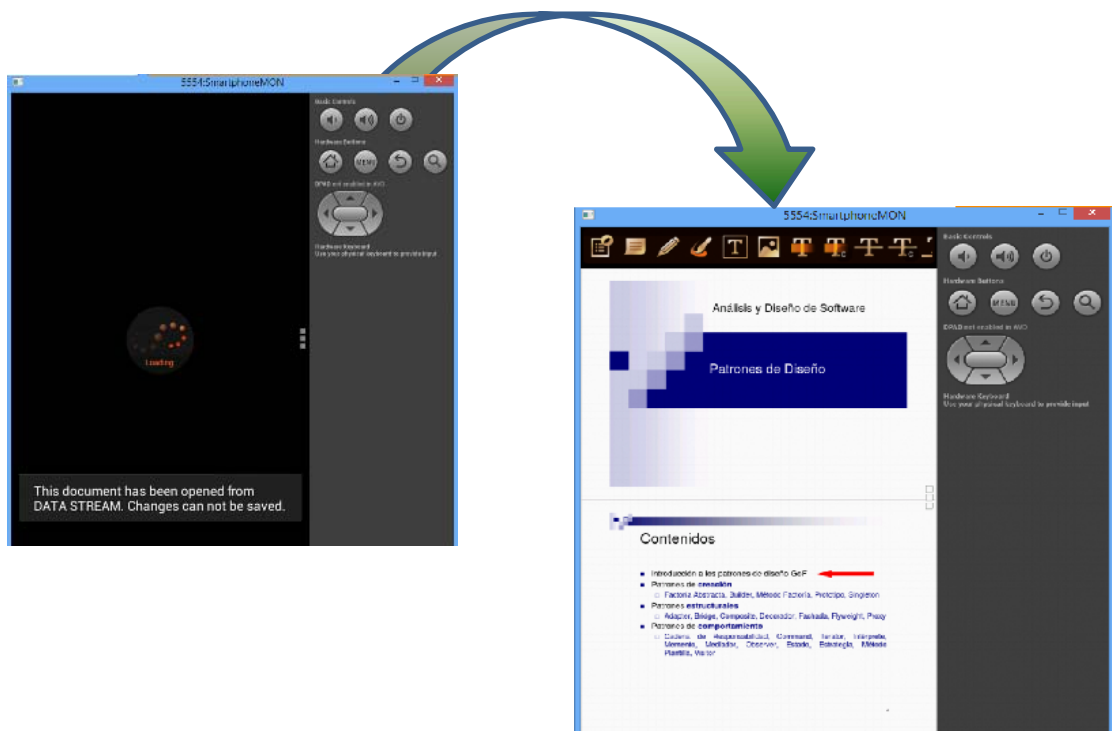
Para poder ver un archivo en concreto, se procede de la misma forma que en la anterior pantalla, pinchando sobre el archivo que queremos ver.



En la siguiente pantalla se muestra los datos del archivo PDF que queremos visualizar, en este caso, tenemos que tocar sobre el botón que nos indica “Ver Archivo” para poder visualizarlo con una aplicación apropiada al formato que tienen los archivos.



Una vez que hemos tocado sobre el botón.....



## 13. EVALUACIÓN

Después de realizar las pruebas que se han expuesto en el anterior capítulo, podemos hacer una evaluación sobre cómo funciona nuestra aplicación dentro del sistema.

Con esta aplicación, buscamos una serie de características que consideramos fundamentales, y teniendo éstas en cuenta, vamos a enunciar distintos puntos sobre las pruebas realizadas.

### 13.1 Evaluación del rendimiento

(Las pruebas de rendimiento son relativas a la velocidad de conexión a la red)

El rendimiento en cuanto a tiempos de carga de páginas y procesamiento de datos es relativamente rápido ya que se realizan casi todas las operaciones en menos de 1 segundo. La operación que tarda más tiempo en procesarse es la búsqueda por nombre o lugar, que depende del número de rutas a comparar puede llegar a tardar unos segundos más.

### 13.2 Evaluación de la robustez

Si el servidor web se cae, tanto la aplicación Web como la Android dejan de funcionar por completo..

Si el servidor de la base de datos se cae, la página emitirá constantes errores de conexión con la base de datos. En el caso de que no se haya accedido a la aplicación con usuario y contraseña, ni si quiera se podría acceder. Lo que supondrá que no se podrá hacer uso de ninguna de las características de ambas aplicaciones, puesto que todo está sujeto al funcionamiento de la BD.

### 13.3 Evaluación de la seguridad

Las contraseñas de acceso de usuarios están codificadas con la función md5 de PHP y se almacenan en la base de datos con este método de cifrado.

Además gracias a la carga dinámica y al método de envío de datos *POST*, la url es siempre constante, lo cual permite un grado de abstracción de las direcciones de las páginas utilizadas en cada momento.



## 14. CONCLUSIONES

Tras realizar el presente proyecto, las conclusiones positivas que he sacado son las siguientes:

### ● **FACILIDAD DE DESARROLLO**

Una de las principales ventajas del desarrollo de aplicaciones ANDROID es que no se ha optado por crear un nuevo lenguaje de programación desde cero, sino que se usó como base JAVA, lo que también permite al desarrollador reutilizar librerías creadas para otros proyectos JAVA. Además, como veremos a continuación, existe mucha documentación para realizar casi cualquier tarea y para todo lo demás, existe una comunidad de desarrolladores que da soporte a cualquier persona.

Sumado a todo esto, tenemos un buen entorno de desarrollo como es eclipse que se integra completamente con el kit de desarrollo de ANDROID, y que permite realizar el autocompletado de código, que facilita mucho la tarea cuando no se conoce en profundidad el lenguaje de programación.

### ● **MUCHA DOCUMENTACIÓN Y EXCELENTE COMUNIDAD DE DESARROLLADORES**

La documentación oficial de ANDROID que está disponible para los desarrolladores es muy abundante y de buena calidad. El desarrollador de ANDROID dispone de mucha información, desde aspectos básicos de desarrollo, hasta detalles muy concretos de la plataforma. Además, casi todo está documentado con ejemplos, con lo que la adaptación a la plataforma es muy rápida por parte del desarrollador.

Adicionalmente, por si toda la información oficial de ANDROID no fuera suficiente, existe una gran comunidad de desarrolladores que, de forma totalmente altruista, dan soporte a los problemas que se puedan presentar a cualquier desarrollador, ya sea principiante o experto.

### ● **SOPORTADO POR UN GRAN FABRICANTE DE SOFTWARE Y UNA GRAN ALIANZA DE EMPRESAS DE TELEFONÍA Y FABRICACIÓN DE DISPOSITIVOS MÓVILES.**

ANDROID es una plataforma software móvil comprada por GOOGLE en 2005. Desde entonces, su desarrollo ha estado a cargo de dicha compañía de software. Por tanto, estamos ante una plataforma que tiene todas las garantías de éxito que aporta GOOGLE a todos sus productos.

Por si no fuera suficiente, se ha formado una gran alianza alrededor de este sistema operativo para móviles, como es la OPEN HANDSET ALLIANCE, entre cuyos miembros destacan T-MOBILE, TELEFÓNICA, VODAFONE, ACER, ASUS, HTC, LG, SAMSUNG, DELL, INTEL y NVIDIA.

## ● RENDIMIENTO POBRE DEL EMULADOR COMPARADO CON UN DISPOSITIVO REAL

El entorno de desarrollo que Google pone a disposición de los desarrolladores incluye una herramienta que permite emular las distintas versiones que se van lanzando del sistema operativo ANDROID. Este emulador permite probar las aplicaciones desarrolladas en un entorno muy parecido al real antes de instalarlas en un dispositivo físico. Sin embargo, durante este desarrollo se ha observado que el rendimiento, en cuanto a la capacidad de procesamiento, está muy lejos de representar la realidad..

## 15. CONSECUCIÓN DE OBJETIVOS

Para comenzar con los objetivos logrados, se puede afirmar que se han alcanzado los objetivos planteados para este proyecto software, desarrollando una aplicación Android que está completamente sincronizada con otra aplicación, en nuestro caso Web.

Cumple el objetivo para el que fue diseñada, que no es otro que poder conocer en cada momento los archivos PDF que se ha subido al servidor, optimizado para un dispositivo móvil.

En el proyecto se mantienen los principios de sencillez de manejo y claridad en la interfaz de usuario y adaptación a los distintos terminales ANDROID existentes. Personalmente creo que la utilidad de una herramienta se mide por la sencillez de su uso, de nada sirve tener algo que resuelva un problema si no se sabe usarlo.

Además, en todo momento se ha tenido en cuenta la importancia de la modularidad y escalabilidad como vía para conseguir una herramienta de larga duración y éxito.

### 15.1 Adquisición y aplicación de conocimientos

Para poder realizar el proyecto, resultó indispensable aplicar los conocimientos que se adquirieron durante el desarrollo de mi anterior proyecto (Ingeniería Técnica Informática de Gestión) ECULTURAL, para poder desarrollar de una forma rápida y eficiente la parte de la aplicación Web.

Antes de comenzar este proyecto, mi conocimiento sobre ANDROID era nulo, no solo sobre esta plataforma, sino también del lenguaje sobre el que está basado, que es JAVA. Por este motivo, tuve que investigar y aprender sobre el uso del lenguaje y de la plataforma de forma simultánea.

Durante el desarrollo del proyecto, la utilización de conocimientos adquiridos en las distintas asignaturas de la carrera se fue sucediendo. Entre los conocimientos destacados están el lenguaje de modelado UML, el diseño y creación de bases de datos, la base de la programación orientada a objetos, la gestión de un proyecto, etc.

### 15.2 POSIBLES AMPLIACIONES

- Mejora de las notificaciones, como por ejemplo, que cuando pinches en la notificación, esta te lleve a la aplicación. También sería interesante añadirle algún atributo más como vibración o personalizarla más.
- Poder hacer una gestión más extensa de los usuarios en la parte de la aplicación Web, añadir o borrar usuarios a la base de datos.



- Poder hacer una gestión de recursos de la interfaz más extensa, en la parte de la aplicación Android. Esto podría ser, redimensionar la pantalla en función del dispositivo en el que se está visualizando o tener en cuenta la orientación de la pantalla, ya esté en vertical u horizontal.
- Añadir eventos *Drag&Drop* en la aplicación Android. Los gestos son cada vez más comunes en los dispositivos táctiles, permitiendo comportarnos, en nuestra interacción con el dispositivo, de una forma más natural.
- Poder subir archivos PDF desde la aplicación Android, siempre y cuando sea el usuario profesor el que pretenda hacerlo.
- Recordar Contraseña cuando un Usuario no recuerda su contraseña al acceder vía email. Tanto en la aplicación Web como en la de Android.
- Posibilidad de enviar mails al profesor de la asignatura, desde la aplicación Web o desde la aplicación de Android.

## 16. BIBLIOGRAFÍA

### 16.1 LIBROS

“Desarrollo de aplicaciones para ANDROID”, Roberto Montero Miguel Ed. Ra-Ma.

“Curso ANDROID. Desarrollo de aplicaciones móviles”, Maestros del web.

“Curso programación ANDROID”, Salvador Gómez Oliver. [www.sgolvier.net](http://www.sgolvier.net)

### 16.2 PÁGINAS WEB

<http://developer.android.com/index.html>

<http://www.sgoliver.net/blog/>

<http://www.aprendeandroid.com/menu.htm>

<http://www.slideshare.net/dcastacun/manual-programacin-android>

<http://www.detodoprogramacion.com/search/label/Android>

<http://www.androidhive.info/>



## 17. ANEXOS

### 17.1 GLOSARIO DE TÉRMINOS

#### ● **ACTIVITY**

Desde la comunidad ANDROID se define una activity como una tarea simple que el usuario puede hacer y en la que el usuario centra su atención. En otras palabras, una activity un medio de interacción con el usuario y es la encargada de presentar la interfaz de usuario y responder a los posibles eventos que se generen. Las activities suelen presentar la interfaz al usuario en forma de pantalla completa pero también pueden ser usadas como ventanas flotantes y pantallas embebidas en otras activities.

#### ● **BBDD O BD**

Abreviatura bases de datos. Se define una base de datos como una colección estructura de datos entre los que existen relaciones lógicas y ha sido diseñada para satisfacer los requerimientos de información de una empresa u organización.

#### ● **BROADCASTING**

Broadcast, difusión en español, es un modo de transmisión de información donde un nodo emisor envía información a una multitud de nodos receptores de manera simultánea, sin necesidad de reproducir la misma transmisión nodo por nodo.

#### ● **CONTEXT**

En ANDROID, se trata de un tipo de objeto que contiene información acerca del entorno de ejecución de la aplicación. A partir del contexto, es posible acceder a los recursos y clases específicas de la aplicación y realizar llamadas a operaciones de nivel de aplicación, como lanzar nuevas activities y recibir o realizar broadcasting de intents.

## ● INTENT

Un Intent se define como una descripción abstracta de una operación a realizar. Se trata de un mecanismo que proporciona al desarrollador la facilidad para realizar enlaces en tiempo de ejecución entre el código de diferentes aplicaciones. Su uso más importante está en el inicio de actividades, que se puede considerar como la unión entre actividades. Se trata básicamente de una estructura de datos pasiva que contiene una descripción abstracta de una acción a realizar. Las unidades principales de la información en un Intent son:

- 📌 Acción → La acción a realizar.
- 📌 Datos → Los datos para operar que requiera la acción a la que se invoca.

## ● LAYOUT

En ANDROID, un layout es el elemento que contiene la definición y jerarquía de las vistas (interfaz de usuario). Los layouts están definidos en archivos con formato XML (eXtensible Markup Language), la razón para usar este lenguaje es que ofrece una estructura legible para el diseño. Cada elemento en el XML del layout es un objeto View o ViewGroup . Los objetos de la vista son las hojas en el árbol, mientras que los objetos ViewGroup son las ramas del árbol.

## ● SQL O *STRUCTURED QUERY LANGUAGE* (LENGUAJE ESTRUCTURADO DE CONSULTA)

Se trata de un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas con el fin de recuperar, de una forma sencilla, información de interés de una base de datos, así como también hacer cambios sobre ella. Es un lenguaje de cuarta generación (4GL).

## 17.2 LICENCIA DEL PROYECTO

Se ha decidido que el proyecto se distribuya bajo licencia GNU GPL (General Public License) versión 3. Se puede consultar la licencia completa bajo estas líneas o desde <http://www.gnu.org/licenses/gpl-3.0.html>.

### GNU GENERAL PUBLIC LICENSE V3

Copyright (C) 2007 Free Software Foundation, Inc. <<http://fsf.org/>>

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

#### Preamble

The GNU General Public License is a free, copyleft license for software and other kinds of works.

The licenses for most software and other practical works are designed to take away your freedom to share and change the works. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change all versions of a program-- to make sure it remains free software for all its users. We, the Free Software Foundation, use the GNU General Public License for most of our software; it applies also to any other work released this way by its authors. You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for them if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs, and that you know you can do these things.

To protect your rights, we need to prevent others from denying you these rights or asking you to surrender the rights. Therefore, you have certain responsibilities if you distribute copies of the software, or if you modify it: responsibilities to respect the freedom of others.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must pass on to the recipients the same freedoms that you received. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

Developers that use the GNU GPL protect your rights with two steps:

(1) assert copyright on the software, and (2) offer you this License giving you legal permission to copy, distribute and/or modify it.

For the developers' and authors' protection, the GPL clearly explains that there is no warranty for this free software. For both users' and authors' sake, the GPL requires that

modified versions be marked as changed, so that their problems will not be attributed erroneously to authors of previous versions.

Some devices are designed to deny users access to install or run modified versions of the software inside them, although the manufacturer can do so. This is fundamentally incompatible with the aim of protecting users' freedom to change the software. The systematic pattern of such abuse occurs in the area of products for individuals to use, which is precisely where it is most unacceptable. Therefore, we have designed this version of the GPL to prohibit the practice for those products. If such problems arise substantially in other domains, we stand ready to extend this provision to those domains in future versions of the GPL, as needed to protect the freedom of users.

Finally, every program is threatened constantly by software patents. States should not allow patents to restrict development and use of software on general-purpose computers, but in those that do, we wish to avoid the special danger that patents applied to a free program could make it effectively proprietary. To prevent this, the GPL assures that patents cannot be used to render the program non-free.

The precise terms and conditions for copying, distribution and modification follow.

## **TERMS AND CONDITIONS**

### **0. Definitions.**

"This License" refers to version 3 of the GNU General Public License.

"Copyright" also means copyright-like laws that apply to other kinds of works, such as semiconductor masks.

"The Program" refers to any copyrightable work licensed under this License. Each licensee is addressed as "you". "Licensees" and "recipients" may be individuals or organizations.

To "modify" a work means to copy from or adapt all or part of the work in a fashion requiring copyright permission, other than the making of an exact copy. The resulting work is called a "modified version" of the earlier work or a work "based on" the earlier work.

A "covered work" means either the unmodified Program or a work based on the Program.

To "propagate" a work means to do anything with it that, without permission, would make you directly or secondarily liable for infringement under applicable copyright law, except executing it on a computer or modifying a private copy. Propagation includes copying, distribution (with or without modification), making available to the public, and in some countries other activities as well.

To "convey" a work means any kind of propagation that enables other parties to make or receive copies. Mere interaction with a user through a computer network, with no transfer of a copy, is not conveying.

An interactive user interface displays "Appropriate Legal Notices" to the extent that it includes a convenient and prominently visible feature that (1) displays an appropriate copyright notice, and (2) tells the user that there is no warranty for the work (except to the extent that warranties are provided), that licensees may convey the work under this License, and how to view a copy of this License. If the interface presents a list of user commands or options, such as a menu, a prominent item in the list meets this criterion.

### **1. Source Code.**

The "source code" for a work means the preferred form of the work for making modifications to it. "Object code" means any non-source form of a work.

A "Standard Interface" means an interface that either is an official standard defined by a recognized standards body, or, in the case of interfaces specified for a particular programming language, one that is widely used among developers working in that language.

The "System Libraries" of an executable work include anything, other than the work as a whole, that (a) is included in the normal form of packaging a Major Component, but which is not part of that Major Component, and (b) serves only to enable use of the work with that Major Component, or to implement a Standard Interface for which an implementation is available to the public in source code form. A "Major Component", in this context, means a major essential component (kernel, window system, and so on) of the specific operating system (if any) on which the executable work runs, or a compiler used to produce the work, or an object code interpreter used to run it.

The "Corresponding Source" for a work in object code form means all the source code needed to generate, install, and (for an executable work) run the object code and to modify the work, including scripts to control those activities. However, it does not include the work's System Libraries, or general-purpose tools or generally available free programs which are used unmodified in performing those activities but which are not part of the work. For example, Corresponding Source includes interface definition files associated with source files for the work, and the source code for shared libraries and dynamically linked subprograms that the work is specifically designed to require, such as by intimate data communication or control flow between those subprograms and other parts of the work.

The Corresponding Source need not include anything that users can regenerate automatically from other parts of the Corresponding Source.

The Corresponding Source for a work in source code form is that same work.



## **2. Basic Permissions.**

All rights granted under this License are granted for the term of copyright on the Program, and are irrevocable provided the stated conditions are met. This License explicitly affirms your unlimited permission to run the unmodified Program. The output from running a covered work is covered by this License only if the output, given its content, constitutes a covered work. This License acknowledges your rights of fair use or other equivalent, as provided by copyright law.

You may make, run and propagate covered works that you do not convey, without conditions so long as your license otherwise remains in force. You may convey covered works to others for the sole purpose of having them make modifications exclusively for you, or provide you with facilities for running those works, provided that you comply with the terms of this License in conveying all material for which you do not control copyright. Those thus making or running the covered works

for you must do so exclusively on your behalf, under your direction and control, on terms that prohibit them from making any copies of your copyrighted material outside their relationship with you.

Conveying under any other circumstances is permitted solely under the conditions stated below. Sublicensing is not allowed; section 10 makes it unnecessary.

## **3. Protecting Users' Legal Rights From Anti-Circumvention Law.**

No covered work shall be deemed part of an effective technological measure under any applicable law fulfilling obligations under article 11 of the WIPO copyright treaty adopted on 20 December 1996, or similar laws prohibiting or restricting circumvention of such measures.

When you convey a covered work, you waive any legal power to forbid circumvention of technological measures to the extent such circumvention is effected by exercising rights under this License with respect to the covered work, and you disclaim any intention to limit operation or modification of the work as a means of enforcing, against the work's users, your or third parties' legal rights to forbid circumvention of technological measures.

## **4. Conveying Verbatim Copies.**

You may convey verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice; keep intact all notices stating that this License and any non-permissive terms added in accord with section 7 apply to the code; keep intact all notices of the absence of any warranty; and give all recipients a copy of this License along with the Program.

You may charge any price or no price for each copy that you convey, and you may offer support or warranty protection for a fee.

### **5. Conveying Modified Source Versions.**

You may convey a work based on the Program, or the modifications to produce it from the Program, in the form of source code under the terms of section 4, provided that you also meet all of these conditions:

- a) The work must carry prominent notices stating that you modified it, and giving a relevant date.
- b) The work must carry prominent notices stating that it is released under this License and any conditions added under section

### **6. This requirement modifies the requirement in section 4 to "keep intact all notices".**

- c) You must license the entire work, as a whole, under this License to anyone who comes into possession of a copy. This License will therefore apply, along with any applicable section 7 additional terms, to the whole of the work, and all its parts, regardless of how they are packaged. This License gives no permission to license the work in any other way, but it does not invalidate such permission if you have separately received it.
- d) If the work has interactive user interfaces, each must display Appropriate Legal Notices; however, if the Program has interactive interfaces that do not display Appropriate Legal Notices, your work need not make them do so.

A compilation of a covered work with other separate and independent works, which are not by their nature extensions of the covered work, and which are not combined with it such as to form a larger program, in or on a volume of a storage or distribution medium, is called an "aggregate" if the compilation and its resulting copyright are not used to limit the access or legal rights of the compilation's users beyond what the individual works permit. Inclusion of a covered work in an aggregate does not cause this License to apply to the other parts of the aggregate.

### **7. Conveying Non-Source Forms.**

You may convey a covered work in object code form under the terms of sections 4 and 5, provided that you also convey the machine-readable Corresponding Source under the terms of this License, in one of these ways:

- a) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by the Corresponding Source fixed on a durable physical medium customarily used for software interchange.

b) Convey the object code in, or embodied in, a physical product (including a physical distribution medium), accompanied by a written offer, valid for at least three years and valid for as long as you offer spare parts or customer support for that product model, to give anyone who possesses the object code either (1) a copy of the Corresponding Source for all the software in the product that is covered by this License, on a durable physical medium customarily used for software interchange, for a price no more than your reasonable cost of physically performing this conveying of source, or (2) access to copy the Corresponding Source from a network server at no charge.

c) Convey individual copies of the object code with a copy of the written offer to provide the Corresponding Source. This alternative is allowed only occasionally and noncommercially, and only if you received the object code with such an offer, in accord with subsection 6b.

d) Convey the object code by offering access from a designated place (gratis or for a charge), and offer equivalent access to the Corresponding Source in the same way through the same place at no further charge. You need not require recipients to copy the Corresponding Source along with the object code. If the place to copy the object code is a network server, the Corresponding Source may be on a different server (operated by you or a third party) that supports equivalent copying facilities, provided you maintain clear directions next to the object code saying where to find the Corresponding Source.

Regardless of what server hosts the Corresponding Source, you remain obligated to ensure that it is available for as long as needed to satisfy these requirements.

e) Convey the object code using peer-to-peer transmission, provided you inform other peers where the object code and Corresponding Source of the work are being offered to the general public at no charge under subsection 6d.

A separable portion of the object code, whose source code is excluded from the Corresponding Source as a System Library, need not be included in conveying the object code work.

A "User Product" is either (1) a "consumer product", which means any tangible personal property which is normally used for personal, family, or household purposes, or (2) anything designed or sold for incorporation into a dwelling. In determining whether a product is a consumer product, doubtful cases shall be resolved in favor of coverage. For a particular product received by a particular user, "normally used" refers to a typical or common use of that class of product, regardless of the status of the particular user or of the way in which the particular user actually uses, or expects or is expected to use, the product. A product is a consumer product regardless of whether the product has substantial commercial, industrial or non-consumer uses, unless such uses represent the only significant mode of use of the product.

"Installation Information" for a User Product means any methods, procedures, authorization keys, or other information required to install and execute modified

versions of a covered work in that User Product from a modified version of its Corresponding Source. The information must suffice to ensure that the continued functioning of the modified object code is in no case prevented or interfered with solely because modification has been made.

If you convey an object code work under this section in, or with, or specifically for use in, a User Product, and the conveying occurs as part of a transaction in which the right of possession and use of the User Product is transferred to the recipient in perpetuity or for a fixed term (regardless of how the transaction is characterized), the

Corresponding Source conveyed under this section must be accompanied by the Installation Information. But this requirement does not apply if neither you nor any third party retains the ability to install modified object code on the User Product (for example, the work has been installed in ROM).

The requirement to provide Installation Information does not include a requirement to continue to provide support service, warranty, or updates for a work that has been modified or installed by the recipient, or for the User Product in which it has been modified or installed. Access to a network may be denied when the modification itself materially and adversely affects the operation of the network or violates the rules and protocols for communication across the network.

Corresponding Source conveyed, and Installation Information provided, in accord with this section must be in a format that is publicly documented (and with an implementation available to the public in source code form), and must require no special password or key for unpacking, reading or copying.

## **8. Additional Terms.**

"Additional permissions" are terms that supplement the terms of this License by making exceptions from one or more of its conditions. Additional permissions that are applicable to the entire Program shall be treated as though they were included in this License, to the extent that they are valid under applicable law. If additional permissions apply only to part of the Program, that part may be used separately under those permissions, but the entire Program remains governed by this License without regard to the additional permissions.

When you convey a copy of a covered work, you may at your option remove any additional permissions from that copy, or from any part of it. (Additional permissions may be written to require their own removal in certain cases when you modify the work.) You may place additional permissions on material, added by you to a covered work, for which you have or can give appropriate copyright permission.

Notwithstanding any other provision of this License, for material you add to a covered work, you may (if authorized by the copyright holders of that material) supplement the terms of this License with terms:

- a) Disclaiming warranty or limiting liability differently from the terms of sections 15 and 16 of this License; or
- b) Requiring preservation of specified reasonable legal notices or author attributions in that material or in the Appropriate Legal Notices displayed by works containing it; or
- c) Prohibiting misrepresentation of the origin of that material, or requiring that modified versions of such material be marked in reasonable ways as different from the original version; or
- d) Limiting the use for publicity purposes of names of licensors or authors of the material; or
- e) Declining to grant rights under trademark law for use of some trade names, trademarks, or service marks; or
- f) Requiring indemnification of licensors and authors of that material by anyone who conveys the material (or modified versions of it) with contractual assumptions of liability to the recipient, for any liability that these contractual assumptions directly impose on those licensors and authors.

All other non-permissive additional terms are considered "further restrictions" within the meaning of section 10. If the Program as you received it, or any part of it, contains a notice stating that it is governed by this License along with a term that is a further restriction, you may remove that term. If a license document contains a further restriction but permits relicensing or conveying under this License, you may add to a covered work material governed by the terms of that license document, provided that the further restriction does not survive such relicensing or conveying.

If you add terms to a covered work in accord with this section, you must place, in the relevant source files, a statement of the additional terms that apply to those files, or a notice indicating where to find the applicable terms.

Additional terms, permissive or non-permissive, may be stated in the form of a separately written license, or stated as exceptions; the above requirements apply either way.

## **9. Termination.**

You may not propagate or modify a covered work except as expressly provided under this License. Any attempt otherwise to propagate or modify it is void, and will automatically terminate your rights under this License (including any patent licenses granted under the third paragraph of section 11).

However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright

holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, you do not qualify to receive new licenses for the same material under section 10.

### **10. Acceptance Not Required for Having Copies.**

You are not required to accept this License in order to receive or run a copy of the Program. Ancillary propagation of a covered work occurring solely as a consequence of using peer-to-peer transmission to receive a copy likewise does not require acceptance. However, nothing other than this License grants you permission to propagate or modify any covered work. These actions infringe copyright if you do not accept this License. Therefore, by modifying or propagating a covered work, you indicate your acceptance of this License to do so.

### **11. Automatic Licensing of Downstream Recipients.**

Each time you convey a covered work, the recipient automatically receives a license from the original licensors, to run, modify and propagate that work, subject to this License. You are not responsible for enforcing compliance by third parties with this License.

An "entity transaction" is a transaction transferring control of an organization, or substantially all assets of one, or subdividing an organization, or merging organizations. If propagation of a covered work results from an entity transaction, each party to that transaction who receives a copy of the work also receives whatever licenses to the work the party's predecessor in interest had or could give under the previous paragraph, plus a right to possession of the Corresponding Source of the work from the predecessor in interest, if the predecessor has it or can get it with reasonable efforts.

You may not impose any further restrictions on the exercise of the rights granted or affirmed under this License. For example, you may not impose a license fee, royalty, or other charge for exercise of rights granted under this License, and you may not initiate litigation (including a cross-claim or counterclaim in a lawsuit) alleging that any patent claim is infringed by making, using, selling, offering for sale, or importing the Program or any portion of it.

## 12. Patents.

A "contributor" is a copyright holder who authorizes use under this License of the Program or a work on which the Program is based. The work thus licensed is called the contributor's "contributor version".

A contributor's "essential patent claims" are all patent claims owned or controlled by the contributor, whether already acquired or hereafter acquired, that would be infringed by some manner, permitted by this License, of making, using, or selling its contributor version, but do not include claims that would be infringed only as a consequence of further modification of the contributor version. For purposes of this definition, "control" includes the right to grant patent sublicenses in a manner consistent with the requirements of this License.

Each contributor grants you a non-exclusive, worldwide, royalty-free patent license under the contributor's essential patent claims, to make, use, sell, offer for sale, import and otherwise run, modify and propagate the contents of its contributor version.

In the following three paragraphs, a "patent license" is any express agreement or commitment, however denominated, not to enforce a patent (such as an express permission to practice a patent or covenant not to sue for patent infringement). To "grant" such a patent license to a party means to make such an agreement or commitment not to enforce a patent against the party.

If you convey a covered work, knowingly relying on a patent license, and the Corresponding Source of the work is not available for anyone to copy, free of charge and under the terms of this License, through a publicly available network server or other readily accessible means, then you must either (1) cause the Corresponding Source to be so available, or (2) arrange to deprive yourself of the benefit of the patent license for this particular work, or (3) arrange, in a manner consistent with the requirements of this License, to extend the patent license to downstream recipients. "Knowingly relying" means you have actual knowledge that, but for the patent license, your conveying the covered work in a country, or your recipient's use of the covered work in a country, would infringe one or more identifiable patents in that country that you have reason to believe are valid.

If, pursuant to or in connection with a single transaction or arrangement, you convey, or propagate by procuring conveyance of, a covered work, and grant a patent license to some of the parties receiving the covered work authorizing them to use, propagate, modify or convey a specific copy of the covered work, then the patent license you grant is automatically extended to all recipients of the covered work and works based on it.

A patent license is "discriminatory" if it does not include within the scope of its coverage, prohibits the exercise of, or is conditioned on the non-exercise of one or more of the rights that are specifically granted under this License. You may not convey a covered work if you are a party to an arrangement with a third party that is in the

business of distributing software, under which you make payment to the third party based on the extent of your activity of conveying the work, and under which the third party grants, to any of the parties who would receive the covered work from you, a discriminatory patent license (a) in connection with copies of the covered work conveyed by you (or copies made from those copies), or (b) primarily for and in connection with specific products or compilations that contain the covered work, unless you entered into that arrangement, or that patent license was granted, prior to 28 March 2007.

Nothing in this License shall be construed as excluding or limiting any implied license or other defenses to infringement that may otherwise be available to you under applicable patent law.

### **13. No Surrender of Others' Freedom.**

If conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot convey a covered work so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not convey it at all. For example, if you agree to terms that obligate you to collect a royalty for further conveying from those to whom you convey the Program, the only way you could satisfy both those terms and this License would be to refrain entirely from conveying the Program.

### **14. Use with the GNU Affero General Public License.**

Notwithstanding any other provision of this License, you have permission to link or combine any covered work with a work licensed under version 3 of the GNU Affero General Public License into a single combined work, and to convey the resulting work. The terms of this License will continue to apply to the part which is the covered work, but the special requirements of the GNU Affero General Public License, section 13, concerning interaction through a network will apply to the combination as such.

### **15. Revised Versions of this License.**

The Free Software Foundation may publish revised and/or new versions of the GNU General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies that a certain numbered version of the GNU General Public License "or any later version" applies to it, you have the option of following the terms and conditions either of that numbered version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of the GNU General Public License, you may choose any version ever published by the Free Software Foundation.



If the Program specifies that a proxy can decide which future versions of the GNU General Public License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Program.

Later license versions may give you additional or different permissions. However, no additional obligations are imposed on any author or copyright holder as a result of your choosing to follow a later version.

#### **16. Disclaimer of Warranty.**

THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

#### **17. Limitation of Liability.**

IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MODIFIES AND/OR CONVEYS THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

#### **18. Interpretation of Sections 15 and 16.**

If the disclaimer of warranty and limitation of liability provided above cannot be given local legal effect according to their terms, reviewing courts shall apply local law that most closely approximates an absolute waiver of all civil liability in connection with the Program, unless a warranty or assumption of liability accompanies a copy of the Program in return for a fee.

END OF TERMS AND CONDITIONS