



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Desarrollo de aplicaciones web para dispositivos móviles con JQuery mobile

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Jose Manuel Casielles Bañón

**Tutor:** Germán Vidal

2014-2015



# Resumen

El proyecto consiste en el desarrollo de una aplicación para dispositivos móviles (Smartphones y tablets).

Mediante esta aplicación, CasiTeka, el usuario podrá utilizar su dispositivo móvil para acceder a la biblioteca online de la UPV y gestionar la reserva de todo el material disponible en ella. Además incluye integración con redes sociales para compartir contenido.

A lo largo de la memoria se explicará el contexto de las aplicaciones móviles y de los sistemas Android, IOS, Blackberry y Windows Phone. Además de los conceptos y herramientas aplicados durante el desarrollo, de cara a resultar lo más didáctico posible.

**Palabras clave:** móvil, biblioteca, redes sociales, aplicaciones.

# Abstract

The project involves the development of an application for mobile devices (Smartphones and tablets).

Using this application, Casiteka, the user can use his mobile device to access the online library of the UPV and manage reservations of all the material available on it. It also includes integration with social networks to share content.

Throughout memory the context of mobile applications and Android, iOS, BlackBerry an Windows Phone systems are explained. Besides the concepts and tools applied during development will be explained, in order to be as didactic as possible.

**Keywords :** mobile, library, social networks, applications.



# Tabla de contenidos

1.	Introducción.....	8
1.1.	SITUACION.....	8
1.2.	MOTIVACION.....	9
1.3.	ALCANCE.....	9
1.4.	OBJETIVO.....	10
1.5.	METODOLOGÍA DE DESARROLLO.....	10
1.6.	ORGANIZACIÓN DE LA MEMORIA.....	11
2.	Análisis de requisitos.....	12
2.1.	ANALISIS INICIAL.....	12
2.2.	ANALISIS DE REQUISITOS.....	12
2.3.	ANALISIS DE STAKEHOLDERS.....	15
2.4.	ANALISIS TECNOLOGICO.....	19
2.4.1	Análisis de sistemas operativos.....	20
2.4.2	Elección de Sistema Operativo.....	23
2.4.3	Análisis de aplicaciones.....	24
2.4.4	Elección de aplicación.....	27
3.	Especificación del sistema.....	29
3.1.	CASOS DE USO.....	29
3.1.1	Actores.....	30
3.1.2	Diagramas de casos de uso.....	30
3.1.3	Descripción de los casos de uso.....	31
3.2.	MODELO CONCEPTUAL.....	37
3.2.1	Descripción del modelo.....	37
4.	Diseño.....	39
4.1.	ARQUITECTURA FÍSICA.....	40
4.2.	TECNOLOGÍAS UTILIZADAS.....	41
4.2.1	One.com.....	41
4.2.2	JQuery Mobile [16].....	43
4.2.3	Comunicaciones.....	48
4.3.	INTERFAZ GRÁFICA.....	51
4.3.1	Mapa de navegación.....	57
4.4.	GESTIÓN DE DATOS.....	58



4.4.1 Documentos .....	58
4.4.2 Usuarios.....	59
4.4.3 Préstamos .....	59
4.4.4 Comentarios .....	59
4.4.5 Peticiones.....	60
4.4.6 Bibliotecas .....	60
5. Implementación.....	61
5.1. CONFIGURACIÓN INICIAL.....	61
5.1.1 Creación de contenido .....	62
5.2. COMUNICACIÓN ENTRE PÁGINAS.....	63
5.3. FUNCIONALIDADES PRINCIPALES.....	64
5.3.1 Inicio de sesión .....	65
5.3.2 Listar documentos.....	65
5.3.3 Reservar documento .....	67
5.3.4 Descargar documento .....	68
5.3.5 Recomendar documento.....	68
5.3.6 Opinar sobre un documento.....	69
5.3.7 Información sobre bibliotecas .....	69
5.3.7 Contacto .....	71
6. Pruebas .....	72
6.1. INICIO DE SESIÓN.....	72
6.2. RESERVAR DOCUMENTO.....	73
6.3. BUSCAR DOCUMENTO.....	73
6.4. DEMANDAR DOCUMENTO .....	73
6.5. CANCELAR RESERVA.....	73
6.6. RECOMENDAR DOCUMENTO .....	73
6.7. OPINAR SOBRE UN DOCUMENTO.....	74
6.8. CONSULTAR BIBLIOTECA.....	74
6.9. CONTACTO .....	74
7. Conclusiones.....	75
7.1. REVISIÓN DE LOS OBJETIVOS.....	75
7.2. PROBLEMAS EN EL DESARROLLO.....	76
7.2.1 Uso de PHP .....	77
7.2.2 Configuración servidor .....	77
7.2.3 Mantener sesión de usuarios y comunicación entre páginas .....	77
7.2.4 Envío de mails en sección contacto.....	78



7.3.	MEJORAS.....	78
7.4.	VALORACIÓN.....	79
8.	Bibliografía.....	81
8.1.	Bibliografía referenciada .....	81
8.2.	Bibliografía no referenciada .....	82
9.	Apéndice.....	84
9.1	Descripción BBDD.....	84
9.1.1.-	ANÁLISIS DE REQUISITOS Y DESCRIPCIÓN DEL SISTEMA.....	84
9.1.2.-	ENTIDADES Y ATRIBUTOS .....	85
9.1.3.-	DIAGRAMA ENTIDAD-RELACIÓN .....	86
9.1.4.-	DISEÑO LÓGICO.....	86
9.1.5.-	CREACIÓN DE TABLAS.....	89
9.1.6.-	INSERCCIÓN DE LOS DATOS .....	91



# 1. Introducción

Este documento pretende materializar el trabajo realizado durante el desarrollo del Trabajo de Fin de Grado **Casiteka**. Se ha realizado la documentación lo más realista posible, así como la descripción adecuada de cada una de las fases por las que pasa un proyecto software de forma habitual.

## 1.1. SITUACION

Actualmente vivimos en una época caracterizada por el rápido avance de las tecnologías que ha conllevado a una mejoría de todas las herramientas tecnológicas, ya sea en ámbito laboral, social, estudiantil.... Concretamente nos importa centrarnos en los avances de la tecnología móvil y todo lo que esto conlleva. Hemos pasado en apenas unos 10 años de usar un teléfono fijo para llamar a nuestros conocidos a usar un teléfono móvil para monitorizar nuestra localización mientras salimos a realizar actividad física y así obtener datos para mejorar la próxima vez. Este solo es un ejemplo de todo lo que podemos hacer y un aliciente para motivarnos de todo lo que aún nos queda por poder hacer con los móviles.

Actualmente los teléfonos móviles ya son conocidos como teléfonos inteligentes o “smartphones”. Estos smartphones han pasado de realizar solo llamadas y enviar sms a ser capaces de sustituir a ordenadores en algunos casos. La mayoría de la gente utiliza los smartphones antes que los ordenadores para ver todas sus redes sociales y ponerse en contacto con sus conocidos mediante aplicaciones que hacen uso de internet. Además de estas aplicaciones podemos hacer fotos, jugar a videojuegos, monitorizar actividades físicas, consultar el tiempo, navegar por internet....

Como se puede observar el Smartphone forma parte de la vida diaria de cualquier persona y un estudio en España desarrollado por Telefónica indica que el 81 % de los móviles en España son Smartphones [1]. Esto conlleva que el mercado de las aplicaciones para móviles sea un mercado con muchos usuarios y ofrece más garantías de éxito que cualquier otro mercado de aplicaciones en estos momentos.

## 1.2. MOTIVACION

En esta época siempre se busca integrar las nuevas tecnologías con la vida diaria y es por ello por lo que se desarrollan multitud de aplicaciones para smartphones que ayuden a las personas a desarrollar estas actividades. Aplicaciones como calendarios, guías de cocina, reproductores de música... nos permiten desarrollar la mayoría de cosas que hacemos en un día normal con la ayuda de un único dispositivo.

Por eso surge la necesidad también de integrar los estudios con las facilidades que otorga un Smartphone y de ahí surge la idea de la **Casiteka**. Una aplicación móvil que permite la reserva de contenido de lectura disponible en bibliotecas al instante o la descarga de contenido multimedia (audios, libros, revistas, presentaciones...). Además de ofrecer esto al estudiante se busca una integración entre los libros y documentos de lectura con las nuevas tecnologías para hacerlos más llamativos a los jóvenes, que cada vez optan más por el uso de ver vídeos o leer pdf en sus dispositivos en lugar de reservar libros.

Esta herramienta debe ser sencilla y fácil de usar ya que las pantallas táctiles tienen un tamaño reducido y no se puede poner mucha información en la ventana que se muestra al usuario.

Para desarrollar la **Casiteka** se necesita tener una herramienta fácil de utilizar y de gestionar, por eso se ha pensado en que la solución ideal es el desarrollo de una aplicación móvil basada en JQuery Mobile. Junta toda la funcionalidad de JQuery pero centrado en una interfaz más minimalista enfocada al uso en plataformas móviles.

## 1.3. ALCANCE

La primera tarea en todo proyecto consiste en definir el alcance del mismo. Esto consiste en establecer las funcionalidades a desarrollar, identificar los distintos “stakeholders”, seleccionar las herramientas más adecuadas para definir y gestionar el proyecto y realizar un desglose del proyecto y sus entregables.

La **Casiteka** consistiría en una aplicación móvil que gestionaría todas las bibliotecas de la UPV. Con esto se pretende que los usuarios puedan reservar los libros de estas bibliotecas, consultar información sobre las bibliotecas, incluso descargar contenido multimedia de las bibliotecas como audios, revistas digitales u otros documentos. Además se pretende integrar las bibliotecas y fomentar el uso de ellas permitiendo a los usuarios expresar su opinión sobre los documentos o compartirlos en las redes sociales.

Vistas estas características distinguimos tres tipos de usuarios característicos: los alumnos, los profesores y el propio personal de la biblioteca. Se pretende que la mayoría de usuarios sean alumnos que busquen apoyo de cara a sus estudios. Por otra

parte los profesores podrán usar la aplicación para compartir con sus alumnos los documentos mediante links directos que podrán enviar incluso por correo electrónico. El rol del personal de biblioteca es el más técnico, ya que serían los encargados de mantener el sistema al día, ya sea manteniendo al día la lista de documentos o actualizando la información de las bibliotecas, gestionando peticiones de documentos....

Para ello, se pretende desarrollar una aplicación web instalada en un servidor con acceso garantizado mediante el dominio [casiteka.com](http://casiteka.com), que permitirá el acceso a la aplicación a todo el personal registrado en la intranet de la UPV. De esta forma el usuario tendría toda la información de su cuenta en la biblioteca en esta aplicación de fácil uso y sencillez visual.

### 1.4. OBJETIVO

El objetivo principal de **Casiteka** es motivar el acceso de los alumnos a las bibliotecas de los campus de la UPV haciendo uso de las tecnologías móviles. Para esto se necesitan alcanzar los siguientes objetivos:

- Desarrollar una aplicación web.
- Optimizar esta aplicación para el uso en móviles.
- Comunicar esta aplicación con una base de datos que disponga de toda la información referente a las bibliotecas.
- Mantener una coherencia con lo almacenado en la base de datos y lo mostrado en la aplicación.

### 1.5. METODOLOGÍA DE DESARROLLO

Para poder desarrollar un proyecto software también es necesario elegir un método de trabajo que nos permita llevar a cabo todo lo planeado.

Hay una gran variedad de modelos de desarrollo software que se podrían dividir en genéricos o ágiles [2]. Los modelos genéricos suelen desarrollar las distintas partes de un proyecto software (análisis de requisitos, especificación, diseño, implementación, pruebas, mantenimiento y documentación) una única vez y cada una durante un periodo de tiempo extenso. Los ágiles sin embargo, dividen el proyecto en fases de unas cuatro semanas en las que se llevan a cabo las distintas partes de un proyecto software a menor escala. De esta forma el proyecto permite más cambios en los requisitos y en el diseño.

Dado que los requisitos de la **Casiteka** no van a sufrir grandes modificaciones con el paso del proyecto se ha considerado oportuno desarrollar una metodología genérica.



Principalmente se basará en la metodología de cascada. El modelo en cascada es un proceso secuencial de desarrollo de las fases de un proyecto software. Permite cierta superposición entre estas fases permitiendo comenzar una poco antes de que acabe la anterior. Un punto fuerte de este modelo es que se invierte gran tiempo en la especificación y diseño software, lo que garantiza mayor éxito a la hora de alcanzar los objetivos y evitar desvíos en la planificación. Lo malo es su rigidez a los casos de uso y la dificultad de cambiar los requisitos una vez ya han pasado su fase.

Finalmente concluimos que este es el modelo que más se adecua a la **Casiteka** ya que los requisitos están claros desde el principio y se tenía muy claro el cómo desarrollarlos.

## 1.6. ORGANIZACIÓN DE LA MEMORIA

La documentación se basa en la siguiente estructura:

En el capítulo de introducción se presenta el proyecto y se definen los conceptos claves de la aplicación y su desarrollo. Además se expone el contexto que motiva a la creación de esta aplicación y los objetivos de esta.

A continuación se realiza un análisis de los elementos importantes a tener en cuenta a la hora de empezar el desarrollo, incluyendo análisis de requisitos, planificación de tiempo detallada y un pequeño análisis de las herramientas disponibles en la actualidad para desarrollar la aplicación.

La siguiente sección consiste en una descripción del diseño y la implementación de la aplicación. Aquí se presentan los diferentes casos de uso las funcionalidades de la aplicación junto con un diagrama de flujo que haga comprender mejor el funcionamiento de la aplicación. Después pasaremos a explicar la fase de diseño de la aplicación dividida en la arquitectura física del sistema, las tecnologías utilizadas para el desarrollo y la arquitectura lógica del sistema. En esta fase de diseño también se presentarán las distintas interfaces desarrolladas junto con una breve explicación de su funcionalidad. Finalmente se documentarán los mayores desafíos a la hora de la implementación junto con las pruebas llevadas a cabo de cada una de las funcionalidades a desarrollar.

Una vez descritas estas etapas del proyecto, se pasará a un análisis final del proyecto y evaluar si lo que se tenía planificado al principio ha podido desarrollarse con satisfacción y sencillez gracias al uso de la tecnología elegida. Junto con todo esto se explica también la experiencia del desarrollador con la herramienta y sus opiniones sobre ella. Como parte extra se presentarán unas pequeñas ideas para aumentar la experiencia del usuario con la aplicación en forma de mejoras para el futuro.

Finalmente, encontramos la bibliografía y unos anexos en los que se describen algunas partes del proyecto como por ejemplo la base de datos y un manual del usuario.



## 2. Análisis de requisitos

### 2.1. ANALISIS INICIAL

En este capítulo está detallado el análisis realizado durante la fase inicial del proyecto. En primer lugar se desarrolló el análisis de requisitos, basados en los objetivos del proyecto y que nos definen el sistema a construir. Posteriormente se realizó un estudio de las tecnologías que se podrían utilizar y así justificar la tecnología que se decidió utilizar.



ILUSTRACIÓN 1: FASES DE DESARROLLO SOFTWARE (ANÁLISIS DE REQUISITOS)

### 2.2. ANALISIS DE REQUISITOS

#### 2.2.1. Requisitos funcionales [3]

<b>Código de requisito</b>	REQ01	<b>Prioridad</b>	Alta / Imprescindible
<b>Nombre</b>	Acceso a los datos de usuario previamente registrado.		
<b>Descripción</b>	Dado que la aplicación será un servicio más ofrecido por la politécnica, esta usará los usuarios ya creados en la intranet de la UPV. De esta forma la aplicación deberá garantizar el acceso a los usuarios a través de sus credenciales de la intranet y a través de ellos gestionar su actividad en la aplicación.		

<b>Código de requisito</b>	REQ02	<b>Prioridad</b>	Alta Imprescindible /
<b>Nombre</b>	Ver información detallada sobre cada documento de la biblioteca.		
<b>Descripción</b>	El usuario una vez dentro de la aplicación deberá poder ver todos los documentos disponibles en la biblioteca junto con toda la información referente a cada uno. Esta información se compone de autor, título, año de publicación, editorial, tipo de documento, páginas, sinopsis, isbn, puntuación y una imagen del documento.		

<b>Código de requisito</b>	REQ03	<b>Prioridad</b>	Alta Imprescindible /
<b>Nombre</b>	Gestión de reservas y descarga de documentos por parte del usuario.		
<b>Descripción</b>	<p>El usuario podrá reservar cualquier documento físico para posteriormente recogerlo en una biblioteca del campus de la UPV. Así mismo podrán descargar cualquier documento que lo permita, estos tipos de documento pueden ser pdfs, presentaciones, audios, vídeos....</p> <p>Además se debe ofrecer al usuario una sección en la que pueda ver de forma clara y sencilla un resumen de todos los documentos que tiene actualmente en su poder y de los que tiene reservado para fechas futuras.</p>		

<b>Código de requisito</b>	REQ04	<b>Prioridad</b>	Media / Deseable
<b>Nombre</b>	Permitir a los usuarios expresar su opinión sobre los documentos.		
<b>Descripción</b>	<p>Sería deseable que cada documento tuviera una sección de comentarios publicados por los usuarios de la aplicación. De esta manera se conseguiría una mayor interacción entre los usuarios y se mostraría más información sobre los documentos.</p> <p>Esta información es más importante que la información genérica, ya que muestra las reacciones de la gente al tratar con el documento y puede facilitar la elección de los demás usuarios que estén interesados en ese documento.</p>		

<b>Código de requisito</b>	REQ05	<b>Prioridad</b>	Media / Deseable
<b>Nombre</b>	Integración de la aplicación con las redes sociales.		
<b>Descripción</b>	<p>Como ya se ha comentado antes la mayoría de personas tiene multitud de redes sociales y gasta una gran parte de su tiempo libre consultándolas.</p> <p>Sería muy importante que se integrará algún mecanismo de recomendación a través de redes sociales de los documentos. De esta forma conseguiríamos más publicidad para la aplicación y suministraríamos a los usuarios un método fácil para recomendar documentos a otros usuarios, ya sea mediante red social o correo electrónico.</p>		

<b>Código de requisito</b>	REQ06	<b>Prioridad</b>	Alta / Imprescindible
<b>Nombre</b>	Integración de la aplicación con las redes sociales.		
<b>Descripción</b>	<p>Como ya se ha comentado antes la mayoría de personas tiene multitud de redes sociales y gasta una gran parte de su tiempo libre consultándolas.</p> <p>Sería muy importante que se integrará algún mecanismo de recomendación a través de redes sociales de los documentos. De esta forma conseguiríamos más publicidad para la aplicación y suministraríamos a los usuarios un método fácil para recomendar documentos a otros usuarios, ya sea mediante red social o correo electrónico.</p>		

<b>Código de requisito</b>	REQ07	<b>Prioridad</b>	Media / Deseable
<b>Nombre</b>	Consultar información sobre las bibliotecas de la UPV.		
<b>Descripción</b>	<p>Otra función necesaria para nuestra aplicación es mostrar una lista con todas las bibliotecas que hay en la UPV. Esto es necesario ya que el usuario deberá pasar por ellas para recoger sus documentos reservados.</p> <p>De esta forma ofrecemos al usuario todos los datos de las bibliotecas: horarios, teléfono, dirección , descripción y hasta un mapa.</p>		

<b>Código de requisito</b>	REQ08	<b>Prioridad</b>	Media / Deseable
<b>Nombre</b>	Disponer de un buzón de sugerencias de los usuarios.		
<b>Descripción</b>	<p>De cara a mejoras y a tener un feedback por parte de los usuarios siempre está bien disponer de un buzón de sugerencias.</p> <p>Para ello se creará una cuenta de email en nuestro dominio. Esta cuenta recibirá todas las dudas y sugerencias que envíen los usuarios a través de la sección diseñada expresamente para esto en la aplicación</p>		

### 2.2.2. Requisitos no funcionales [3]

- Usabilidad: La aplicación debe tener una interfaz simple e intuitiva orientada a las pantallas móviles, además de mostrar mensajes de error al usuario.
- Escalabilidad: La aplicación debe estar bien estructurada para poder añadir funcionalidad en el futuro de forma sencilla.
- Facilidad de pruebas: La aplicación contará con un sistema de facilidades a la hora de encontrar los fallos en las pruebas.
- Validación: Se llevará a cabo comprobación de los campos a la hora de validar formularios.
- Instalación: La aplicación no requerirá ninguna instalación, haciendo al usuario más fácil su acceso.
- Diseño: Uso adecuado de las tecnologías móviles y del doble sentido de la pantalla.
- Compatibilidad con el navegador web genérico de Android así como los demás navegadores diseñados para móviles. No se asegura compatibilidad con navegadores para ordenador.

### 2.3. ANALISIS DE STAKEHOLDERS

Antes de centrarnos en cómo desarrollar los requisitos anteriormente explicado, es necesario desarrollar un pequeño análisis de los “stakeholders” de nuestra aplicación.

Este análisis consiste en analizar todos los interesados que puedan afectar al desarrollo de nuestro proyecto. En proyectos comerciales se estudian desde los usuarios objetivos de nuestra aplicación hasta posibles usuarios que supongan una amenaza para el éxito de la misma.

Dado que nuestra aplicación consistirá en una herramienta adicional para la comunidad universitaria de la UPV, sin ningún ánimo de lucro, hemos considerado que

no hay ninguna amenaza a nivel de usuarios para nuestra aplicación. De esta forma nos centramos en los usuarios potenciales que harán uso de la aplicación.

A continuación se han elaborado tres fichas para cada uno de los usuarios. En función de estas fichas luego se llegará a una conclusión sobre las características principales que deberá tener nuestra aplicación.

## Luis Juan Ortuño – personal principal (profesor)

### Biografía



- 53 años de edad
- Nacido en Alcoy
- Trabaja de profesor de programación orientada a videojuegos en la escuela superior de ingeniería informática en la Universidad Politécnica de Valencia, campus de Alcoy
- Vive en Alcoy con su mujer y dos hijos
- Titulado en ingeniería informática en San Sebastián
- Cobra unos 70000 euros anuales
- Tiene un Ford Mondeo, con el que se desplaza a la universidad cada día

### Objetivos

- Ponérselo fácil a sus alumnos a la hora de estudiar
- Optimizar su tiempo libre para compaginar su vida privada con la de profesor
- Suministrar documentación adicional a los alumnos
- Seguir aprendiendo sobre las nuevas tecnologías

### Tecnología

- Está acostumbrada a usar ordenadores debido a su trabajo
- Tiene un iPhone 5S
- Usa multitud de redes sociales y está al día gracias a ellas
- No tiene problema en adaptarse a nuevas interfaces gráficas
- Le gusta experimentar nuevas tecnologías

### Salud

- Tiene un problema de garganta debido a su adicción a fumar, por lo que usa contenido de audio digital de la biblioteca para recomendárselo a sus alumnos

### ILUSTRACIÓN 2: PERSONA PRINCIPAL - PROFESOR

# Michael Van Cassidy – persona secundaria (Alumno)

## Biografía



- 21 años de edad
- Nacido en Breda, Holanda
- Es estudiante de ingeniería informática en la Hogeschool van Amsterdam
- Es estudiante Erasmus esta año en la escuela de ingeniería informática de la universidad politécnica de Valencia
- Vive en un piso compartido con otros estudiantes Erasmus de Valencia
- Recibe una beca por deportista de elite, lo que le permite pagar sus estudios con facilidad
- Utiliza el transporte público, ya que no dispone de vehículo propio.
- Apasionado de las bicicletas

## Objetivos

- Ser el mejor en el ciclismo y en sus estudios
- Combinar sus aficiones con los estudios para disfrutar más de ambas cosas
- Acabar sus estudios y vivir de ello
- Seguir aprendiendo sobre las nuevas tecnologías

## Tecnología

- Tiene un ordenador portátil potente para poder desarrollar aplicaciones
- Tiene un Smartphone y es imprescindible en su día a día
- Usa multitud de redes sociales y está al día gracias a ellas
- Desarrolla aplicaciones web de forma esporádica fuera de los estudios
- Le encanta probar programas en desarrollo y se apunta a todas las betas que le interesan
- Como buen informático prefiere usar multitud de tutoriales en la web antes que sacar libros de bibliotecas
- Le encanta trabajar con música, tanto que es un joven dj que ha empezado recientemente a producir su propia música

## Salud

- Desarrollar deporte casi a diario le hace tener una salud casi perfecta

### ILUSTRACIÓN 3: PERSONA SECUNDARIA - ALUMNO

## Corina Randazzo – Persona secundaria (Auxiliar)

### Biografía



- 25 años de edad
- Nacida en Argentina pero vive en España desde que tenía 5 años
- Trabaja en la biblioteca de la Universidad Politécnica de Valencia como auxiliar
- Vive en un piso con su hermana en la misma ciudad de Valencia
- Titulada en magisterio con especialidad para trabajar en bibliotecas
- Sus padres viven en Málaga
- Cobra unos 20000 euros anuales
- Le apasiona la lectura

### Objetivos

- Desarrollar su trabajo lo más eficiente posible
- Tener algo de tiempo libre para visitar a sus padres
- Tener vida social fuera de la biblioteca

### Tecnología

- Está acostumbrada a usar ordenadores debido a su trabajo
- Tiene un smartphone y una tablet que usa a diario
- Usa multitud de redes sociales y está al día gracias a ellas

### ILUSTRACIÓN 4 : PERSONA SECUNDARIA - AUXILIAR

Como podemos ver tras analizar 3 personas tipo llegamos a las siguientes conclusiones:

- Los profesores querrán compartir contenido con sus alumnos y en cualquier momento. Puede ser que el profesor esté en su casa preparando la clase del día siguiente y vea un documento interesante de cara a esa cara. Debido a esto sería interesante que hubiera una opción de compartir los documentos mediante varios métodos para que el profesor se lo pudiera mandar a sus alumnos de forma sencilla y directa.
- La mayoría de usuarios usan más de un dispositivo móvil, por lo que consideramos acertada la opción de desarrollar la aplicación orientada a ellos.
- Debido a lo anterior y como los usuarios no tiene mucho tiempo para aprender, la aplicación deberá ser fácil de utilizar.

- Vemos como los estudiantes prefieren hacer consultas en internet para ver que documentos les van a venir mejor de cara a sus exámenes o trabajos. Por esto sería de gran importancia tener un buen sistema de comentarios en los documentos para que así puedan conocer de antemano las experiencias de otros alumnos y ver si realmente es lo que necesitan.
- Además estos alumnos vemos como prefieren buscar tutoriales directamente en la red antes que consultar libros u otro tipo de documentos. Sería deseable que nuestra aplicación además alojara documentos online como presentaciones o vídeos para enseñar a los alumnos de forma más dinámica.
- Dado que los profesores usarán la plataforma para sugerir publicaciones a sus alumnos, puede que llegue el momento de que vayan a sugerir un documento y este no se encuentre en la biblioteca. Sería deseable desarrollar un sistema de peticiones para los profesores. De esta forma podrían solicitar un documento a los administradores.
- Finalmente para que cada uno tuviera una idea de la situación de sus préstamos o reservas, es necesario hacer una sección específica para cada usuario en la que pueda ver el estado de estos.

## 2.4. ANALISIS TECNOLÓGICO

Esta fase del análisis es tan importante como el análisis de requisitos, ya que con una buena elección de la tecnología para desarrollar la aplicación lograremos un gran avance de cara a la implementación y diseño de la misma.

Una vez establecidos los objetivos y las necesidades del producto a desarrollar descritas en el punto anterior vamos a proceder a elegir la tecnología que utilizaremos para desarrollar nuestra herramienta.

El primer paso de todos consiste en elegir el sistema objetivo para nuestra aplicación. Hay multitud de sistemas operativos móviles, entre los que destacan: Android, IOS, Blackberry y Windows Phone. La elección del sistema objetivo es el primer paso importante, por lo que vamos a realizar un análisis de cada uno de ellos.



## 2.4.1 Análisis de sistemas operativos

### 2.4.1.1 Android

El sistema operativo de Google. Su principal característica es su libre distribución y licencia de código abierto. Esto significa que cualquiera puede tener acceso al código del sistema operativo y modificarlo a su gusto para poder distribuirlo luego. Esto también conlleva que haya una amplia comunidad de desarrolladores que pueden dar soporte ya sea en comunidades, foros o portales creados expresamente para ello.

Este sistema ha tenido tanto éxito entre las compañías de teléfonos que brinda al usuario un amplio catálogo de distintos modelos de diferentes compañías. Con esto podemos tener móviles con una gran resolución de pantalla, otros solo para jugar, uno diseñado para hacer fotos de gran calidad o incluso algunos para visitar las redes sociales y hacer un uso mínimo de las demás funcionalidades.

Este amplio abanico nos permite encontrar móviles en el mercado a cualquier precio.

La gran desventaja de este sistema operativo es su mercado de aplicaciones “Play Store”. Este mercado es donde los desarrolladores suben sus aplicaciones y al ser un sistema de código abierto no hay nadie revisando las aplicaciones que se suben, originando así un gran número de malware en el mercado.

### 2.4.1.2 iOS

El sistema operativo que revolucionó la telefonía tal y como la conocíamos hace apenas unos años. Es un sistema exclusivo de la marca Apple y brinda una experiencia única al usuario otorgándole un funcionamiento sencillo, intuitivo y un gran rendimiento.

Tiene un amplio mercado de aplicaciones que además es el mercado que más beneficios reporta anualmente. Las aplicaciones son revisadas una a una antes de ser puestas en el mercado por un equipo de trabajadores de Apple para asegurar una calidad mínima en las aplicaciones.

Uno de los principales inconvenientes de este sistema operativo es la poca variedad de terminales que lo utilizan. Además de que estos terminales tienen un coste muy elevado debido a la exclusividad de la marca. Se puede concluir que ofrecen terminales de máximo rendimiento pero se dejan por cubrir la cuota de calidad baja y media del mercado.



### 2.4.1.3 Windows Phone

Windows comenzó su andadura con los sistemas operativos móviles con Windows Mobile, un sistema operativo para las PDAs. Esto supuso un gran éxito en aquel entonces pero con el paso del tiempo se vio relegado a un segundo plano con la irrupción de iOS y Android en el mercado.

Ahora tras el lanzamiento de Windows 8 que permite usar el mismo sistema operativo tanto en ordenadores como en tablets están intentando ganar terreno a las otras dos compañías. Bien es sabido que Windows nunca ha sido código abierto pero con la nueva actualización a Windows 10 se prevee que esto cambie, ya que se ha anunciado que será gratuita.

Actualmente Windows no tiene gran cuota de mercado ni el mejor rendimiento en los dispositivos móviles pero tras la adquisición de Nokia empezaron a lanzar smartphones bajo esa marca de distintas características ofreciendo una amplia gama de dispositivos. Con el paso del tiempo otras compañías también han optado por usar este sistema operativo en algunos de sus terminales como por ejemplo HTC.

Debido a la baja cuota de mercado también tiene un catálogo de aplicaciones escaso, aunque se encuentra en continuo crecimiento.

Así pues Windows puede dar un gran salto en el mercado si consigue unificar los sistemas operativos de ordenadores, tablets y smartphones. Esto permitiría al usuario saber utilizar un solo sistema operativo para usar tres dispositivos distintos, reduciendo en mucho tiempo la curva de aprendizaje del usuario y haciendo más satisfactoria su experiencia.

### 2.4.1.4 Blackberry

El comienzo de Blackberry estuvo orientado al entorno profesional, ofreciendo herramientas de comunicación precisa y soluciones eficientes para los trabajadores que necesitaban estar conectados todo el día. Con el paso del tiempo se vio relegada a un segundo plano en favor de Apple.

El principal atractivo de estos dispositivos era un teclado físico QWERTY en lugar del típico teclado virtual que suele ocasionar frustración a los usuarios. Esto no fue suficiente para atraer a los clientes y se vieron obligados a reinventarse con el lanzamiento de Blackberry 10.

Este sistema operativo cambiaba por completo la apariencia de las Blackberrys, otorgándoles una interfaz similar a los Android o iOS. Su principal punto a favor consistía en la forma de desarrollar aplicaciones. Permitían el desarrollo de aplicaciones mediante un SDK normal como el resto de sistemas pero además la propia marca fomentó el uso de HTML5 entre los desarrolladores. Esto se explica debido a que



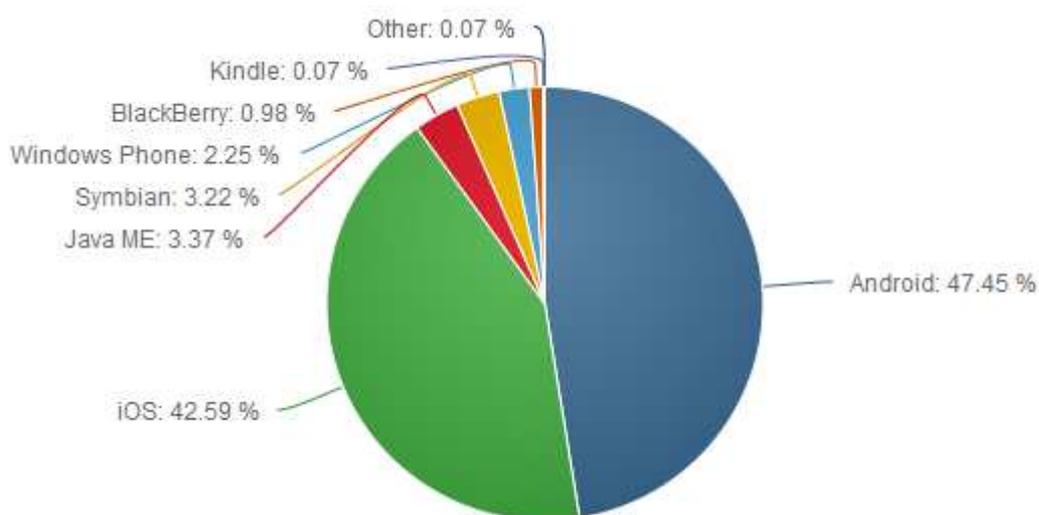
Blackberry está orientado a que se pueda usar el máximo del dispositivo mediante el código HTML5, mientras que los demás sistemas operativos tienen algo limitada esta característica.

Además con el lanzamiento del nuevo sistema operativo se fomentó la creación de aplicaciones entre los desarrolladores con un programa de puntuación para llenar el mercado rápidamente.

A pesar de esta revolución se ha podido ver como no ha tenido mucho éxito entre los usuarios y continúa relegada en el segundo plano.

#### 2.4.1.5 Comparación de SO [4]

Vamos a proceder a una breve comparación respecto a las cuotas de mercado de los sistemas operativos analizados y algunos minoritarios más.



ILUSTRACION 5: SISTEMAS OPERATIVOS PARA DISPOSITIVOS MÓVILES

En la ilustración 2 se puede apreciar como Android supera a iOS por un margen de un 5%. Esta figura representa el total de dispositivos móviles y su respectivo sistema operativo. Esto demuestra un claro avance de Android respecto a su máximo competidor iOS. Android aumenta un 12% respecto al total su cuota de dispositivos, que coincide justo con el porcentaje sobre el total que pierde iOS.

Esto hace pensar que en el futuro Android seguirá aumentando su presencia en el resto de dispositivos móviles y lo confirma como el sistema a batir. Además vemos como Windows Phone y BlackBerry no alcanzan ni un 3% sobre el total, lo que los hace poco interesantes como objetivo de nuestra aplicación.

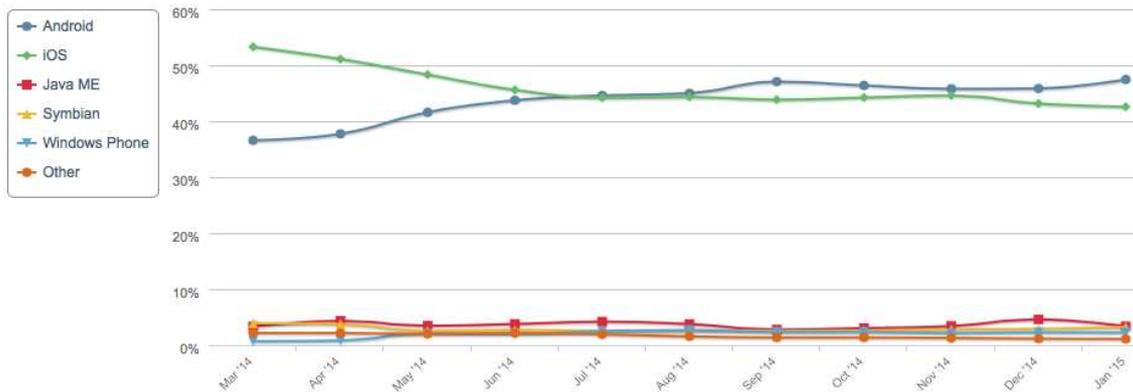


ILUSTRACIÓN 6: TENDENCIA DE LOS SISTEMAS OPERATIVOS DURANTE EL AÑO 2014.

La figura anterior ilustra la evolución de los sistemas operativos durante el año 2014, donde podemos apreciar claramente el gran avance de Android y a caída de iOS respecto a él.

Sin embargo se puede observar como el resto de sistemas minoritarios no suponen amenaza alguna para los otros dos sistemas.

Además de todo esto durante este último año se han producido intentos de desbancar a Google y su Android como mayor sistema operativo como **Tizen, Firefox, Baidu...** Aunque se ha podido comprobar que no han tenido éxito alguno y se han quedado en un mero intento.

Debido a todas estas conclusiones es fácil pensar que el objetivo de nuestra aplicación debe ser Android o iOS, ya que nos aseguraríamos que la mitad de smartphones pudieran acceder a nuestra aplicación. Pero ahora surge un gran debate por cuál de las dos escoger y vamos a intentar resolverlo en el próximo punto.

#### 2.4.2 Elección de Sistema Operativo.

Una vez hecho el análisis anterior vamos a elegir el sistema operativo para el que desarrollaremos nuestra aplicación.

Para comenzar propondremos algunas características que debería tener nuestra aplicación y ver qué sistema se adecua mejor a ella.

- En primer lugar necesitamos que nuestra aplicación llegue al máximo número de usuarios. Como la mayoría de usuarios son estudiantes suponemos que no todos pueden costearse un Smartphone de gama alta así que nuestra aplicación debe estar disponible para las gamas medias y bajas también. De esta forma descartamos por el momento desarrollarla para iOS.



- En segundo lugar nos centramos en el rendimiento del móvil. Como en el punto anterior se ha destacado queremos que se utilice en todas las gamas de potencia por lo que necesitaremos que nuestra aplicación consuma el mínimo de recursos del dispositivo móvil.
- En tercer lugar tenemos que investigar los mercados de aplicaciones para ver cuál es el más interesante para nosotros y debido a la gran cuota de mercado que tienen Android y iOS son los que más nos interesan.
- Como último punto clave necesitaremos una gran comunidad que nos de soporte en caso de que nos surjan dudas y problemas durante la programación de la aplicación nos resulte fácil encontrar una solución. Además debido a la experiencia durante la carrera en desarrollo de aplicaciones Java y aplicaciones web sería interesante que el lenguaje de programación fuera uno de estos para evitar aprender desde cero un nuevo lenguaje.

La solución a todas estas exigencias no se puede encontrar en un único sistema operativo, por lo tanto ahora es cuando debemos centrarnos en la idea que tuvo Blackberry con el lanzamiento de Blackberry 10 y el desarrollo de aplicaciones en HTML5.

Así que la gran decisión no es elegir el sistema objetivo si no el tipo de aplicación que vamos a desarrollar, por lo que ahora vamos a examinar los distintos tipos de aplicaciones que hay y lo que esto conllevaría.

### 2.4.3 Análisis de aplicaciones

#### 2.4.3.1 Aplicaciones nativas

Una aplicación nativa es una aplicación desarrollada específicamente para un sistema operativo.

Cada empresa, ya sea Google, Apple, Windows o BlackBerry, pone a disposición de los desarrolladores un SDK(Software Development Kit). Este SDK incluye un conjunto de herramientas y programas de desarrollo que permiten a los desarrolladores crear aplicaciones para un sistema operativo en este caso. En la mayoría de los casos incluyen herramientas de debugger, códigos de ejemplo e incluso un entorno de programación propio.

En función de esto los 4 sistemas anteriormente estudiados se podrían separar por su SDK y su lenguaje de programación:

- Android: Utiliza el lenguaje Java y sus SDK más importante es Android Studio [5].
- iOS: Utiliza el lenguaje Objective C y su SDK está disponible en Apple Developer [6].
- Windows Phone: Se programa en lenguaje C# y Visual Basic .NET con la ayuda de su SDK [7].
- BlackBerry: Hace uso del lenguaje C++ mediante su SDK [8]. Es la plataforma más evolucionada en este tema ya que actualmente han adaptado su SDK para poder programar también en Java y en ActionScript.

Desarrollar una aplicación nativa es la mejor opción en caso de que se quiera sacar el máximo rendimiento al software y hardware del dispositivo. Todo depende de la experiencia de los desarrolladores pero con un buen código el resultado será el mejor ya que se desarrolla específicamente para un sistema operativo siguiendo las directrices que imponen los desarrolladores del mismo.

Las principales ventajas de una app nativa son:

- Se tiene acceso a todo el hardware del dispositivo como la cámara, el GPS, el acelerómetro y cualquier otro extra que tenga el dispositivo como pueden ser escáner de huellas dactilares, pulsómetro, podómetro....
- Se obtiene completo acceso a las librerías gráficas del SO.
- Se permite el envío de notificaciones “push” [9], que son la principal herramienta de comunicación de las apps.
- Presencia de la aplicación en los stores oficiales.
- Permiten el funcionamiento sin conexión a internet.

Sin embargo no todo son ventajas en este tipo de aplicaciones. Como vimos en la decisión para elegir sistema operativo, este tipo de aplicaciones te limita el usuario objetivo de la aplicación ya que solo se puede acceder a una mitad del mercado si se escoge Android y a la otra mitad si se escoge iOS. Desarrollar dos aplicaciones iguales pero para distintos sistemas supondría un alto coste económico y de tiempo. Además de lo que supondría la curva de aprendizaje de aprender los lenguajes de programación.

La otra gran desventaja de las aplicaciones nativas son las continuas actualizaciones del sistema operativo. Estas actualizaciones obligan al desarrollador a modificar la aplicación para adecuarla a las nuevas funcionalidades.



### 2.4.3.2 Aplicaciones híbridas

Básicamente son aplicaciones que contienen en su interior el navegador web del dispositivo. Para su desarrollo se utilizan frameworks de desarrollo basados en lenguajes de programación web. Entre los frameworks más utilizados destacan los siguientes [10]:

- **PhoneGap:** La aplicación se realiza con HTML5, CSS3 y JavaScript. Son aplicaciones web empaquetadas para ser tratadas como aplicaciones. Básicamente lo que se construye es una web pero orientada a la usabilidad y experiencia del usuario de un móvil. Una vez hecho esto el framework se encarga de crear la aplicación y adaptarla para que sea visible correctamente en el navegador del dispositivo.  
La mayor ventaja es el ahorro de tiempo programando.
- **Titanium Appcelerator:** Este framework hace uso de JavaScript como lenguaje de programación. Además ofrece un entorno de desarrollo propio al desarrollador basado en Eclipse.  
Lo que la diferencia de la anterior es que al compilar traduce a código nativo la aplicación, haciendo que sus aplicaciones den mayor experiencia al usuario y mejor rendimiento.  
De esta forma tendríamos que hacer una versión para un SO, compilar y luego retocar algunas partes del código para recompilar para otro SO.
- **Strawberry:** Este framework divide la programación en dos: la lógica se hace en C++ y el diseño en HTML/CSS. Así se puede separar los roles de maquetador y desarrollador.  
Utiliza la aceleración por hardware OpenGL [11] y tiene un re-renderizado propio, lo que hace que las aplicaciones se vean igual en todas las plataformas y dispositivos.

Vistos los principales frameworks nos podemos hacer una idea de cómo son las aplicaciones híbridas. El grado de integración con el SO dependerá en mayor medida del framework elegido y como de abierto sea el SO. De esta forma se tiene bastante acceso al hardware del teléfono e incluso a veces a las librerías del sistema, aunque no ha conseguido mejorar la respuesta y experiencia de usuario de una aplicación nativa.

Finalmente cabe destacar el progreso de este tipo de aplicaciones que ahora están combinando funcionalidades ejecutadas como web y otras en nativo. De esta forma para las funcionalidades que requieran uso del hardware del teléfono da mayor rendimiento ejecutarlo como nativo.

### 2.4.3.3 Aplicaciones Web

Estas aplicaciones se caracterizan por ser simplemente una página web. Consiste en una web a la que se accede a través de un navegador del dispositivo y esta se adapta al formato de la pantalla para que tenga un aspecto de navegación App. Al tratarse de una web normal y corriente los lenguajes que se utilizan para su programación son HTML5, CSS3 y JavaScript.

También se dispone de frameworks que ayudan el desarrollo de estas webs y proporcionan estilos propios para crear interfaces específicas para dispositivos móviles. Estos frameworks funcionan de tal manera que el desarrollador debe enlazar las librerías JavaScript y las hojas de estilos para luego poder hacer uso de sus funcionalidades.

El framework más conocido es JQuery Mobile[12]:

- Es un framework optimizado para dispositivos táctiles basado en el framework JQuery [13]. Es compatible con la gran variedad de dispositivos móviles (smartphones, tablets...) y además también se puede integrar dentro de los frameworks para aplicaciones híbridas como PhoneGap.
- La estructura básica de un proyecto de JQuery Mobile consiste en unas líneas de código para enlazar las librerías JavaScript de JQuery y JQuery Mobile y las hojas de estilos propias. La propia web oficial del framework pone a disposición del desarrollador un creador de estilos muy sencillo en el que puedes ver como quedarían los elementos de la interfaz al momento.
- Una vez enlazadas las librerías y hojas de estilos se define el cuerpo del documento mediante unos divs con atributos "data-role" y así el propio framework se encarga de aplicar los estilos.
- Una vez el usuario conoce los atributos de los elementos que aporta JQuery Mobile a los mismo de HTML5 le resultará muy sencillo crear distintas páginas ya que todas siguen la misma estructura.

La mayor ventaja de este tipo de aplicaciones es que no ocupan espacio en el dispositivo del cliente ya que están alojadas en un servidor web. Con esto también se evita un uso excesivo de recursos del dispositivo. Lo malo de esto es que la aplicación solo será accesible cuando haya conexión a internet.

La gran desventaja es el limitado acceso al hardware del dispositivo por no decir nulo. Además de que estas apps no se encuentran en las Stores, a diferencia de los otros dos tipos de aplicaciones.

### 2.4.4 Elección de aplicación

Con toda la información recogida en el punto anterior vamos a volver a plantear el debate de qué tipo de aplicación nos conviene más. Por ello vamos a volver a presentar



las características expuestas en el punto de elección de sistema operativo y ver qué tipo de aplicación se adecua más a nuestra aplicación.

- En primer lugar necesitamos que nuestra aplicación llegue al máximo número de usuarios. De esta forma descartamos las aplicaciones nativas ya que no disponemos del tiempo necesario para desarrollar aplicaciones para los cuatro sistemas operativos.
- En segundo lugar nos centramos en el rendimiento del móvil. Nuestra aplicación no necesita gran capacidad de procesamiento ni acceso al hardware del dispositivo por lo que las ventajas que nos ofrecían las aplicaciones nativas no nos importan. Además las web apps al estar instaladas en un servidor externo evitan el ocupar espacio en el dispositivo móvil, por lo que es una gran ventaja.
- En tercer lugar tenemos que decidir si nuestra aplicación debe estar o no en un mercado de aplicaciones. Como es un servicio adicional a la UPV y toda la intranet y servicios suministrados están alojados en servidores web. De esta forma para mantener una estabilidad entre todos los servicios que se proporcionan sería deseable que el acceso a la aplicación fuera mediante un link en la propia intranet.
- Como último punto clave necesitaremos una gran comunidad que nos de soporte en caso de que nos surjan dudas y problemas durante la programación de la aplicación nos resulte fácil encontrar una solución. Como durante la carrera se aprende a programar en Java, HTML, JavaScript, Php.... Lo más factible sería hacer una aplicación nativa de Android o desarrollar aplicaciones híbridas o web apps.

Vistos todos estos puntos nos faltaría analizar un último punto. Nuestra aplicación tiene que hacer consultas a una base de datos externa para poder llevar a cabo la mayoría de sus funcionalidades. Esto conlleva que para que la aplicación funcione correctamente se necesita hacer uso de internet, lo que hace que la ventaja de las aplicaciones híbridas y nativas de poder ser usadas offline no sea un punto a favor de ellas, ya que nuestra aplicación siempre necesitará acceso a internet.

Con todo lo analizado se estima conveniente desarrollar una Web App. Nos basamos en los 4 puntos analizados anteriormente ya que con desarrollar una única aplicación conseguimos toda la cuota de mercado a nivel de sistema operativo. Lo segundo es el tiempo que nos ahorramos programando respecto a las híbridas, ya que las híbridas necesitan retoques para adaptarlas a cada SO. El punto para decidir entre híbridas y web apps es que nuestra aplicación no necesita hacer uso del hardware del teléfono móvil. Debido a esto se prefiere desarrollar una webApp ya que el desarrollo será más sencillo y se eliminan todas estas características extras que no son necesarias.

# 3. Especificación del sistema

La fase de especificación consiste en documentar de forma sencilla y precisa, las necesidades y funciones específicas de un sistema determinado.

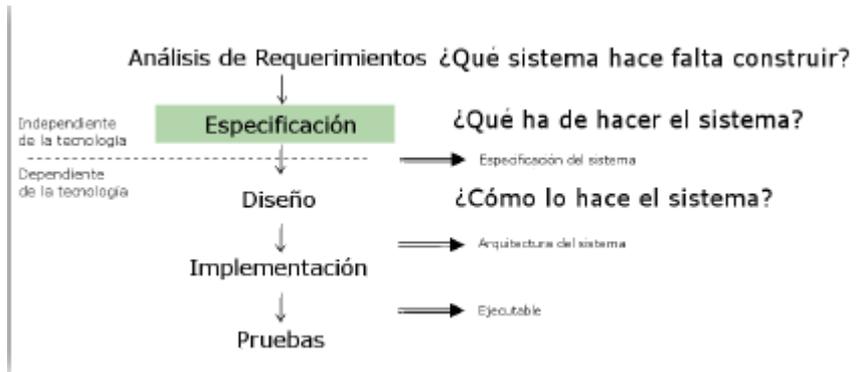


ILUSTRACIÓN 7: FASES DE DESARROLLO SOFTWARE (ESPECIFICACIÓN)

En la fase de especificación vamos a definir los casos de uso. Esta técnica sirve para definir las interacciones entre un sistema y sus actores. De esta forma se detalla la forma en que el sistema reacciona a un evento iniciado por un actor.

Los casos de uso surgen de los requisitos funcionales del sistema y ya hace una pequeña especificación de cómo será el diseño de la aplicación, a nivel de que debe haber en las secciones por las que debe pasar el usuario pero no el cómo se implementan.

Primero se mostrarán los diagramas de casos de uso, que definirán de forma gráfica los casos de uso, y seguidamente se describirá cada uno de ellos de forma detallada.

En el segundo apartado se mostrará el diagrama de clases y el modelo entidad relación de la base de datos.

## 3.1. CASOS DE USO

Una vez visto que son los casos de uso, vamos a proceder a mostrar los actores y el diagrama de casos de uso. Los actores son entidades externas al sistema que guardan relación con este y le demanda funcionalidad.



### 3.1.1 Actores

Dentro de la aplicación diferenciamos dos actores muy parecidos pero con alguna pequeña diferencia, el usuario alumno y el profesor. Un alumno puede acceder a todo el contenido de la aplicación al igual que un profesor. La diferencia está en que el profesor tiene acceso a algunas funciones más para contactar de forma rápida con los administradores.

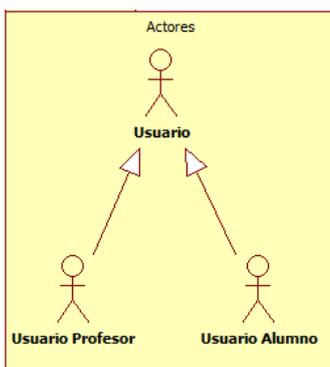


ILUSTRACIÓN 8: JERARQUIZACIÓN DE ACTORES DE LA APLICACIÓN

### 3.1.2 Diagramas de casos de uso

A continuación podemos ver la siguiente figura que ilustra los casos de uso de la aplicación móvil junto con los actores que los realizan.

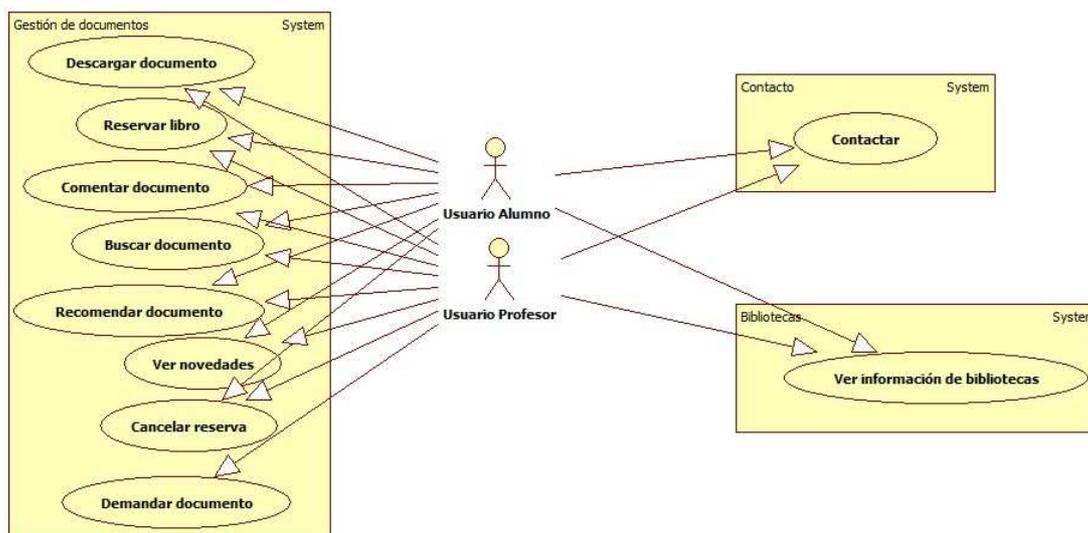


ILUSTRACIÓN 9: DIAGRAMA DE CASOS DE USO DE LA APLICACIÓN

### 3.1.3 Descripción de los casos de uso

A continuación se va a detallar cada uno de los casos de uso de nuestra aplicación.

<b>UC1: RESERVAR LIBRO</b>
<b>Descripción :</b> El usuario puede reservar un libro disponible para luego pasarse por la biblioteca y recogerlo.
<b>Actores:</b> Usuario profesor y usuario alumno.
<b>Flujo principal:</b> <ol style="list-style-type: none"><li>1. El usuario abrirá la aplicación.</li><li>2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.</li><li>3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.</li><li>4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.</li><li>5. El usuario deberá elegir la opción de libros.</li><li>6. El sistema le mostrará una lista con todos los documentos, del tipo libro, disponibles en la biblioteca.</li><li>7. El usuario selecciona un libro.</li><li>8. El sistema muestra una nueva pantalla en la cual se pueden ver los datos referentes al libro, así como las opciones de recomendar, comentar y un selector de fecha.</li><li>9. El sistema mostrará un selector de fecha para seleccionar la fecha en la que quiere el libro y dispondrá de dos semanas desde esa fecha para devolverlo.</li><li>10. El usuario selecciona tal fecha y pulsa en reservar.</li><li>11. El sistema muestra un mensaje de confirmación</li><li>12. Finalmente el usuario debe ir a la biblioteca y coger el libro.</li></ol>
<b>Primer flujo alternativo – Reservar cualquier otro documento físico (Paso 4 flujo básico)</b> <ol style="list-style-type: none"><li>1. El usuario puede seleccionar novedades, destacados, audio y revistas para poder continuar con el flujo básico y reservar un documento de ese tipo de la misma forma que el anterior.</li></ol>
<b>Segundo flujo alternativo – Libro no disponible actualmente (Paso 9 flujo básico)</b> <ol style="list-style-type: none"><li>1. En caso de que el libro no esté disponible en este momento al usuario se le indicará mediante un mensaje diciendo que no está disponible y que seleccione fecha para anticipar reserva.</li><li>2. El sistema mostrará ese mensaje y un selector de fecha.</li><li>3. El usuario seleccionará la fecha en la que deseará recoger el libro de entre las disponibles una vez el documento sea devuelto. Cuando la seleccione pulsará en reservar.</li><li>4. El sistema muestra un mensaje de confirmación</li><li>5. Finalmente el usuario debe ir a la biblioteca y coger el libro.</li></ol>



<b>UC2: BUSCAR DOCUMENTO</b>
<p><b>Descripción</b> : El usuario puede buscar un documento introduciendo cualquier carácter. Estos caracteres se cotejarán con los resultados de compararlos con la base de datos y los caracteres introducidos. Se comparará por título, autor, editorial y lengua.</p>
<p><b>Actores:</b> Usuario profesor y usuario alumno.</p>
<p><b>Flujo principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario abrirá la aplicación.</li> <li>2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.</li> <li>3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.</li> <li>4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.</li> <li>5. El usuario introducirá el texto que quiera buscar y luego pulsa sobre buscar.</li> <li>6. El sistema le mostrará una lista con todas las coincidencias de comparar el texto introducido con la base de datos</li> <li>7. El usuario ahora podrá seleccionar cualquier documento y continuar con otros casos de uso.</li> </ol>
<p><b>Primer flujo alternativo – No hay coincidencias (Paso 5 flujo básico)</b></p> <ol style="list-style-type: none"> <li>1. El sistema no encuentra coincidencias en la búsqueda y muestra la lista de resultados vacía.</li> </ol>

<b>UC3: VER NOVEDADES</b>
<p><b>Descripción</b> : El usuario puede ver una lista de los últimos documentos añadidos a la biblioteca.</p>
<p><b>Actores:</b> Usuario profesor y usuario alumno.</p>
<p><b>Flujo principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario abrirá la aplicación.</li> <li>2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.</li> <li>3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.</li> <li>4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.</li> <li>5. El usuario seleccionará la opción de novedades.</li> <li>6. El sistema le mostrará una lista con los últimos documentos que se han añadido a la biblioteca.</li> <li>7. El usuario ahora podrá seleccionar cualquier documento y continuar con otros casos de uso.</li> </ol>

#### UC4: DEMANDAR DOCUMENTO

**Descripción :** El usuario (con privilegios de profesor) puede solicitar a los administradores de la biblioteca la inserción de un nuevo documento a ella misma.

**Actores:** Usuario profesor.

**Precondición:** El usuario debe tener permisos especiales para solicitar un documento. Solo los profesores o auxiliares pueden usar esta funcionalidad, así que el usuario deberá poseer uno de estos dos roles.

**Flujo principal:**

1. El usuario abrirá la aplicación.
2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.
3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.
4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.
5. El usuario seleccionará la opción de me sección.
6. El sistema le mostrará tres secciones: mis libros, mis reservas, mis peticiones.
7. El usuario pulsará sobre el botón de solicitar documento, que se encuentra en la sección de mis peticiones.
8. El sistema le muestra un formulario para introducir el título, el autor, la editorial y el año de publicación del documento.
9. El usuario rellena los datos y pulsa en el botón de solicitar.
10. El sistema le muestra un mensaje de confirmación.

#### UC5: CANCELAR RESERVA

**Descripción :** El usuario dispondrá de la opción de cancelar una reserva anticipada que el usuario haya realizado.

**Actores:** Usuario profesor y usuario alumno.

**Precondición:** El usuario debe haber realizado el flujo alternativo 2 del caso de uso 1 (Anticipar reserva).

**Flujo principal:**

1. El usuario abrirá la aplicación.
2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.
3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.
4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.
5. El usuario seleccionará la opción de me sección.
6. El sistema le mostrará tres secciones: mis libros, mis reservas, mis peticiones.
7. El usuario pulsará sobre el botón de cancelar, que se encuentra en la sección de mis reservas, correspondiente a libro que anticipó su reserva y ahora desea cancelarlo.
8. El sistema le mostrará un mensaje confirmando su cancelación.



<b>UC6: RECOMENDAR DOCUMENTO</b>
<p><b>Descripción</b> : El usuario dispondrá de la opción de compartir un documento a través de diferentes mecanismos en caso de que le resulte necesario.</p>
<p><b>Actores:</b> Usuario profesor y usuario alumno.</p>
<p><b>Flujo principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario abrirá la aplicación.</li> <li>2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.</li> <li>3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.</li> <li>4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.</li> <li>5. El usuario deberá elegir la opción de libros.</li> <li>6. El sistema le mostrará una lista con todos los documentos, del tipo libro, disponibles en la biblioteca.</li> <li>7. El usuario selecciona un libro.</li> <li>8. El sistema muestra una nueva pantalla en la cual se pueden ver los datos referentes al libro, así como las opciones de recomendar, puntuar, comentar y reservar.</li> <li>9. El usuario seleccionará la opción recomendar.</li> <li>10. El sistema le muestra una lista al usuario con las posibles opciones de recomendación del libro: email, twitter, facebook, whatsapp.</li> <li>11. El usuario seleccionará la opción que más desee y ya seguirá un flujo distinto dependiendo de la opción que elija. Este flujo es independiente a la aplicación “Casiteka” y depende de la API de cada opción disponible.</li> </ol>
<p><b>Primer flujo alternativo – Recomendar cualquier otro documento (Paso 4 flujo básico)</b></p> <ol style="list-style-type: none"> <li>1. El usuario puede seleccionar novedades, destacados, audio, revistas, docs online o resultados de búsqueda para poder continuar con el flujo básico y reservar un documento de ese tipo de la misma forma que el anterior.</li> </ol>

### UC7: OPINAR SOBRE UN DOCUMENTO

**Descripción :** El usuario dispondrá de la opción de dar su opinión sobre un documento.

**Actores:** Usuario profesor y usuario alumno.

**Flujo principal:**

1. El usuario abrirá la aplicación.
2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.
3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.
4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.
5. El usuario deberá elegir la opción de libros.
6. El sistema le mostrará una lista con todos los documentos, del tipo libro, disponibles en la biblioteca.
7. El usuario selecciona un libro.
8. El sistema muestra una nueva pantalla en la cual se pueden ver los datos referentes al libro, así como las opciones de recomendar, puntuar, comentar y reservar.
9. El usuario se desplazará hasta la sección de comentarios y pulsará sobre comentar.
10. El sistema le mostrará una ventana con los campos a rellenar de nombre, email y comentario.
11. El usuario los rellenará y pulsará sobre comentar. El botón comentar estará inutilizable hasta que se rellenen los tres campos.
12. El sistema procesa el comentario y muestra la confirmación al usuario.

### UC8: VER INFORMACIÓN DE BIBLIOTECA

**Descripción :** El usuario podrá consultar información sobre cualquier biblioteca de la UPV. De esta forma podrá conocer con exactitud su ubicación, horarios, teléfonos de contacto...

**Actores:** Usuario profesor y usuario alumno.

**Flujo principal:**

1. El usuario abrirá la aplicación.
2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.
3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.
4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.
5. El usuario deberá elegir la opción de bibliotecas y horarios.
6. El sistema le mostrará una lista con todas las bibliotecas que pertenecen a la UPV.
7. El usuario seleccionará una biblioteca de la lista.
8. El sistema muestra una nueva pantalla en la cual se pueden ver los datos referentes a la biblioteca.
9. El usuario ahora puede ver toda la información referente a la biblioteca seleccionada, como los horarios, la ubicación, teléfono de contacto....



<b>UC9: CONTACTAR</b>
<p><b>Descripción</b> : El usuario podrá contactar con administración en caso de que tenga dudas referentes a cualquier tema de la aplicación.</p>
<p><b>Actores:</b> Usuario profesor y usuario alumno.</p>
<p><b>Flujo principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario abrirá la aplicación.</li> <li>2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.</li> <li>3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.</li> <li>4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.</li> <li>5. El usuario deberá elegir la opción de contacto.</li> <li>6. El sistema mostrará una ventana en la que habrá distintos campos a rellenar por el usuario: nombre, email, asunto y mensaje.</li> <li>7. El usuario rellenará todos los campos y en caso de no rellenar todos, el botón de enviar permanecerá bloqueado. Una vez rellenados pulsará en enviar.</li> <li>8. El sistema mostrará un mensaje de confirmación</li> <li>9. Posteriormente los administradores responderán al usuario mediante email.</li> </ol>

<b>UC10: DESCARGAR DOCUMENTO</b>
<p><b>Descripción</b> : En la biblioteca hay también documentos digitales, los cuales podrán ser descargados por los usuarios.</p>
<p><b>Actores:</b> Usuario profesor y usuario alumno.</p>
<p><b>Flujo principal:</b></p> <ol style="list-style-type: none"> <li>1. El usuario abrirá la aplicación.</li> <li>2. Al abrir la aplicación el sistema muestra un menú para introducir los credenciales de usuario y permitir así el login de los usuarios.</li> <li>3. El usuario debe introducir su usuario y contraseña. Una vez hecho esto procede a loguearse.</li> <li>4. El sistema muestra menú con distintas opciones: mi sección, novedades, destacados, libros, audio, revistas, docs online, bibliotecas y contacto. Además de todo esto se incluye un campo para realizar búsquedas.</li> <li>5. El usuario deberá elegir la opción de docs online,</li> <li>6. El sistema mostrará una lista con todos los documentos del tipo online. Estos documentos están alojados en el servidor y están disponibles para descargar. Pueden ser del tipo pdf, word, mp3, avi....</li> <li>7. El usuario seleccionará uno de la lista.</li> <li>8. El sistema muestra una nueva pantalla en la cual se pueden ver los datos referentes al libro, así como las opciones de recomendar, puntuar, comentar y descargar.</li> <li>9. El usuario pulsará sobre descargar y automáticamente se iniciará la descarga del documento.</li> <li>10. Una vez descargado el usuario puede utilizar el documento en su dispositivo en cualquier momento.</li> </ol>

## 3.2. MODELO CONCEPTUAL

Una vez determinados los casos de uso, se debe determinar un modelo de datos para el sistema. Un modelo de datos está compuesto por clases, atributos, métodos y relaciones que representarán los objetos del sistema.

Este modelo conceptual se define a través de un diagrama de clases en el que cada clase contiene unos atributos y unos métodos. Una vez tenemos definidas todas las clases se procede a establecer la relación que hay entre cada una de ellas y de esta forma obtenemos el diagrama de clases.

En función de este diagrama de clases podemos obtener también el modelo entidad relación que servirá para definir la estructura de nuestra base de datos y las relaciones entre las tablas de la misma.

### 3.2.1 Descripción del modelo

A continuación se muestra el diagrama de clases de la aplicación. En él se pueden apreciar el conjunto de clases con relaciones bastante simples. Hay que tener como objetivo principal la simplicidad del sistema, de cara a que sea más fácil el aprendizaje del usuario.

Este modelo de datos representa un sistema en el que todos los usuarios pueden interactuar con el sistema en función de su rol.

De esta forma vemos como el núcleo principal de la aplicación se basa en la interacción entre usuarios y documentos. Un usuario puede interactuar con los documentos de distintas formas.

La primera de ellas consiste en descargar un documento cualquiera, donde vemos que un documento puede ser descargado por cualquier usuario y un usuario puede descargar tantos documentos como quiera.

La segunda forma es mediante la creación de préstamos. Un usuario puede solicitar un documento de forma física y esto queda registrado en el sistema como un préstamo. Un usuario puede realizar cuantos préstamos quiera pero un préstamo solo puede pertenecer a un usuario. Además un préstamo depende siempre de un único documento. En caso de que un documento desaparezca y tenga un préstamo asociado, este préstamo se cancelará.

Lo mismo pasa con los comentarios. Un usuario puede comentar tantos documentos como quiera las veces que quiera pero un comentario solo puede estar hecho por un usuario. Un documento tiene comentarios ilimitados mientras que un comentario es específico para un documento.

A parte de la gran relación entre usuarios y documentos existen otras dos relaciones más en nuestro modelo.



La relación entre usuarios y bibliotecas que consiste simplemente en que los usuarios pueden consultar la información de cualquier biblioteca y las bibliotecas pueden ser consultadas por cuantos usuarios quieran.

La última relación es la de usuarios con peticiones. Esta relación especial, ya que solo los usuarios que tenga el atributo “rol” igual a profesor podrán tener esta relación. Un usuario profesor puede crear cuantas peticiones quiera, mientras que una petición solo puede ser creada por un usuario.

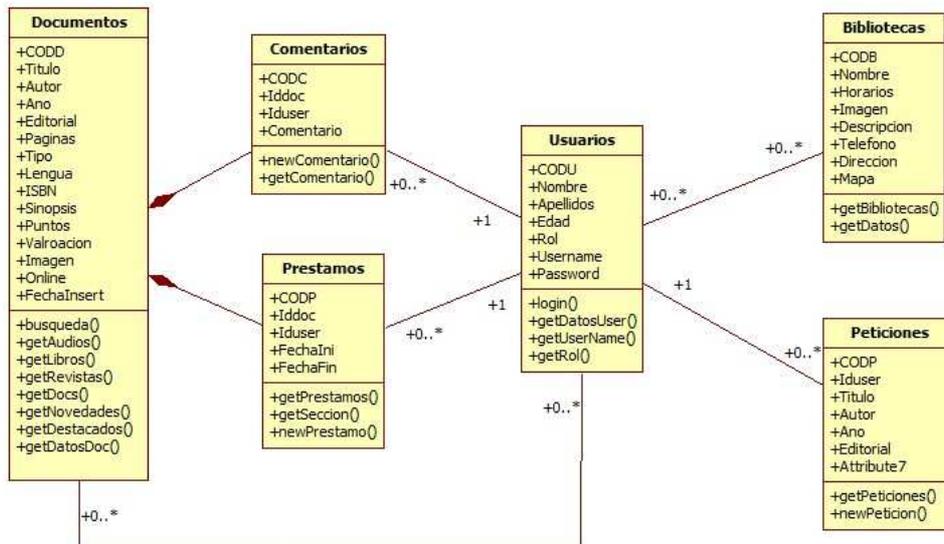


ILUSTRACIÓN 10: DIAGRAMA DE CLASES

Una vez explicada la estructura de nuestra aplicación podemos ver en el siguiente modelo entidad relación la estructura que tendrá nuestra base de datos.

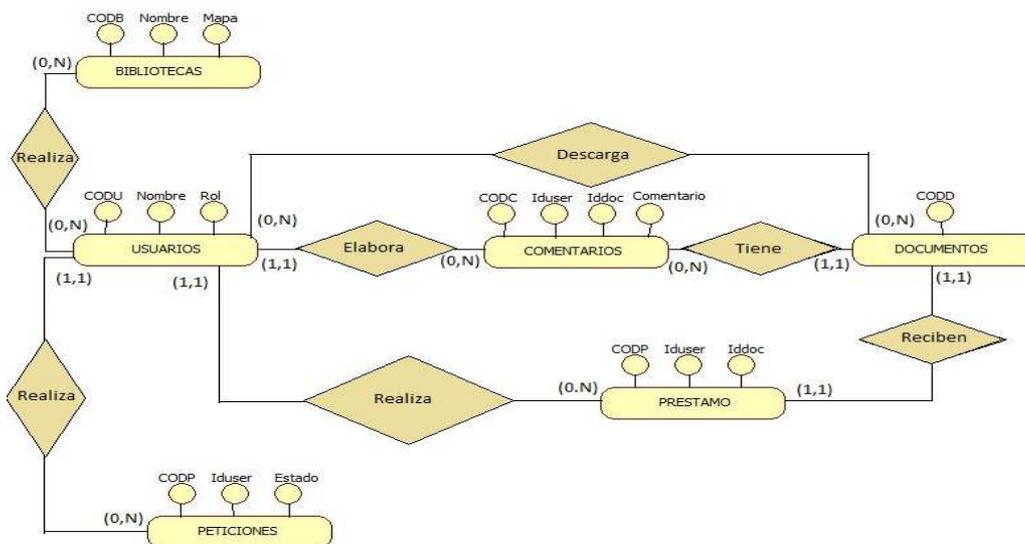


ILUSTRACIÓN 11: MODELO ENTIDAD RELACIÓN

## 4. Diseño

La fase siguiente a la especificación es la de diseño. En esta fase se va a estudiar y decidir cómo va a funcionar el sistema, basándonos en su arquitectura y teniendo en cuenta todos los requisitos no funcionales. El objetivo final es definir un sistema con el suficiente detalle que permita su implementación.

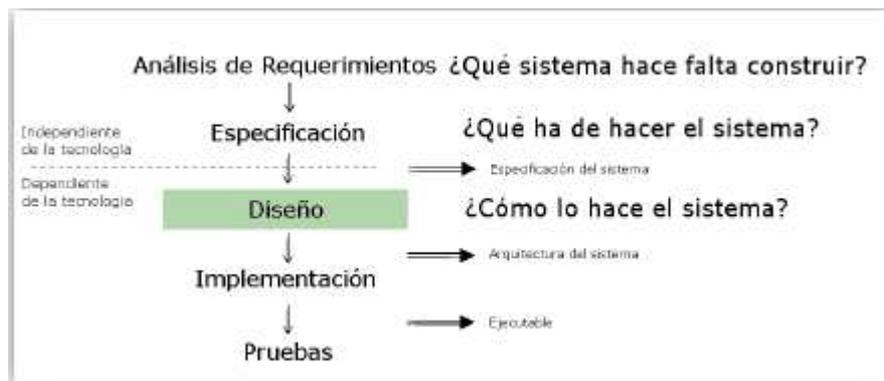


ILUSTRACIÓN 12: ETAPAS DE DESARROLLO SOFTWARE (DISEÑO)

El principal objetivo de la fase de diseño consiste en fomentar la calidad del proyecto, ya que se debe materializar con precisión los requisitos del cliente. Además de esto otros objetivos del diseño son:

- Constituir el sistema en base al lenguaje de programación utilizado.
- Conocer y analizar en profundidad los requisitos no funcionales del sistema y obtener una arquitectura de calidad acorde a los requisitos establecidos.
- Crear una abstracción del desarrollo de todos los componentes del sistema. Esto permitirá en un futuro la posibilidad de rediseñar el proyecto y su implementación de forma más sencilla.

Para comenzar con la sección de diseño se describirá la arquitectura física del sistema para que estos se adecuen a las funcionalidades descritas anteriormente de la mejor forma posible. Posteriormente se describirán las tecnologías elegidas para el desarrollo de la aplicación en el análisis tecnológico. Finalmente se mostrará la descripción de las interfaces junto con un diagrama de transición de estados para entender mejor que hace cada pantalla.

#### 4.1. ARQUITECTURA FÍSICA

Debido al análisis tecnológico llegamos a la conclusión de que en función de nuestros requisitos y necesidades necesitábamos desarrollar una web app. Como ya se ha explicado una web app es una web alojada en un servidor externo al sistema del cliente. Una vez aclarado esto podemos ver como la arquitectura básica del sistema seguirá el patrón Cliente-Servidor que se justifica por las siguientes razones:

- La centralización de datos hará que no sea necesaria ninguna replicación de datos en cada uno de los clientes. Al igual que permitirá mantener dos entornos de desarrollo y aplicar los cambios al de producción de forma transparente al usuario.
- Escalabilidad: En caso de que sea necesario se puede mejorar la estructura del sistema. Si en algún momento se incrementa el tráfico de usuarios en un gran número podremos añadir nuevos servidores en los que replicar la información y montar un servicio QoS mediante colas y prioridades. De esta forma evitaríamos cuellos de botella sobre un servidor y aumento del tiempo de respuesta de la aplicación. Sería tan simple como repartir el número de clientes entre el número de servidores y así reducir la carga sobre cada servidor.
- Al estar alojado en un servidor podemos añadir tantas medidas de seguridad como queramos, ya sean firewalls, host bastion, más entornos de preproducción... Además la posibilidad de hacer un backup de toda la aplicación y los datos es mucho más sencilla ya que no tenemos que acceder a la información del sistema cliente.

Para resumir un poco la estructura del proyecto sería la siguiente. Un servidor que contenga toda la aplicación y los datos de ella en una base de datos al que acceden los clientes. Los clientes acceden a la aplicación a través del navegador de su dispositivo móvil. También pueden acceder a través de un ordenador pero la aplicación está optimizada para el uso en dispositivos móviles y no se garantiza el correcto funcionamiento en ordenadores.

Para desarrollar todo esto se hace uso de las siguientes tecnologías:

- La aplicación se desarrollará con JQuery Mobile, ya que como vimos en el análisis tecnológico es el framework más conocido y del que más documentación se dispone.
- Una base de datos desarrollada en MySQL alojada en el mismo servidor que la aplicación. Sería un módulo totalmente independiente de la aplicación al que se accedería mediante consultas PHP.
- La comunicación entre la base de datos y la aplicación se lleva a cabo mediante métodos PHP que encapsulan la información en formato JSON y se parsea en la aplicación. Ya sea para enviar información a la base de datos o para recibirla.

- Además se dispone de un servicio de gestión de correo en el mismo servidor para poder atender todas las consultas que nos lleguen al buzón mediante la sección de “contacto”.

El servidor donde se alojará todo será por el desarrollador, pero su ubicación no estará relacionada con el desarrollador. Concretamente el servidor es suministrado por la empresa one.com [14], que tiene alojados sus servidores en Dinamarca como se puede ver en algunas páginas que muestran donde están físicamente alojados los dominios web [15].

## 4.2. TECNOLOGÍAS UTILIZADAS

A continuación se van a definir todas las tecnologías escogidas anteriormente. Estas tecnologías son las que se utilizan tanto para implementar la aplicación, como el servidor e incluso las comunicaciones.

### 4.2.1 One.com

Esta es la empresa que nos suministra el servidor físico donde se aloja toda la aplicación. Nos hemos decantado por ella solamente por el aspecto económico ya que durante el primer año ofrecen un dominio gratuito con una capacidad máxima de 15 GB.

En nuestro caso la aplicación excede ese tamaño y tampoco los datos almacenados en la base de datos. Si en un futuro se llegara a ocupar todo ese espacio, la empresa ofrece la posibilidad de ampliación de tamaño de servidor. De esta forma para el comienzo tenemos un espacio reducido aunque suficiente pero tenemos la opción de ampliar este espacio si el sistema lo necesita.

Las principales ventajas que nos ofrece este servicio de alojamiento son las siguientes:

- Editor web: Un editor WYSIWYG (What You See Is What You Get) permite realizar páginas web mucho más fácil. Un editor web no es necesariamente un paquete de software que se instala en el ordenador personal tras haber sido comprado.

En este caso se proporciona al usuario un editor web en línea y se pueden utilizar desde cualquier ordenador con acceso a internet. Con este editor se proporcionan multitud de plantillas para utilizar como base para un nuevo sitio web.

Como nuestra aplicación no es una aplicación web normal, ya que usa el framework JQuery Mobile no hemos hecho uso de este editor pero si hemos usado el editor de texto normal que ofrecen en su File Manager, en lugar de utilizar otro como SublimeText o Dreamweaver. Esta decisión se ha tomado



para poder programar la aplicación desde cualquier ordenador en función de nuestras necesidades.

- **Asistencia 24/7:** Se ofrece una asistencia las 24 horas del día durante los 7 días de la semana mediante un chat. De esta forma tenemos apoyo técnico respecto a cualquier duda que nos surja durante el uso de los servicios de one.com en cualquier momento.
- **SSL:** One.com utiliza SNI (Server Name Indicator) que hace posible tener varios certificados SSL en el mismo número IP. Además te aseguran un certificado SSL a coste 0. Ofrecen ejecución y mantenimiento de este certificado y garantizan que la seguridad es de primera clase.
- **Estadísticas:** Ofrecen una herramienta avanzada y fácil de usar para analizar y optimizar los sitios web de manera a ofrecer el mejor servicio y altas cifras de visitantes. La interfaz gráfica muestra multitud de datos como el número de visitantes, de dónde vienen y cuánto tiempo pasan en el sitio web.



ILUSTRACIÓN 14: ESTADÍSTICAS DE PÁGINAS VISITADAS DURANTE UN MES

- **Blogs:** Ofrecen la creación de uno o más blogs en un dominio. Es fácil de utilizar, ya que con unos pocos clics permite al usuario crear un blog único basado en una selección de plantillas. También cabe la posibilidad de instalar el sistema Wordpress.
- **Wordpress:** Wordpress es tan sencillo de utilizar que el mayor inconveniente de utilizarlo es la instalación y selección de todos los módulos necesarios para nuestro blog. Debido a esto One.com ofrece un instalador específico de WordPress muy sencillo de utilizar.
- **SSH y FTP:** Gracias a FTP se pueden transferir archivos directamente a un espacio web. Se puede utilizar cualquier cliente FTP para subir archivos al servidor, pero One.com ofrece su propio File Manager. Este permite transferir

los documentos de forma rápida y sencilla, además de la posibilidad de editarlos gracias al editor de texto que incorpora. Este editor ha sido el utilizado para programar la aplicación.

- **PHP y MySQL:** También ofrecen acceso a PHP 5 y a una base de datos MySQL, que es la que hemos utilizado para guardar toda la información de la aplicación referente a usuarios, bibliotecas, documentos....
- **Google Adwords:** Como One.com es socio de Google, estos ofrecen a sus clientes un cupón de anuncios AdWords. De esta forma nuestro sitio web se ofrecería dentro del sistema de marketing dirigido creado por Google. Este sistema consiste en mostrar publicidad a los visitantes que realmente están buscando lo que el sitio web anunciante puede ofrecerles.

De esta forma nuestro sitio web saldrá como sugerencia en el motor de búsqueda de Google y también anunciado en los sitios web de los clientes de google que estén relacionados con la aplicación anunciante.

- **Servicio de Email:** One.com proporciona la creación de un número ilimitado de direcciones de correo electrónico bajo el dominio solicitado. Además ofrece WebMail, para poder consultar el correo y otras características como antivirus o filtros de spam. Haremos uso de este servicio para gestionar todos los emails recibidos mediante la aplicación.
- Además de todos estos servicios ofrecen otros como: Administrador DNS, Anycast DNS, BIX (sincronización de dispositivos con el servidor, como Dropbox o Google Drive).

#### 4.2.2 JQuery Mobile [16]

JQuery Mobile es un framework desarrollado en javascript perteneciente a jQuery Foundation, el cual nos proporciona una interfaz compatible y usable para desarrollar aplicaciones móviles para teléfonos y tablets, y que cuenta con un sistema de transición entre paginas mediante tecnología AJAX. Además es compatible con la mayoría de las plataformas móviles y navegadores para dispositivos móviles (iOS, Android, Blackberry, WebOS, Symbian, Windows Phone, etc.).

JQuery Mobile lleva el famoso dicho “Write less, do more” (escribe menos, haz más) a un nuevo nivel: Este diseña webs de forma automática con una estética fácil de usar y que además funcionará en la mayoría de dispositivos móviles.

En lugar de escribir una aplicación para cada dispositivo móvil o sistema operativo jQuery Mobile resuelve este problema mediante el uso de HTML5, CSS3 y JavaScript para desarrollar una aplicación web estándar que será visible a través de cualquier navegador móvil.



#### 4.2.2.1 Introducción al diseño en jQuery Mobile

Como bien se ha explicado a lo largo de esta memoria vamos a utilizar los lenguajes de programación HTML5, CSS3 y JavaScript para desarrollar la interfaz y lógica interna de la aplicación.

Para empezar vamos a describir brevemente en qué consisten estos tres lenguajes:

- **HTML:** En realidad no es un lenguaje de programación, sino un lenguaje de marcado de hipertexto que se utiliza para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código para la definición de contenido web.

Se basa en la referencia en lugar de la inserción de elementos. De forma más sencilla significa que en lugar de añadir una imagen al archivo se crea un link a la ruta donde este archivo está alojado y luego el navegador se encarga de interpretar esta línea y busca el archivo en la ruta indicada para mostrarlo en la web.

- **CSS:** También conocido como hojas de estilo es un lenguaje utilizado para definir y crear la presentación de un documento estructurado escrito en HTML o XML. El objetivo principal del CSS es separar la estructura de un documento de su presentación.

La información de estilos se puede definir en un documento separado o en el mismo documento HTML.

La sintaxis CSS es muy sencilla y la estructura de una hoja de estilos está compuesta de una lista de reglas. Una regla consiste en uno o más selectores y un bloque de declaración. Un bloque de declaración está compuesto por una o varias duplas “propiedad:valor” que luego serán aplicadas al selector indicado en la misma regla.

- **JavaScript:** Es un lenguaje de programación interpretado. Se define como un lenguaje orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza para realizar operaciones y por norma en el marco de la aplicación cliente sin acceso a funciones del servidor. JavaScript interpreta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Como podemos ver con estos tres lenguajes podemos hacer todo lo que queremos. Pero si nos fijamos en la descripción de JavaScript, dice que no tiene acceso a funciones del servidor. Ahí es donde entra en juego el papel de PHP que se encargará de acceder a esas funciones y JavaScript se limitará a interpretar los resultados que devuelven los métodos PHP para mostrarlos en la aplicación. Luego se entrará más en detalle sobre PHP y la comunicación entre la base de datos y la aplicación.

#### 4.2.2.2 Añadiendo jQuery Mobile a una página web

Una vez explicados por encima los distintos lenguajes que se usarán vamos a explicar cómo comenzar a desarrollar una página web basada en jQuery Mobile.

El primer paso consiste en añadir jQuery Mobile al sitio web. Para hacer esto hay dos métodos distintos:

- Añadir un link a la librería desde el CDN: El CDN (Content Delivery Network) se usa para distribuir archivos muy utilizados a través de la red. Esa opción hace que la velocidad de descarga sea mucho mayor para el usuario que respecto a una librería alojada en un servidor propio. A continuación se muestra el código necesario para incluir las librerías y hojas de estilo en la página HTML.

```
<head>
<!--Metadatos para autoescalar la pantalla y fijar el zoom
predeterminado -->
<meta name="viewport" content="width=device-width, initial-
scale=1">
<!--Incluye la hoja de estilos de jQuery Mobile -->
<link rel="stylesheet"
href="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.css">
<!--Incluye la librería de Jquery -->
<script src="http://code.jquery.com/jquery-1.11.3.min.js">
</script>
<!-- Incluye la librería de Jquery Mobile -->
<script src="http://code.jquery.com/mobile/1.4.5/jquery.mobile-
1.4.5.min.js"></script>
</head>
```

Estas son las líneas imprescindibles para comenzar un proyecto. Además de estas se pueden insertar tantas importaciones de librerías como de hojas de estilos se estimen necesarias. En caso de usar muchas se deberá tener cuidado de que no se apliquen dos o más estilos a un mismo elemento ya que habría conflicto entre ellos y obtendríamos una página distinta de la esperada.

- Añadir un link a la librería alojada en el servidor: Como ya se ha comentado en el punto anterior esta opción es menos aconsejable ya que la descarga de librerías es más lentas. Pero en caso de tener hojas de estilo o librerías propias es necesario usar este método.

Simplemente tenemos que cambiar la ruta “src” en caso de los javascript y “href” en caso de las hojas de estilo por la ruta en la que está alojado el archivo dentro de nuestro proyecto. Con poner la ruta desde el directorio raíz donde se encuentra el archivo index.html es suficiente.



### 4.2.2.3 Creando la primera página

Ahora que ya tenemos claro cómo debemos comenzar cualquier proyecto jQuery Mobile vamos a mostrar cómo es la estructura básica de una página.

```
<body>
<div data-role="page">
  <div data-role="header">
    <h1>Welcome To My Homepage</h1>
  </div>
  <div data-role="main" class="ui-content">
    <p>I Am Now A Mobile Developer!!</p>
  </div>
  <div data-role="footer">
    <h1>Footer Text</h1>
  </div>
</div>
</body>
```

- La etiqueta data-role="page" define que esta página será la mostrada por el navegador. De esta forma podemos definir más de una página en un mismo archivo html y dotar la aplicación de navegación mediante el uso de las etiquetas <a href> convencionales.
- La etiqueta data-role="header" crea una barra de herramientas en la parte superior de la página. Esta sección se suele utilizar para añadir funcionalidades de navegación o el propio título de la aplicación.
- La etiqueta data-role="main" define el contenido de toda la página. Este contenido puede ser desde textos, imágenes, botones, listas y demás formas.
- La etiqueta class="ui-content" añade estilo al div. De esta forma definimos que la página es de clases ui-content y el navegador busca el estilo referente a esta clase en la hoja de estilos para aplicárselo.
- La etiqueta data-role="footer" crea una barra de herramientas en la parte inferior de la página. Esta sección se suele utilizar para dar información relativa al desarrollador de la página o información de poco interés al usuario sobre la empresa.
- Dentro de estos contenedores se puede añadir código HTML para crear cualquier elemento como si fuera una página web normal. Lo único que hay que tener en cuenta es las etiquetas de cada forma para luego llamarlas correctamente.

### 4.2.2.3 Efectos visuales

JQuery Mobile incluye efectos CSS3 que permiten al usuario multitud de opciones. Entre ellos destacan:

- **Transiciones:** el CSS permite al desarrollador elegir la forma en la que una página se debe abrir. Ofrece una gran variedad de transiciones entre una página y otra pero estas solo se pueden ejecutar en un navegador que soporte transformaciones 3D.

De esta forma nos quedamos con la siguiente tabla de compatibilidad con los navegadores más importantes junto con el número de versión mínimo para ejecutar las transiciones.

				
12.0	10.0	16.0	4.0	15.0

ILUSTRACIÓN 15: NAVEGADORES COMPATIBLES CON TRANSICIONES

Para añadir una transición tan sólo debemos añadir la etiqueta *data-transition="tipoTransicion"* a la etiqueta *<a href>* que queramos dotar de transición. El valor de "tipoTransicion" tomará un valor en función del tipo de transición que se quiera poner (flip, flow, pop, slide, sideup...)

- **Iconos:** Un inconveniente principal de utilizar una web app es que esta tiene una limitación muy grande a la hora de hacer uso del hardware del dispositivo móvil. Esto hace que no sea posible acceder a las librerías de iconos.

Aquí es donde jQuery provee al usuario un sistema de iconos propio y de fácil acceso para el desarrollador. Simplemente debemos añadir las etiquetas *ui-icon-X*, *ui-btn-icon-Y* a la etiqueta *class* del botón al que queramos añadir un icono.

La primera añade el botón de tipo X (arrow, info, delete, search, back....) y la segunda etiqueta indica la posición del botón donde queremos que lo muestre (top, right, left, bottom).

Se ofrecen multitud de opciones de personalización como eliminar el texto en el botón y mostrar solo el icono, quitar el círculo que sale alrededor del icono o cambiar el color del icono entre blanco y negro.

- **Popups:** Un popup es muy parecido a los diálogos y estos pueden ser utilizados para mostrar un pequeño texto, fotos, mapas u otros contenidos.

Para crear un popup debemos tener dos elementos: un *<a>* y un *<div>*. La funcionalidad consiste en que el elemento *<a>* llama al elemento *<div>* que



tiene la característica *data-role="popup"* y de esta forma se muestra como un popup en lugar de como una página nueva.

#### 4.2.2.3 Elementos principales

Para desarrollar una interfaz compatible con las pantallas de dispositivos móviles, jQuery Mobile dispone de multitud de elementos diseñados específicamente para uso en pantallas táctiles y de tamaño reducido.

- Botones: Para crear los botones podemos hacerlo de tres formas distintas:
  - `<input type="button" value="Button">`: Utilizado para validar formularios.
  - `<button class="ui-btn">Button</button>`: Su uso es similar al del anterior.
  - `<a href="#anylink" class="ui-btn">Button</a>`: Aconsejado para enlazar con otras páginas.

Del mismo modo que en cualquier página normal HTML existen otros tipos de botones pero modificados para los dispositivos móviles. Estos botones pueden ser: botón de atrás, botones alineados, botones agrupados de selección única o checkbox entre otros.

- Toolbars: Estas barras de herramientas se suelen colocar tanto los headers como en los footers, previamente explicados. De esta forma y mediante los atributos de posición de botones también explicados podemos colocar varios botones en estas secciones de forma sencilla e intuitiva. Los botones se van colocando alineados de forma horizontal, por lo que no hace falta añadir ningún atributo, tan solo tendríamos que poner si queremos que este a la derecha o la izquierda de la sección.
- Paneles: Consisten en secciones que aparecen de un lado u otro de la pantalla con información adicional. Para crear un panel tan solo tenemos que añadir la ya conocida etiqueta *data-role="panel"* a un elemento `<div>` y posteriormente llamar a esta sección mediante un elemento `<a>`.

Además de estos tipos de elementos existen muchos otros. Más adelante describiremos en más detalle los elementos que más hemos utilizado en la aplicación junto con algunos ejemplos de su uso.

#### 4.2.3 Comunicaciones

La conexión de datos se realiza directamente entre la base de datos y la aplicación. Toda la lógica está desarrollada en la parte de la aplicación que se encarga de establecer la conexión con la base de datos, hacer la petición de los mismos y una vez recibidos parsea los resultados para mostrarlos en la web.

Todas las conexiones a la base de datos se realizan mediante métodos PHP.

PHP [17] es un lenguaje de código abierto adecuado para el desarrollo web y puede ser incrustado en HTML. Mediante el uso de las etiquetas `<?php` y `</?php>` podemos entrar y salir respectivamente del modo php dentro de un archivo HTML.

Lo mejor de utilizar PHP es su extrema simplicidad para el principiante, pero a su vez ofrece muchas características avanzadas para los programadores avanzados. Este factor es de gran importancia para nuestra aplicación, ya que nunca antes hemos programado en PHP.

Principalmente se usa PHP en tres ámbitos distintos:

- **Scripts del lado del servidor:** Es el objetivo principal de PHP y el ámbito más tradicional. Para que esto funcione son necesarias tres cosas: analizador de PHP, servidor web y navegador web. Es necesario ejecutar el servidor con una instalación de PHP conectada. Se puede acceder al resultado del código mediante un navegador.
- **Scripts desde la línea de comandos:** Se pueden crear scripts y ejecutarlos sin necesidad de servidor o navegador. Solamente es necesario el analizador de PHP para utilizarlo. Esto es ideal para scripts que se ejecuten con regularidad empleando el planificador de tareas.
- **Implementar aplicaciones de escritorio:** Aunque PHP no sea el lenguaje más adecuado para crear aplicaciones con interfaz gráfica si se puede utilizar para emplear las características avanzadas de PHP en aplicaciones del lado del cliente.

Como hemos visto nuestro uso sería el primero, ya que tenemos el analizador de PHP instalado en el servidor web y un navegador desde el que poder acceder al servidor y ver el resultado de nuestras consultas.

Una vez descrito brevemente en que consiste PHP podemos explicar de forma más técnica como se han desarrollado las comunicaciones.

Todos los métodos PHP tienen la misma estructura. Primero realizamos la conexión a la base de datos junto con los credenciales de usuario:

```
$dbhost = "casiteka.com.mysql"; // pondremos nuestro host
$dbuser = "casiteka_com"; // nuestro usuario...
$dbpass = "6iZCRkxM"; // y nuestro pass
$db = "casiteka_com"; // el nombre de la db
$con = new mysqli($dbhost,$dbuser,$dbpass);
$con->set_charset("utf8");
$con->select_db($db);
```



A continuación se envía la petición a la base de datos. Esta petición contiene la sentencia SQL que deberá ejecutar la base de datos una vez reciba esta sentencia.

```
$res = $con->query("SELECT * FROM documentos where codd = '$codd'");
```

Una vez se hace la consulta, se recibe toda la información que cumple la condición de la misma. Ahora tan sólo hay que crear un array con formato JSON y enviarlo de vuelta a la aplicación.

```
$principio="[";
$container="";

while($row = $res->fetch_row())
{
    $codd = $row[0];
    $titulo = $row[1];
    $autor = $row[2];
    $ano = $row[3];
    $editorial = $row[4];
    $paginas = $row[5];
    $tipo = $row[6];
    $lengua = $row[7];
    $isbn = $row[8];
    $sinopsis = $row[9];
    $puntos = $row[10];
    $voto = $row[11];
    $valoracion = $row[12];
    $imagen = $row[13];
    $online = $row[14];
    $iduser = $row[15];
    $fechainsert = $row[16];
    $container.="{"codd\":\"".$codd
    ."\",\"titulo\":\"".$titulo."\", \"autor\":\"".$autor."\", \"ano\":\"".$ano."\", \"editorial\":
    \"".$editorial."\", \"paginas\":\"".$paginas."\", \"tipo\":\"".$tipo."\", \"lengua\":
    \"".$lengua."\", \"isbn\":\"".$isbn."\", \"sinopsis\":\"".$sinopsis."\", \"puntos\":
    \"".$puntos."\", \"voto\":\"".$voto."\", \"valoracion\":\"".$valoracion."\", \"imagen\":
    \"".$imagen."\", \"online\":\"".$online."\", \"iduser\":\"".$iduser."\", \"fechainsert\":
    \"".$fechainsert."
    \"}";
}
$fin="]";
$json=$principio.$container.$fin;
$json = str_replace(",",""],$json);

echo $json;
```

Como podemos ver los métodos PHP son muy sencillos. Solamente tenemos que establecer conexión con la base de datos y enviar los datos devuelta en formato JSON a la aplicación. Posteriormente la aplicación deberá tratar estos datos haciendo uso de la función de getJson que tiene implementada la librería de jquery. Esta función se

encarga de llamar al método PHP anteriormente explicado y de recibir los datos. Luego tan sólo se debe recorrer el array que se recibe para coger los valores y mostrarlos en nuestra aplicación.

Como se puede ver la comunicación es muy simple ya que siempre sigue el mismo patrón y todas las comunicaciones se basan en esto.

Como un pequeño apéndice vamos a explicar en qué consisten los archivos JSON [18]:

- JSON es un formato ligero para intercambio de datos. Es un subconjunto de la notación literal de objetos en JavaScript.
- Debido a su simplicidad ha empezado a sustituir al modelo XML, que también sirve para intercambiar datos. La gran ventaja de JSON respecto a XML es que es mucho más sencillo desarrollar un parser de JSON que de XML. Este ha sido el punto decisivo para elegir JSON en lugar de XML como formato para el intercambio de datos.
- Otra gran diferencia es que JSON es mucho más ligero que XML y por lo tanto permite la transferencia de más datos en menos tamaño.

```
{ "menu": {  
  "id": "file",  
  "value": "File",  
  "popup": {  
    "menuitem": [  
      { "value": "New", "onclick": "CreateNewDoc()" },  
      { "value": "Open", "onclick": "OpenDoc()" },  
      { "value": "Close", "onclick": "CloseDoc()" }  
    ]  
  }  
}
```

ILUSTRACIÓN 16: EJEMPLO JSON

### 4.3. INTERFAZ GRÁFICA

La interfaz gráfica, que es la parte que el sistema presenta al usuario, le comunica información y captura información del usuario. La interfaz es muy conveniente que tenga características amigables para el usuario.

A continuación se van a presentar una serie de diagramas de navegación genéricos por usuario y las debidas capturas de pantallas que representen el diseño gráfico realizado.

Un mapa de navegación permiten saber de forma rápida cual será la forma de navegar a través del software, o dicho de otra forma, pasar de unas pantallas de la aplicación a otras.

Para comenzar vamos a ir presentando una a una todas las interfaces desarrolladas y explicando su funcionamiento para luego relacionarlas todas en el mapa de navegación.



ILUSTRACIÓN 17: VENTANA DE INICIO

En esta imagen vemos la ventana principal de la Casiteka. Vemos como su objetivo no es más que permitir a los usuarios loguearse en el sistema. De esta forma los usuarios introducirán sus credenciales de la UPV y podrán acceder a la aplicación.



ILUSTRACIÓN 18: MENÚ PRINCIPAL

Aquí se muestra el menu principal con las diferentes funcionalidades que la aplicación presta al usuario. Esta es el eje principal de toda la aplicación y desde ella se puede acceder a todo el contenido que ofrece la aplicación.



ILUSTRACIÓN 19: MI SECCIÓN

En esta ventana el usuario podrá consultar todos los libros que tiene reservados actualmente y en la segunda sección los libros que tiene reservados para el futuro. El usuario de tipo alumno tendrá estas dos secciones, mientras que el profesor además dispondrá de la opción de realizar peticiones y ver su estado.



ILUSTRACIÓN 20: MI SECCIÓN2

Esta es la sección que solo podrán ver los usuarios del tipo profesor, en ella se pueden ver una lista de las peticiones realizadas y un pequeño formulario para realizar más.



ILUSTRACIÓN 21: LISTADO

Esta es la ventana en la que se mostrarán todos los listados de documentos que cumplan los requisitos de la búsqueda. se podrá acceder a ella pulsando los botones del menú: libros, audio, revistas, docs online, novedades, destacados y mediante la búsqueda de palabras.

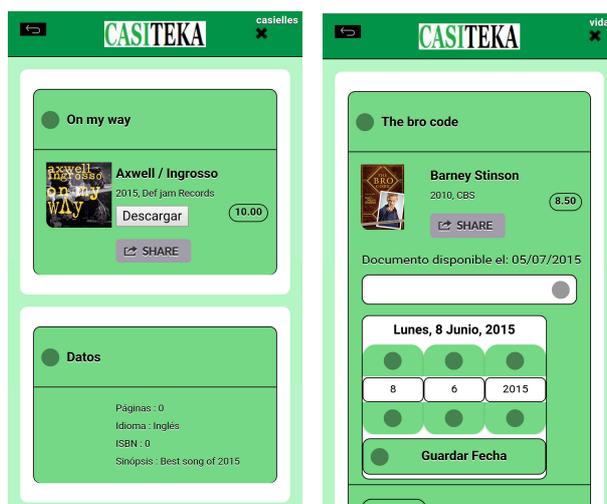


ILUSTRACIÓN 22 Y 23: DOCUMENTO

Tras pulsar sobre un elemento de la lista en la venta de listado, accederemos a todos los datos de un documento. En este caso se muestra un documento online y un libro. para el online vemos como se muestra el botón de descargar y para el libro físico se muestra un aviso de cuando está disponible y un selector de fecha para seleccionar cuando queremos el libro.



ILUSTRACIÓN 24: DOCUMENTO2

Si bajamos más en la ventana de documento llegaremos a la sección donde se muestran los comentarios y un formulario para introducir más.



ILUSTRACIÓN 25: DOCUMENTO3

Otra funcionalidad deseable era la de compartir un documento. En esta ilustración vemos como al pulsar el botón “SHARE” de un documento se muestra un menú para compartir el documento mediante: Pinterest, Twitter, Facebook, Google + y por correo.



ILUSTRACIÓN 26: BIBLIOTECAS

Otra función es la de consultar información sobre las bibliotecas de la universidad. Para ello tenemos que pulsar sobre el botón de “BIBLIOTECAS” en el menú principal y se nos mostrará esta lista. Si pulsamos en cualquiera accederemos a una ventana con su información.



ILUSTRACIÓN 27: BIBLIOTECA

Esta es la ventana que muestra la información de la biblioteca. Vemos como muestra su horario, teléfono, dirección, descripción y un mapa de google maps, con el que el usuario podrá interactuar libremente.



ILUSTRACIÓN 28: CONTACTO

Desde el menú también podemos pulsar en “CONTACTO” y accederemos a esta ventana. En ella se puede apreciar un formulario en el que el usuario podrá escribir su mensaje a los administradores para luego recibir una respuesta a sus sugerencias o dudas.

#### 4.3.1 Mapa de navegación

En el mapa de navegación presentado a continuación, se contempla las diferentes pantallas a las que puede acceder un usuario. Se ha creado el menú de navegación basado en un diagrama de transición de estados. De esta forma cada ventana se consideraría un estado al que se llega mediante una acción descrita en el propio mapa.

Anteriormente ya se ha descrito como se puede acceder a cada una de las funcionalidades, por lo que ahora solo se mostrará de forma esquemática lo explicado anteriormente.

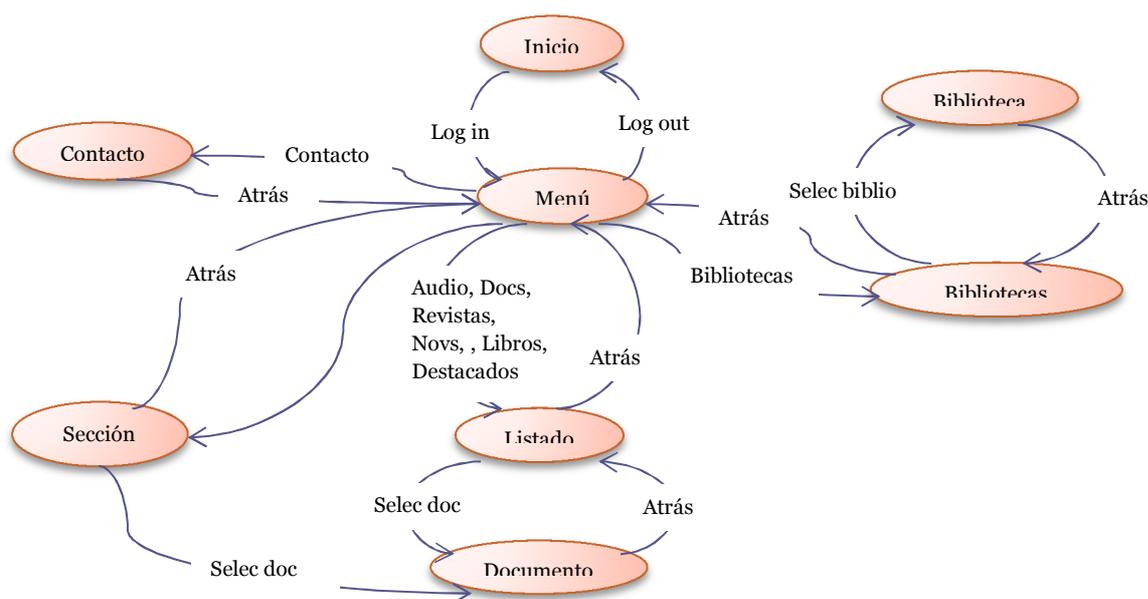


ILUSTRACIÓN 29: MAPA NAVEGACIÓN

#### 4.4. GESTIÓN DE DATOS

La gestión de datos se encarga de la persistencia de datos, haciendo referencia al almacenamiento y obtención de datos usando un sistema de almacenamiento permanente.

Como One.com nos ofrece un sistema de gestión de bases de datos relacional (PhpMyAdmin) vamos a utilizarlo. Al venir ya pre configurado nos evitamos la instalación, configuración y posibles problemas que pudiera darnos instalar otro gestor. Al ser relacional no nos permite almacenar objetos.

En la **Casiteka**, los datos más importantes con los que vamos a trabajar son los diferentes tipos de contenido necesarios para cada funcionalidad. Como ya se vió anteriormente en el modelo entidad relación la estructura que tendrá nuestra base de datos vamos a pasar a explicar cada una de las tablas y sus atributos más importantes. De esta forma se proporcionará una idea general de cómo se almacenan los datos que se muestran en la aplicación en nuestra base de datos.

##### 4.4.1 Documentos

Los documentos son el tipo de contenido más importante de la **Casiteka**. Un documento es cualquier archivo físico o digital que puede ser consultado o reservado por los usuarios. Como son el eje principal de la aplicación están relacionados directamente con casi todas las demás tablas.

Los campos más importantes de un documento son:

- CODD: Código identificativo de cada documento.
- Título: Nombre que define al documento tanto en las listas como en la sección individual de un documento.
- Valoración: Valoración que la **Casiteka** suministra al documento para luego ordenar los documentos por “destacados”.
- Online: Url del archivo en caso de que sea descargable.
- FechaInsert: Nos permite ordenar los documentos de los más recientes a más antiguos en la biblioteca para mostrarlos como “novedades”.

#### 4.4.2 Usuarios

Los usuarios contienen toda la información de cada usuario. Gracias a este tipo de contenido podemos gestionar los usuarios y sus contraseñas, junto con el tipo de usuario que son para darles más o menos privilegios en la aplicación.

Los campos más importantes de un usuario son:

- CODU: Código identificativo de un usuario.
- Rol: Código que nos permite diferenciar los usuarios alumnos de profesores.
- Username: Nombre que el usuario utiliza para hacer el login.
- Password: Contraseña que el usuario utilizara para hacer el login.

#### 4.4.3 Préstamos

Los préstamos contienen la información referente a los documentos reservados por los usuarios. De esta forma podemos tratarlos como una entidad propia y dotarla de atributos para así añadir caducidad a los préstamos y poder gestionarlos en una línea de espacio.

Los campos más importantes de un préstamo son:

- CODZ: Código identificativo para cada préstamo.
- IDdoc: Código identificativo del documento asociado al préstamo.
- IDuser: Código identificativo del usuario que ha realizado el préstamo.
- fechaFin: Fecha en la que el documento debe ser devuelto a la biblioteca.

#### 4.4.4 Comentarios

Los comentarios permiten a los usuarios dar su opinión sobre un documento. Este contenido nos permite dotar de información suplementaria a los documentos y de forma transparente a los desarrolladores, ya que serán los usuarios quienes añadan esta información en función de su experiencia con el documento.

Los campos más importantes de un comentario son:

- CODC: Código identificativo de cada comentario.
- IDdoc: Código identificativo del documento asociado al comentario.



- IDuser: Código identificativo del usuario que ha realizado el comentario.
- Comentario: Opinión del usuario sobre el documento.

#### 4.4.5 Peticiones

Las peticiones sirven para gestionar los documentos demandados por los profesores. Así se permite una fácil gestión de las peticiones y la posibilidad de consultar su estado al creador.

Los campos más importantes de una petición son:

- CODN: Código identificativo de una petición.
- IDuser: Código identificativo del usuario que ha realizado al petición.
- Título: Título del documento que ha solicitado el usuario.
- Estado: Estado de la petición que se mostrará al usuario para que conozca como va su petición.

#### 4.4.6 Bibliotecas

Este contenido almacena toda la información referente a las bibliotecas. Con esto podemos suministrar al usuario desde los horarios hasta un mapa con la ubicación concreta de la biblioteca.

Los campos más importantes de una biblioteca son:

- CODB: Código identificativo de cada biblioteca.
- Dirección: Dirección de la biblioteca para que el usuario sepa dónde ir a recoger un documento reservado.
- Mapa: Url de un mapa de Google para mostrar al usuario donde se sitúa la biblioteca.

# 5. Implementación

Una vez visto el diseño del producto, se va a proceder a describir de forma detallada las tareas realizadas durante el desarrollo de la aplicación.

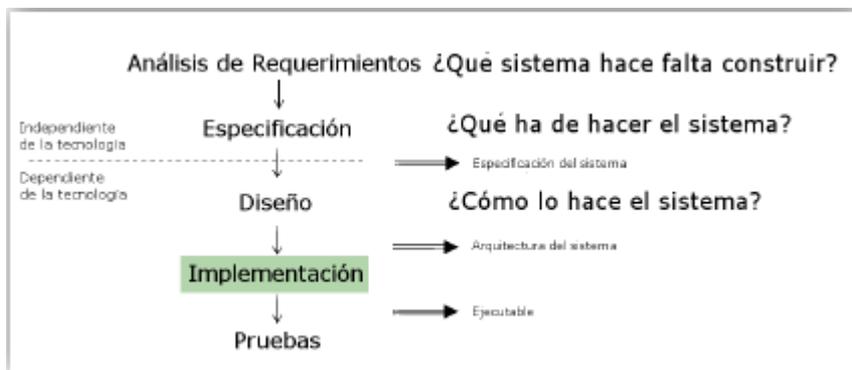


ILUSTRACIÓN 30: ETAPAS DE DESARROLLO SOFTWARE (IMPLEMENTACIÓN)

Este capítulo va a estar dividido en tres grandes partes. Para comenzar vamos a describir el trabajo previo para poner el servidor a punto para nuestra aplicación seguido de una explicación de cómo se comunica una página con otra. Para finalizar, se explicará el desarrollo de las funcionalidades del sistema.

## 5.1. CONFIGURACIÓN INICIAL

Una vez comprado el dominio del servidor y con esto conseguido el acceso a la plataforma de One.com, el primer paso era como alojar la aplicación en nuestro servidor. Como se ha explicado antes One.com ofrece multitud de servicios, entre ellos un File Manager en el que se pueden alojar ficheros.



ILUSTRACIÓN 31: FILE MANAGER

A través de esta herramienta se subirán todos nuestros ficheros html, js, php o css, además de todas las imágenes y demás documentos necesarios para la aplicación. De esta forma se creará un directorio raíz, el cuál contendrá los ficheros “.html” y “.php” más las carpetas “CSS”, “JavaScripts” e “Imágenes”. Cada carpeta contendrá lo que su propio nombre indica, archivos css, JavaScript y las imágenes que se utilicen en la aplicación.

Como punto extra cabe añadir que se usará Dropbox para alojar los archivos online. Si bien es cierto que podríamos alojarlos en nuestro propio servidor, hemos optado por esta opción ya que la velocidad de descarga a través de Dropbox es mayor que si nuestro servidor ofreciera la descarga. Además este cambio de alojamiento no supone ningún problema para la implementación, ya que solo hace falta una url que asociar a cada documento y Dropbox ofrece url públicas a documentos alojados en la carpeta Public [19].

### 5.1.1 Creación de contenido

Una vez decidido cómo vamos a estructurar nuestra aplicación en el servidor, el próximo paso es crear el contenido del que esta se va a nutrir. Para esto haremos uso de la herramienta de “Php & MySQL” que nos ofrece One.com



ILUSTRACIÓN 32: PHP & MYSQL

Esta herramienta permite crear una base de datos perfectamente integrable con la Casiteka. La herramienta es “PhpMyAdmin”, que es la misma que ofrece XAMPP. En la siguiente ilustración se puede apreciar la interfaz básica en la que a la izquierda se muestran todas las tablas de la base de datos y en la sección central se puede ir cambiando la vista para ver el contenido de una tabla, la estructura, ejecutar sentencias SQL, opciones....

Finalmente la descripción de todo el proceso de creación de contenido está explicada técnicamente en el apéndice 1, que se encuentra al final de esta memoria.

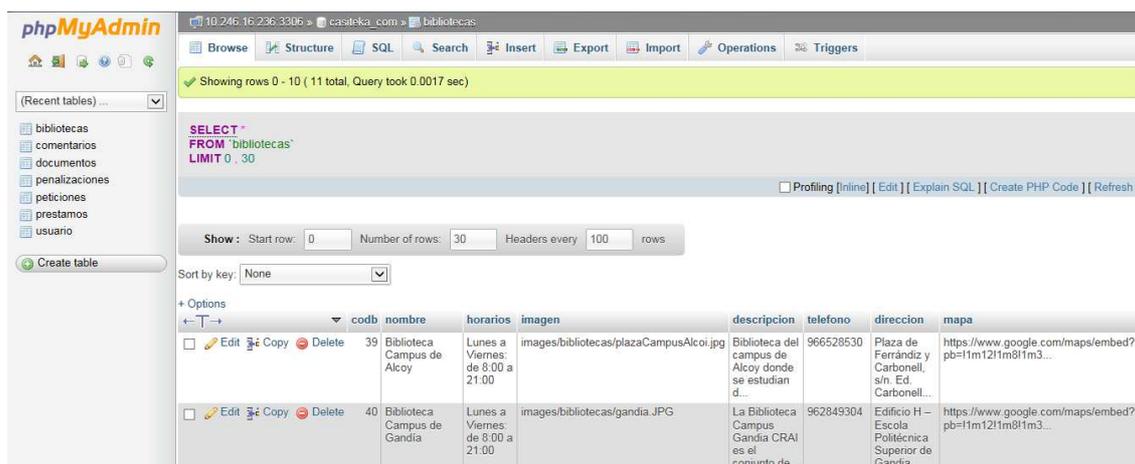


ILUSTRACIÓN 33: PHPMYADMIN

## 5.2. COMUNICACIÓN ENTRE PÁGINAS

Una vez creado el contenido y con esto también los usuarios, el mayor problema que se nos plantea es la gestión de usuarios. Dado que no usamos ningún gestor de contenidos que nos ofrezca una gestión de usuarios ni ningún tipo de framework que nos facilite este cometido debemos implementar una solución mediante código.

Tras investigar un poco las posibles opciones se decide que la más acertada para nuestro cometido es la comunicación mediante el paso de parámetros por la url. Con esto conseguimos una comunicación directa y sencilla de implementar. Para implementarla básicamente tenemos que seguir dos pasos muy sencillos. Primero tenemos que enviar la url con los parámetros previamente leídos de la página y después parsear los parámetros de la url para obtenerlos en la nueva página.

A continuación se muestra un ejemplo del código para enviar unos parámetros y como posteriormente se leen en la siguiente página.

```

window.location.replace("biblioteca.html?usuario="+usuario+"&codu="+codu+
"&codb="+idbib);

```

Con esta línea simplemente se llama a la siguiente página correspondiente del mapa de navegación junto con unos parámetros. Se aprecia como para añadir parámetros se debe añadir una interrogación seguida de una dupla *nombre=valor*. En el caso de querer pasar más de un parámetro se añadirán más duplas separadas por el signo &.

```
function gup( name ){
    var regexS = "[\\?&]" + name + "=( [^&#]* )";
    var regex = new RegExp ( regexS );
    var tmpURL = window.location.href;
    var results = regex.exec( tmpURL );
    if(results == null){
        return "";
    }
    else{
        return results[1];
    }
}
var codu = gup( 'codu' );
var usuario = gup( 'usuario' );
var codb = gup( 'codb' );
```

En el código superior vemos un script que se encarga de acceder a la url de la página actual y parsea el texto de la misma. De esta forma tenemos que llamar a esta función manualmente dentro de nuestra página indicándole el nombre del parámetro que queremos obtener. El método simplemente busca el nombre en la url y devuelve su valor, el cuál asignaremos a una variable global de nuestra página para poder utilizarlo.

En este ejemplo se obtienen de la url los parámetros “codu”, “usuario” y “codb” correspondientes al código de usuario, al username y al código de la biblioteca de la que se va a mostrar la información.

### 5.3. FUNCIONALIDADES PRINCIPALES

Una vez con la estructura de contenidos preparada, junto con los usuarios y además elegido un método de comunicación entre páginas ya se puede empezar a desarrollar todas las funcionalidades que tendrá el sistema.

A modo de destacar las tareas que han requerido mayor esfuerzo, se va a proceder a describir el trabajo realizado para implementar las principales funcionalidades de la aplicación.

Como bien se sabe para usar jQuery Mobile un fichero html siempre tiene que llevar los links a las librerías y hojas de estilos. Además debe tener una estructura fija mínima basada en los tags de *header*, *main* y *footer* dentro de un *div page* como ya se ha explicado en la sección de diseño.

### 5.3.1 Inicio de sesión

El inicio de sesión nos permite entrar a la aplicación. Mediante el inicio de sesión se consiguen los datos del usuario para luego ir navegando por toda la aplicación e ir comunicándolos entre páginas para evitar un login cada vez que se abre una página. De esta forma conseguimos además diferenciar a los alumnos de los profesores y en función de esto habilitar la funcionalidad de las peticiones.

Además de todo esto, el usuario es necesario para cargar la sección de cada uno y mostrar toda su actividad en la **Casiteka**.

Para su desarrollo lo único que hizo falta fue implementar un método en php que permitiera la consulta de usuario y contraseña en nuestra base de datos. Este método comprueba que el usuario indicado tiene asignada la contraseña indicada en la tabla usuarios.

Además se necesita crear un formulario en la página de inicio en el que el usuario introduzca su nombre de usuario y contraseña y que llame al método php. Si son incorrectos el sistema está programado para avisar al usuario de que los datos introducidos son incorrectos.

### 5.3.2 Listar documentos

La **Casiteka** dispone de diferentes tipos de documentos, por esto surge la necesidad de mostrarlos todos o algunos según su tipo en una lista. De esta manera se desarrolla una única página que varía su contenido de forma dinámica en función de la opción que elija el usuario.

Para comenzar el usuario tiene que elegir en el menú principal que tipo de listado quiere ver o introducir una palabra clave en el buscador. A continuación se llama a un método php que realiza un tipo de búsqueda u otro en la tabla *documentos* de la base de datos.

Este método nos devuelve campos como el título, el autor, la imagen o la valoración del documento. Con esto conseguimos los datos necesarios para luego cargarlos en una lista en la que el usuario pueda ver la información más relevante de cada documento.

La estructura del método php es muy parecida al de usuarios, además de que ya ha sido descrita en la sección de diseño y lo único que nos quedaría por implementar es la manera de mostrar los datos en la web.

Para esto hemos creado un elemento de lista con thumb. El thumb nos permite mostrar una imagen en el margen izquierdo del elemento y en el resto se muestra la demás información. A continuación se muestra un ejemplo de elemento.





ILUSTRACIÓN 34: LISTITEM

Para crear cada elemento se ha utilizado el siguiente código:

```
$.getJSON(tipo2, function(json, textStatus) {
    //optional stuff to do after success
    for (var i = 0; i < json.length; i++) {
        var object = json[i];
        j=0;
        for(property in object){
            result[j] = object[property];
            j++;
        }
        codd = result[0];
        titulo = result[1];
        autor = result[2];
        imagen = result[3];
        valoracion = result[4];
        $("#doc").append("<li class='ui-li-has-thumb ui-first-child ui-corner-all ui-last-child'><a href='#' class='ui-btn ui-icon-carat-r'> <img src=' + imagen + "
/><span class='ui-li-count ui-btn-up-c ui-btn-corner-all'>" + valoracion
+ "</span><h3 id='iddoc' class='ui-hidden-accessible'>" + codd + " </h3> <h3>" + titulo
+ "</h3> <p>" + autor + "</p></a></li> <p></p>");
    }
});
```

Con este método llamamos al método php “tipo2” y parseamos el array obtenido de nuestra consulta. Cada objeto del array es un documento que contiene 5 propiedades. Esas propiedades se corresponden con los datos del documento que se van a mostrar y por lo tanto el código coge estos atributos y los añade al listitem mediante la función de jQuery `$.append()`.

Con este código conseguimos crear la lista de todos los documentos que cumplen con los criterios establecidos por el usuario.

### 5.3.3 Reservar documento

Una vez el usuario selecciona un documento el sistema vuelve a hacer una petición php a la base de datos para recuperar todos los datos de dicho documento. Posteriormente estos datos son tratados por la aplicación de la misma forma que en la sección 5.3.2 y son mostrados en tres secciones: Información principal del libro, información adicional y documentos.

Esta sección se centra en la parte de información principal. En esta sección se muestra un selector de fecha o un botón de descargar en función del tipo de documento que estamos viendo. Esto se ha manejado de forma que si el documento tiene el atributo online vacío se muestra el selector de fecha y en caso contrario se muestra el botón de descargar.

Ahora nos vamos a centrar en el selector de fecha. Si el documento se puede reservar de forma física adicionalmente se muestra un mensaje diciendo cuando este documento estará disponible. Entonces el usuario debe seleccionar una fecha en la que desea recoger el documento en la biblioteca. En caso de que la fecha no esté disponible o sea incorrecta la reserva no se tramitará y se volverá a mostrar al usuario a partir de qué fecha está disponible el documento.

Para implementar el selector de fechas se ha tenido que hacer uso de un plugin externo [20], ya que jQuery no proporciona ningún elemento tipo calendario ni parecidos.

Una vez el usuario selecciona una fecha se vuelve a llamar a otra función php muy parecida a la descrita en la sección anterior:

```
$.getJSON( 'newPrestamo.php', {
iduser:codu,iddoc:codd,fechaini:fechacal,fechafin:fechacalfin})
    .done(function(json) {
        for (var i = 0; i < json.length; i++) {
            var object = json[i];
            j=0;
            for(property in object){
                result[j] = object[property];
                // This alerts "id=1", "created=2010-03-19", etc..
                j++;
            }
            respuesta =result[0];
        }
    });

    if(reservadaok==0){
        alert("La reserva no se pudo efectuar para las fechas seleccionadas");
        location.reload();
    }
    if(reservadaok==1){
        alert("Reserva realizada correctamente hasta el día: "+fechacalfin);
        location.reload();
    }
```



Esta llamada al php es algo distinta, ya que se le pasan unos argumentos al método. El método php es igual que la mayoría salvo que en lugar de hacer un select de una tabla, hace un insert de los datos pasado en la tabla de préstamos. De esta forma el método mostrado arriba envía los datos de código usuario, código documento, fecha inicio y fecha fin (fecha inicio +15 días) al método y se espera a que se tramite la petición. En función de la respuesta del php luego se muestra el aviso de si se ha podido realizar o no la reserva.

### 5.3.4 Descargar documento

Esta sección se vuelve a centrar en el apartado de información principal de la ventana de un documento. Ahora se va a explicar la funcionalidad de la alternativa a los documentos físicos, los documentos descargables.

Como ya se ha comentado antes la forma en la que se diferencia documentos online de físicos se va a explicar cómo actúa el sistema ante estos documentos. Lo primero que se tiene que saber es que cada documento contiene un atributo en llamado online. Este atributo contiene la url de Dropbox donde se encuentra el documento en cuestión.

A la hora de crear la ficha de los datos está ahora mostrará un botón de descargar, que simplemente hace referencia a la url de Dropbox del archivo a descargar. Una vez el usuario pulsa en este botón se redirigirá a la página de descarga. El usuario podrá realizar tantas descargas como quiera de cada documento.

### 5.3.5 Recomendar documento

Esta funcionalidad consiste en añadir un pequeño botón que permite al usuario compartir un documento a través de sus redes sociales o mediante correo electrónico. Este botón se muestra en la sección de información principal de un documento y al pulsarlo muestra una lista con los diferentes métodos de recomendación (Twitter, Facebook, Google +, Pinterest o email).

JQuery Mobile no tiene ninguna función desarrollada parecida a esta, por lo que ha sido necesario hacer uso de un plugin externo [21]. Para hacer uso de él se tienen que asociar a la página las librerías de javascript y las hojas de estilo de este plugin.

A continuación se añade la llamada al plugin en la creación de los datos del documento mediante las siguientes líneas de código:

```
<div class='share-button share-button-top'></div><p></p> </a></li> <p></p>  
<script>var share_button_top = new Share('.share-button-top', {title: 'Share Button  
Multiple Element
```

```
Test',url:'http://www.casiteka.com/compartirDoc.html?codd="+codd+"', ui: {flyout: 'top center'}});"
```

Aquí se aprecia cómo se crea un *div* con la nomenclatura especificada por el plugin. Posteriormente mediante un script se definen las características principales del botón, entre las que se pueden apreciar el título, la url que se compartirá y un estilo de interfaz.

Cuando el usuario pulse sobre el botón “SHARE” entonces tendrá que elegir la opción donde quiere compartirlo y el resto del proceso ya se llevará a cabo en la propia red social.

### 5.3.6 Opinar sobre un documento

Esta sección es la tercera que se muestra en la ventana de un documento. Está al final de la página y simplemente muestra una lista con los comentarios de todos los usuarios sobre el documento que estamos visualizando.

Esta sección ha sido muy sencilla de implementar ya que sigue la misma estructura que la consulta de los datos de un documento pero sobre la tabla de comentarios. El proceso lee el código del documento abierto y carga sobre la lista todos los comentarios cuyo identificador de documento sea igual al que se está visualizando.

Una vez hecho esto el usuario puede dejar un comentario mediante el formulario que se muestra al final de esta sección. Este formulario es un único campo de texto que el usuario deberá rellenar antes de pulsar en “Comentar” o el sistema le indicará que debe introducir un comentario.

Cuando se pulsa en comentar, se ejecuta otro método php que hace una consulta de insert sobre la tabla “comentarios” y añade el comentario junto con el código del documento y el código del usuario que lo ha hecho.

### 5.3.7 Información sobre bibliotecas

Previamente a la visualización de los datos de una biblioteca el usuario antes verá un listado con todas las bibliotecas de la universidad y deberá elegir una. Este listado es prácticamente igual al listado de documentos, por lo que no se va a entrar en detalles de implementación sobre esto.

El objetivo de esta sección es comentar la implementación de la información de las bibliotecas. La carga de toda la información también es muy parecida a la carga de la



información de los documentos, con el extra de que en las bibliotecas se dispone de un Google Map.

Este Google Map ha sido creado mediante el acceso a la aplicación de navegador. Tan sólo se ha tenido que buscar cada biblioteca en Google Maps y pulsar en el botón situado en la zona inferior derecha de compartir o insertar mapa.



ILUSTRACIÓN 35: COMPARTIR MAPA

A continuación se muestra la siguiente ventana:

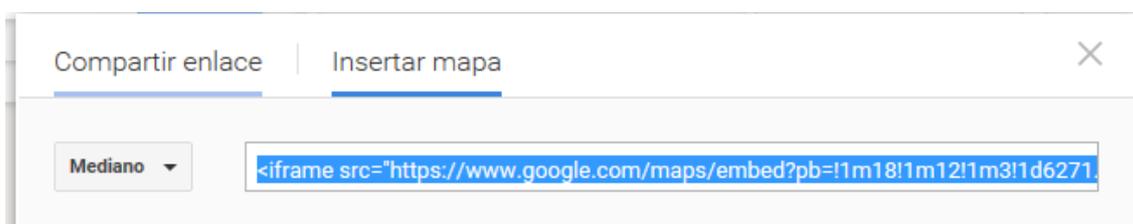


ILUSTRACIÓN 36: INSERTAR MAPA

En este caso debemos clicar sobre “insertar mapa” y copiar la línea de código html que generará Google Maps automáticamente. Esta línea se guarda en la base de datos para cada biblioteca y luego solo se tiene que añadir por completo en la aplicación al leer la información de cada biblioteca y se mostrará un mapa dinámico mostrando la ubicación de la biblioteca.



ILUSTRACIÓN 37: MAPA

### 5.3.7 Contacto

La sección de contacto es la más simple visualmente y lógicamente de todas. Pero esto no conlleva que haya sido fácil su implementación.

Para comenzar la sección consta de 4 campos de texto que componen el formulario que compondrá el mensaje que se enviará a los administradores. Situado más abajo se encuentra el botón que da funcionalidad a esta página.

Una vez se pulsa el botón se accede al método php que tramita los datos y envía el email. En este paso se podían seguir diferentes opciones. La más sencilla era utilizar la ya creada por javascript que permite hacer uso de Microsoft Outlook. Esta opción se descartó, ya que el uso de Outlook en dispositivos móviles no está muy expandido debido a que Google incluye Gmail en todos los dispositivos de Android como gestor principal de correo.

Es aquí donde entra en juego el php puro. A través de la siguiente función se consigue enviar un email.

```
$to = "administrator@casiteka.com";
$subject = $asunto;
$message = $correo;
$headers = 'From: ' . $nombre . "\r\n" .
'Reply-To: ' . $email . "\r\n" .
'X-Mailer: PHP/' . phpversion();

$variable=mail($to, $subject, $message, $headers);
```

Las variables \$asunto, \$correo, \$nombre y \$email se crean a partir de los datos introducidos por el usuario en los campos del formulario de la sección contacto. Este crea una cadena de texto a partir de estos datos y los envía a través del método *mail* que devuelve una variable booleana en función de si se ha podido enviar o no. Así en función de esta variable podemos indicarle al usuario si su email ha sido enviado correctamente o no.



## 6. Pruebas

La fase de pruebas, llevada a cabo conjuntamente con el desarrollo de todas las funcionalidades del proyecto, es una parte muy importante. Con esta fase nos aseguramos que todas las funcionalidades funcionan correctamente en tiempo de desarrollo ya que las se iban ejecutando cada vez que se completaba una funcionalidad. Esto además permite ver vulnerabilidades de la aplicación y posibles mejoras leves. Debido a esto ahora se mostrarán las pruebas realizadas para testear cada caso de uso y junto a ella el resultado de las mismas.

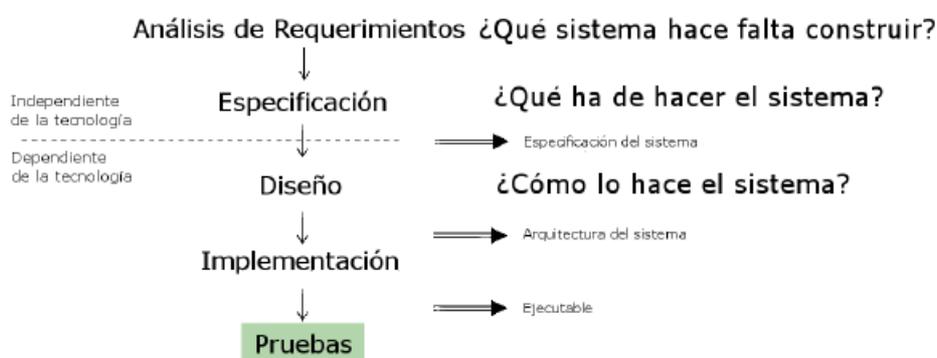


ILUSTRACIÓN 38: ETAPAS DE DESARROLLO SOFTWARE (PRUEBAS)

A continuación se describen brevemente las comprobaciones realizadas durante la fase de implementación para cada una de las funcionalidades de la aplicación. Para describir el resultado de la misma se usarán los símbolos (✓) y (✗)

### 6.1. INICIO DE SESIÓN

<b>Mensaje de aviso cuando usuario y/o contraseña incorrectos</b>	✓
<b>Autenticación y acceso al menú</b>	✓

## 6.2. RESERVAR DOCUMENTO

Listado de documentos correcto	✓
Carga de los datos del documento exitosa	✓
Gestión de fechas a la hora de hacer la reserva	✓
Préstamo realizado correctamente	✓
Mensajes de error y/o confirmación	✓
Descarga de documento en caso de que sea online	✓

## 6.3. BUSCAR DOCUMENTO

Éxito de búsqueda por libros	✓
Éxito de búsqueda por audio	✓
Éxito de búsqueda por revistas	✓
Éxito de búsqueda por docs online	✓
Éxito de búsqueda por novedades	✓
Éxito de búsqueda por destacados	✓
Éxito de búsqueda por palabra clave	✓

## 6.4. DEMANDAR DOCUMENTO

Petición tramitada correctamente	✓
Acceso solo disponible a profesores	✓
Actualización del estado en la petición	✓

## 6.5. CANCELAR RESERVA

Reservas mostradas en la sección de usuario	✓
Tramitación de cancelaciones correcta	✓

## 6.6. RECOMENDAR DOCUMENTO

Correcto funcionamiento del plugin	✓
Publicación en twitter	✓
Publicación en facebook	✓
Publicación en pinterest	✓
Publicación en en google +	✓



#### 6.7. OPINAR SOBRE UN DOCUMENTO

<b>Mostrar lista de comentarios</b>	✓
<b>Comprobación comentario vacío</b>	✓
<b>Publicación del comentario</b>	✓

#### 6.8. CONSULTAR BIBLIOTECA

<b>Mostrar lista de bibliotecas</b>	✓
<b>Mostrar datos de la biblioteca seleccionada</b>	✓
<b>Correcto funcionamiento del Google map</b>	✓

#### 6.9. CONTACTO

<b>Comprobación de campos del formulario</b>	✓
<b>Envío correcto del email</b>	✓
<b>Aviso al usuario y redirección al menú</b>	✓

# 7. Conclusiones

Este capítulo contiene revisiones y opiniones sobre todo el trabajo realizado.

Primero se realizará una revisión de los objetivos iniciales marcados, seguida de las desviaciones y problemas que se han producido durante el desarrollo de la aplicación con respecto a lo planificado inicialmente. Finalmente se concluirá el contenido principal de esta memoria con una valoración personal y algunas posibles mejoras del sistema.

## 7.1. REVISIÓN DE LOS OBJETIVOS

Es importante ver si se han cumplido con los objetivos iniciales del proyecto, una vez acabado el desarrollo, por lo que se volverán a nombrar cada uno de ellos junto con una breve explicación de los resultados obtenidos.

- Inicio de sesión para cada uno de los usuarios, de forma que se distingan los distintos usuarios y así ofrecer distintas funcionalidades en función de ellos.

**La aplicación cumple con las expectativas iniciales. Mediante el sistema de login creado se garantiza el acceso a todos los usuarios almacenados en el sistema y la gestión de los mismos en función de su rol.**

- Gestión de los documentos del usuario mediante una interfaz que resuma el estado de sus reservas o peticiones.

**Se ofrece al usuario una sección privada en la que pueda comprobar el estado de sus reservas o peticiones, así como la posibilidad de cancelar las reservas a futuro.**

- Sistema que gestione de manera coherente las reservas de documento, así como la descarga de los documentos online y opiniones de los usuarios.

**Se ha implementado una sección de información de un documento de acuerdo a todo lo especificado al principio, permitiendo reservas, descargas, recomendaciones y opiniones.**



Desarrollo de aplicaciones web para dispositivos móviles con JQuery mobile

- Sistema que gestione información de las bibliotecas de la universidad y ofrezca su información a los usuarios.

**Se ha desarrollado una ventana donde el usuario puede ver toda la información referente a cada biblioteca.**

- Posibilidad de demanda de documentos y zona de contacto.

**Se permite al usuario contactar con los administradores mediante correo y a los profesores solicitar nuevos documentos para las bibliotecas.**

- La aplicación debe tener una interfaz simple e intuitiva orientada a las pantallas móviles, además de mostrar mensajes de error al usuario.

**Gracias al uso de jQuery Mobile se ha logrado una interfaz simple e intuitiva, a la que el usuario se acostumbrará en el primer uso fácilmente.**

- La aplicación contará con un sistema de facilidades a la hora de encontrar los fallos en las pruebas.

**Durante la fase de desarrollo se iban poniendo *alerts* de forma estratégica para ver en qué parte del proceso fallaba algún método o la comunicación entre páginas.**

- Se llevará a cabo comprobación de los campos a la hora de validar formularios.

**Se ha elaborado un sistema de alertas al usuario en caso de que no haya completado todos los datos mínimos en cada formulario y se bloquea la acción hasta que estos hayan sido correctamente completados.**

Como revisión final se puede asegurar que se ha cumplido con todos los objetivos propuestos.

## 7.2. PROBLEMAS EN EL DESARROLLO

En el análisis del proyecto se realizó una estimación de las tecnologías a utilizar. Debido a que no se tenía mucha experiencia en algunas o simplemente eran desconocidas, se han encontrado algunos problemas que han alargado el tiempo estimado del proyecto.

A continuación se mostrarán estos problemas y como han conseguido solventarse.

### 7.2.1 Uso de PHP

Antes de empezar el proyecto no se tenía experiencia previa ninguna con PHP. Esto conllevó que a la hora de desarrollar los primeros métodos como el login, o algunas consultas para mostrar el listado de documentos se perdiera mucho tiempo.

Las mayores dificultades fueron a la hora de empaquetar los resultados. Al principio se usaba el método `json_encode()`. Este método convertía un array a formato JSON, pero no era muy efectivo, ya que si alguna variable superaba una cierta longitud el método fallaba por algún extraño motivo y al final no se devolvía nada.

Debido a esto se consultó con otras personas como poder hacer este tipo de consultas y al final se optó por la opción de crear un string en formato JSON manualmente en el fichero PHP. De esta forma funciona correctamente y así justificamos la elección de una aplicación web, ya que debido a la gran comunidad de usuarios que conoce todos estos lenguajes nos resulta fácil encontrar una solución como se preveía antes de comenzar el proyecto.

### 7.2.2 Configuración servidor

Antes de comenzar el proyecto se tenía pensado hospedar la aplicación en un servidor XAMPP en el ordenador donde se programaba. Esta opción se descartó muy pronto ya que conllevaba configurar PHP 5 y habilitar algunas opciones de seguridad que no se lograba entender bien porque no funcionaba. Además de esto conllevaba la instalación de algún tipo de programa que hiciera público nuestro servidor con una url.

Para evitar todos estos problemas y pérdidas de tiempo se decidió contratar un dominio como el que ofrecía One.com. Además como ya se ha explicado antes, esto nos ofrecía multitud de ventajas y ya venía configurado para subir el *index* al servidor y arrancar la aplicación.

De esta forma se ahorró tiempo y trabajo, que luego vino bien para depurar mejor el código.

### 7.2.3 Mantener sesión de usuarios y comunicación entre páginas

La forma de mantener una “sesión” de usuarios fue uno de los principales problemas desde el principio. Como también se ha comentado antes habían múltiples opciones y la más fácil era la de utilizar PHP SESSION.

Esta opción se descartó pronto debido al gran desconocimiento que se tenía sobre PHP y de cómo integrarlo en HTML. Entonces se contempló la opción de pasar el código de usuario mediante la barra de direcciones para así poder leerlo en cada página y poder mantener una especie de sesión.



Como después haría falta pasar más códigos como el de documentos a la hora de ver información de un documento esta opción se estimó como la más indicada. Así no solo podríamos mantener la información de los usuarios, si no que podríamos mantener información de documentos, bibliotecas...

El gran inconveniente de este mecanismo es la seguridad, ya que si alguien sabe cómo se estructura la url con los parámetros podría acceder a información sin permisos. En este caso se propone una mejora del sistema más adelante.

### 7.2.4 Envío de mails en sección contacto

El envío de emails parecía algo muy sencillo en la planificación ya que se pensaba que ya habría una función en jQuery predefinida para esto. La sorpresa surgió cuando la única función programada en las librerías de jQuery necesitaba que Microsoft Outlook estuviera instalado en el dispositivo, ya que la función preparaba el correo pero no lo enviaba sino que lo abría en Outlook para que el usuario pulsara en enviar.

Esta opción no parecía correcta ya que en dispositivos Android se suele utilizar el gestor de correos Gmail. Así que tocó investigar en otra forma de enviar correos y tras buscar en la red y probar distintos códigos finalmente se encontró un método php capaz de enviar un correo electrónico en función de los parámetros que se le pasaran haciendo uso de la función *mail* de php.

Una solución sencilla y poco extensa en código pero que costó de entender debido a que se le pasaban algunas variables mediante GET y otras mediante POST y no funcionaba correctamente, por lo que se le pasaron todas mediante GET y funcionó la prueba.

## 7.3. MEJORAS

A continuación se procede a describir algunas mejoras que se podrían realizar sobre la aplicación, haciendo así a la **Casiteka** de un producto más funcional y seguro.

Todos los puntos descritos no se han podido llevar a cabo por una limitación clara como el tiempo. Está claro que una aplicación nunca es perfecta en su primera versión y siempre están en continua actualización para ofrecer al usuario más funcionalidades y corregir fallos. Por eso se eligió una aplicación web, debido a su facilidad para instaurar las actualizaciones del sistema de forma transparente al usuario.

Las posibles mejoras son las siguientes:

- Mejoras de seguridad mediante el paso de parámetros por url. Cifrar los datos antes de escribirlos en la url para evitar que los usuarios sepan datos confidenciales y no se pueda intentar entrar a la aplicación de forma fraudulenta.

- Añadir un módulo para los auxiliares de bibliotecas. Como se estudió en el análisis de stakeholders los auxiliares son los que mantendrán el sistema actualizado, así que sería deseable programar un módulo para que les resulte mucho más sencillo gestionar la aplicación sin tener que entrar al phpMyAdmin y modificar la base de datos en lenguaje SQL.
- Convertir las operaciones de reservas de documentos en atómicas. Puede darse el caso de que dos usuarios pulsen en reservar el mismo documento a la vez y a lo mejor se procesen las dos reservas en lugar de descartar una. Este punto se debería estudiar en el futuro y ver cómo reacciona el sistema para ver si es necesaria una corrección.
- Mejora en el tiempo de carga mediante algoritmos más eficientes en los métodos php.
- Crear una interfaz nueva, diseñada por un diseñador gráfico.

#### 7.4. VALORACIÓN

La **Casiteka** ha sido una experiencia increíble que ha aumentado en gran cantidad mis conocimientos sobre los distintos tipos de aplicaciones para dispositivos móviles, así como las tecnologías que permiten implementarlas.

Previamente ya había desarrollado dos aplicaciones específicas solo para Android mediante el SDK de Android para Eclipse. Si bien la programación de las comunicaciones es mucho más potente y segura, el desarrollo de interfaces se queda muy pobre en este tipo de programación. La programación de estas dos aplicaciones fue básicamente en java y xml para definir la lógica y la interfaz respectivamente. Mientras que con la **Casiteka** se han tocado muchos más lenguajes de programación y me ha servido para profundizar en las tecnologías móviles y en lenguajes como HTML.

También ha supuesto mi primer paso en el aprendizaje de PHP, que actualmente es el mayor lenguaje de cara a producir aplicaciones de comercio electrónico y una herramienta muy potente para desarrollar cualquier otro tipo de aplicación con un poco más de práctica.

Focalizándonos en la **Casiteka** cabe mencionar que me ha parecido muy interesante el ámbito de que la aplicación pueda servir a la comunidad de estudiantes, profesores y demás personal de la UPV. Cuando me propusieron una aplicación basada en jQuery Mobile de complejidad media como una agenda electrónica o un portal web para el comercio electrónico, lo primero que se me vino a la cabeza fue la idea de la **Casiteka**. De esta forma elaboraría un proyecto que me serviría a mí como método de aprendizaje y a los demás si se decide implementar como método de gestión de las bibliotecas y reservas.

También se merecen una mención especial algunas asignaturas de la carrera que me han servido de gran ayuda para poner en práctica los conocimientos aprendidos gracias a ellas. Puede ser el caso de “Desarrollo de bases de datos” para hacer todas las consultas a la base de datos o “Desarrollo centrado en el usuario” para pensar cómo hacer las interfaces lo más sencillas de cara al usuario, otras como “Ingeniería del software” y “Gestión de proyectos software” para la elaboración de la memoria y la metodología elegida para desarrollar el proyecto, y así con muchas más.

Finalmente para concluir me gustaría añadir que me siento muy orgulloso del trabajo realizado y de cómo lo he realizado. Está claro que si hubiera sabido mucho más sobre los lenguajes y técnicas de desarrollo se podría haber conseguido una aplicación mucho más elaborada, pero mi objetivo ahora y siempre será aprender y esta aplicación es uno de los primeros pasos y de los más importantes en el resto de mi vida laboral.

# 8. Bibliografía

En este capítulo se recogen las distintas fuentes y referencias que han servido de apoyo para desarrollar el proyecto. Dividiremos la bibliografía dividida en dos partes. La primera está compuesta por la bibliografía referenciada a lo largo de la memoria y la segunda es la que hace referencia a material utilizado en general pero no de forma específica en algún punto.

## 8.1. Bibliografía referenciada

- [1] España primer país europeo en el uso de smartphones  
<http://www.rtve.es/noticias/20150121/espana-primer-pais-europeo-uso-smartphones-2014-81/1084704.shtml>
- [2] Metodologías del desarrollo software  
[https://es.wikipedia.org/wiki/Metodolog%C3%ADa\\_de\\_desarrollo\\_de\\_software](https://es.wikipedia.org/wiki/Metodolog%C3%ADa_de_desarrollo_de_software)
- [3] Requisitos funcionales y no funcionales  
<http://es.slideshare.net/Lismirabal/requerimientos-funcionales-y-no-funcionales>
- [4] Comparativas de sistemas operativos  
<https://blog.uchceu.es/informatica/ranking-de-sistemas-operativos-mas-usados-para-2015/>
- [5] Android Studio  
<https://developer.android.com/sdk/index.html>
- [6] Apple Developer  
<https://developer.apple.com/ios/download/>
- [7] Windows Phone 8.1  
<https://dev.windows.com/en-us/develop/download-phone-sdk>
- [8] BlackBerry Developer  
<https://developer.blackberry.com/>
- [9] Notificaciones Push  
<http://javiercancela.com/2009/07/13/notificaciones-push-en-dispositivos-moviles/>
- [10] Frameworks híbridos  
<http://necsia.es/frameworks-multiplataforma-para-el-desarrollo-movil/#>
- [11] OpenGL  
<http://academiaandroid.com/graficos-en-android-descripcion-opengl/>



- [12] Información general sobre jQuery Mobile  
[https://es.wikipedia.org/wiki/JQuery\\_Mobile](https://es.wikipedia.org/wiki/JQuery_Mobile)
- [13] Página oficial del proyecto jQuery  
<http://jquery.com/>
- [14] Página oficial de One.com  
<http://www.one.com/es/>
- [15] Geolocalización de la Casiteka  
<http://websitehop.com/www/casiteka.com#>
- [16] Documentación adicional técnica sobre jQuery Mobile  
<http://www.nsolver.com/noticias/aplicacion-movil-jquery.html>  
[http://www.w3schools.com/jquerymobile/jquerymobile\\_get\\_started.asp](http://www.w3schools.com/jquerymobile/jquerymobile_get_started.asp)
- [17] Manuales de PHP  
<http://php.net/manual/es/intro-whatism.php>
- [18] Descripción del estándar JSON  
<https://es.wikipedia.org/wiki/JSON>
- [19] Carpeta public de Dropbox  
<https://www.dropbox.com/es/help/16>
- [20] Datepicker plugin externo  
<http://dev.jtsage.com/jQM-DateBox/>
- [21] ShareButton plugin externo  
<https://github.com/carrot/share-button>

## 8.2. Bibliografía no referenciada

- [1] Maximiliano Firtman (O'Reily 2012) *jQuery Mobile: Up and Running*
- [2] Jon Reid (O'Reily 2011) *jQuery Mobile*
- [3] Mark Pilgrim (O'Reily & Google Press 2010) *HTML5 Up and Running*
- [4] Raymond Camdem & Andy Matthews (Packt Publishing 2012) *jQuery Mobile Web Development Essentials*
- [5] David Flanagan (O'Reily 2011) *JavaScript: The Definitive Guide v6*



- [6] Robin Nixon (O'Reily 2009) *Learning PHP, MySQL and JavaScript*
- [7] Consultas MySQL con PHP  
<http://www.taringa.net/posts/hazlo-tu-mismo/14629402/Tutorial-Crear-un-sistema-de-Login-con-PHP-y-MySQL.html>  
<http://www.emiliort.com/2011/05/login-con-php-y-mysql.html>  
<http://enatorv.blogspot.com.es/2013/07/hacer-login-con-php-y-mysql.html>  
[http://www.tutorialspoint.com/php/mysql\\_insert\\_php.htm](http://www.tutorialspoint.com/php/mysql_insert_php.htm)  
<http://stackoverflow.com/questions/12120433/php-mysql-insert-date-format>
- [8] Comunicación mediante parámetros en url  
<http://www.php.net/manual/es/tutorial.firstpage.php>  
<http://stackoverflow.com/questions/2383401/javascript-setting-window-location-href-versus-window-location>  
<http://www.anieto2k.com/2006/08/17/coge-los-parametros-de-tu-url-con-javascript/>  
<http://blog.aplicacionesweb.cl/opensource/leer-parametros-de-la-url-con-javascript/>  
<http://www.webestilo.com/foros/mensaje.phtml?foro=3&msg=43571>
- [9] Enviar emails con php  
<http://stackoverflow.com/questions/5335273/how-to-send-an-email-using-php>
- [10] Operaciones sobre variables en format DATE  
<http://reviblog.net/2014/07/01/sumar-dias-a-una-fecha-en-javascript/>  
<https://advenis.wordpress.com/2014/05/28/comparar-dos-fechas-en-javascript/>  
<http://stackoverflow.com/questions/1531093/how-to-get-current-date-in-javascript>
- [11] Algoritmo de búsqueda  
<http://www.desarrolloweb.com/articulos/2087.php>
- [12] Tipos de aplicaciones móviles  
<http://appio.es/tipos-de-apps/>
- [13] Información sobre sistemas operativos móviles  
<http://www.tecnologiadetuatu.elcorteingles.es/telefonía/sistemas-operativos-para-moviles-en-la-variedad-esta-el-gusto/>  
<http://www.pactual.com/articulo/laboratorio/comparativas/13412/ponemos-nota-los-sistemas-operativos-moviles-mas-populares.html>

## 9. Apéndice

### 9.1 Descripción BBDD

#### 9.1.1.- ANÁLISIS DE REQUISITOS Y DESCRIPCIÓN DEL SISTEMA

La base de datos que pretendemos realizar debe gestionar todo lo referente a la biblioteca de la UPV

Esta biblioteca será usada por tres tipos distintos de usuarios.

La modalidad de trabajo es siempre la misma. El usuario abre la aplicación y puede consultar el catálogo de libros sin poder reservar. Si se loguea entonces accede a su página personal en la que puede gestionar sus reservas o hacer nuevas reservas de cualquier documento en la biblioteca. Si el usuario es profesor o auxiliar de biblioteca además puede pedir nuevos documentos a la biblioteca. El auxiliar además puede gestionar todas las peticiones, el inventario, los préstamos y las penalizaciones.

Cuando un usuario se loguea, la aplicación debe saber que tipo de usuario es para mostrarle una u otra interfaz. Para ello la tabla usuarios tendrá nombre, apellidos, edad y rol de cada usuario. Con el rol podremos diferenciar el tipo de usuarios que son. Además, a cada usuario se le asignará un Código identificador único (CODU).

Cada documento debe estar guardado en la base de datos junto a cierta información relevante a él. Esta información consiste en título, autor, fecha de publicación, editorial, tipo, páginas, lengua, ISBN, sinopsis, puntos, votos, valoración e imagen. Además cada documento tendrá un Código de Documento único (CODD). También dispondrá del enlace del documento en caso de que sea un documento digital y pueda ser descargado.

Para gestionar los préstamos se necesita otra tabla, ya que se requieren varios atributos para gestionar un préstamo. Estos atributos son: CODU y CODD, que referencian al usuario y al documento respectivamente, fecha inicio y fecha fin. Además para hacer los préstamos únicos se necesitará un Código identificador (CODP).

Los usuarios pueden dar su opinión referente al documento mediante una sección que recoge comentarios. Esto es posible gracias al campo opinión y el CODU y CODD, que referencian al usuario y al documento relacionados con el comentario. Además cada comentario tendrá un código que lo identifique (CODC).

Si un usuario se retrasa en la devolución de algún documento debe ser penalizado durante un periodo de tiempo. Para llevar a cabo esta gestión se

necesitará el CODU, referente al usuario, el tipo de penalización y las fechas de inicio y fin. Además se establece un Código que identifica cada penalización de forma única (CODZ).

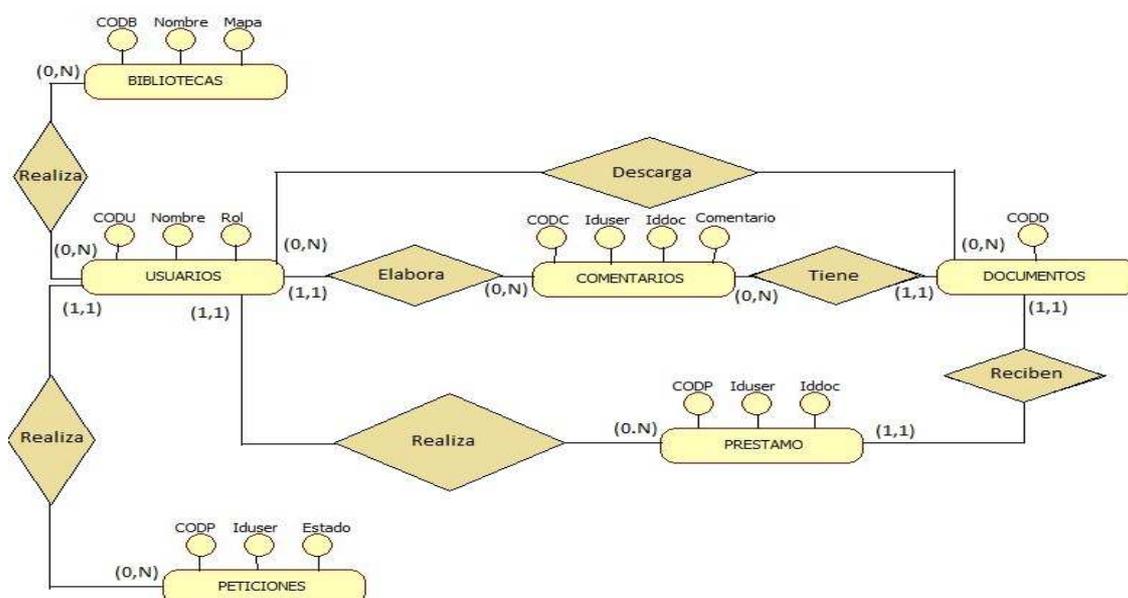
Finalmente para gestionar las peticiones que realicen los profesores y/o auxiliares se necesita otra tabla en la que se almacene cada petición. Para ello se necesitan el CODU, referente al usuario que realiza la petición, el título, el autor, año de publicación y editorial del libro solicitado. También será necesario un campo estado en el que se podrá comprobar si está pendiente de resolución la petición o ha sido aceptada o no. Cada petición tendrá un código único (CODT).

### 9.1.2.- ENTIDADES Y ATRIBUTOS

- **USUARIO:** CODU<sup>(1)</sup>, nombre, apellidos, rol, edad, username, password.
- **DOCUMENTOS:** CODD<sup>(2)</sup>, titulo, autor, fecha, editorial, paginas, tipo, lengua, ISBN, sinopsis, puntos, votos, valoracion, imagen, online, fechaInsert.
- **PRESTAMOS:** CODP<sup>(4)</sup>, IDdoc<sup>(2)</sup>, IDuser<sup>(1)</sup>, fechaIni, fechaFin.
- **COMENTARIOS:** CODC<sup>(5)</sup>, IDdoc<sup>(2)</sup>, IDuser<sup>(1)</sup>, opinion.
- **PETICIONES:** CODN<sup>(7)</sup>, IDuser<sup>(1)</sup>, titulo, autor, ano, editorial, estado.
- **BIBLIOTECAS:** CODB<sup>(8)</sup>, nombre, horarios, imagen, descripcion, telefono, dirección, mapa.



### 9.1.3.- DIAGRAMA ENTIDAD-RELACIÓN



### 9.1.4.- DISEÑO LÓGICO

	N	Columna	Tipo	Anchura	Escala	Restricción
<b>USUARIO</b>	1	CODU	INT	9		PRIMARY KEY AUTOINCREMENT
	2	NOMBRE	VARCHAR	50		NOT NULL
	3	APELLIDOS	VARCHAR	50		NOT NULL
	4	EDAD	INT	3		>0
	5	ROL	INT	1		NOT NULL
	6	PASSWORD	VARCHAR	30		NOT NULL
	7	USERNAME	VARCHAR	30		NOT NULL

Tabla 1. Tabla Usuarios.

	N	Columna	Tipo	Anchura	Escala	Restricción
DOCUMENTOS	1	CODD	INT	9		PRIMARY KEY AUTOINCREMENT
	2	TITULO	VARCHAR	30		NOT NULL
	3	AUTOR	VARCHAR	50		NOT NULL
	4	ANO	INT	4		NOT NULL
	5	EDITORIAL	VARCHAR	15		
	6	PAGINAS	INT	5		
	7	TIPO	VARCHAR	15		
	8	LENGUA	VARCHAR	15		
	9	ISBN	CHAR	13		
	10	SINOPSIS	VARCHAR	255		
	11	PUNTOS	INT	10		
	12	VOTOS	INT	10		
	13	VALORACION	DOUBLE	2	2	
	14	IMAGEN	VARCHAR	255		
	15	ONLINE	VARCHAR	255		
	16	FECHAINsert	DATE			NOT NULL

Tabla 2. Tabla Documentos.

	N	Columna	Tipo	Anchura	Escala	Restricción
COMENTARIOS	1	CODC	INT	9		PRIMARY KEY AUTOINCREMENT
	2	IDDOC	INT	9		NOT NULL FOREIGN KEY
	3	IDUSER	INT	9		NOT NULL FOREIGN KEY
	4	COMENTARIO	VARCHAR	255		NOT NULL

Tabla 3. Tabla Comentarios.

	N	Columna	Tipo	Anchura	Escala	Restricción
PRESTAMOS	1	CODP	INT	9		PRIMARY KEY AUTOINCREMENT
	2	IDDOC	INT	9		NOT NULL FOREIGN KEY
	3	IDUSER	INT	9		NOT NULL FOREIGN KEY
	4	FECHAINI	DATE			NOT NULL
	5	FECHAFIN	DATE			NOT NULL

Tabla 4. Tabla Prestamos.



	N	Columna	Tipo	Anchura	Escala	Restricción
<b>PETICIONES</b>	1	CODN	INT	9		PRIMARY KEY AUTOINCREMENT
	2	IDUSER	INT	9		NOT NULL FOREIGN KEY
	3	TITULO	VARCHAR	30		NOT NULL
	4	AUTOR	VARCHAR	50		NOT NULL
	5	ANO	INT	4		NOT NULL
	6	EDITORIAL	VARCHAR	15		NOT NULL
	7	ESTADO	VARCHAR	15		NOT NULL

Tabla 5. Tabla Peticiones.

	N	Columna	Tipo	Anchura	Escala	Restricción
<b>BIBLIOTECAS</b>	1	CODB	INT	9		PRIMARY KEY AUTOINCREMENT
	2	NOMBRE	VARCHAR	70		NOT NULL
	3	HORARIOS	VARCHAR	255		
	4	IMAGEN	VARCHAR	255		
	5	DESCRIPCION	VARCHAR	255		
	6	TELEFONO	VARCHAR	15		
	7	DIRECCION	VARCHAR	120		
	8	MAPA	VARCHAR	500		

Tabla 6. Tabla Bibliotecas.

## 9.1.5.- CREACIÓN DE TABLAS

### 9.1.5.1.- USUARIO

```
CREATE TABLE usuario (  
    codu INT(9) primary key auto_increment,  
    nombre VARCHAR(50) not null,  
    apellidos VARCHAR(50) not null,  
    edad INT(3) CHECK(edad>0),  
    rol DOUBLE(1) not null,  
    username VARCHAR(30) not null,  
    password VARCHAR(30) not null  
);
```

### 9.1.5.2.- DOCUMENTOS

```
CREATE TABLE documentos (  
    codd INT(9) primary key auto_increment,  
    titulo VARCHAR(30) not null,  
    autor VARCHAR(50) not null,  
    ano INT(4) not null,  
    editorial VARCHAR(15),  
    paginas INT(5),  
    tipo VARCHAR(15) ,  
    lengua VARCHAR(15) ,  
    isbn CHAR(13) ,  
    sinopsis VARCHAR(255),  
    puntos INT(10),  
    votos INT(10),  
    valoracion DOUBLE(4,2),  
    imagen VARCHAR(255),  
    online VARCHAR(255),  
    fechaInsert DATE not null  
);
```

### 9.1.5.3.- COMENTARIOS

```
CREATE TABLE comentarios (  
    codc INT(9) primary key auto_increment,  
    iddoc INT (9) not null references documentos,  
    iduser INT (9) not null references usuario,  
    comentario VARCHAR(255)  
);
```



#### **9.1.5.4.- PRESTAMOS**

```
CREATE TABLE prestamos (  
    codp INT(9) primary key auto_increment,  
    iddoc INT (9) not null references documentos,  
    iduser INT (9) not null references usuario,  
    fechaini DATE not null,  
    fechafin DATE not null  
);
```

#### **9.1.5.5.- PETICIONES**

```
CREATE TABLE peticiones (  
    codn INT(9) primary key auto_increment,  
    iduser INT (9) not null references usuario,  
    titulo VARCHAR(30) not null,  
    autor VARCHAR(50) not null,  
    ano INT(4) not null,  
    editorial VARCHAR(15) not null,  
    estado VARCHAR(15) not null  
);
```

#### **9.1.5.6.- BIBLIOTECAS**

```
CREATE TABLE bibliotecas (  
    codb INT(9) primary key auto_increment,  
    nombre VARCHAR(70) not null,  
    horarios VARCHAR(255),  
    imagen VARCHAR(255),  
    descripcion VARCHAR(255),  
    telefono VARCHAR(15),  
    direccion VARCHAR(120),  
    mapa VARCHAR(500)  
);
```

## 9.1.6.- INSERCCIÓN DE LOS DATOS

### 9.1.6.1.- USUARIO

```
INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Jose', 'Casielles', 22, 0,
'casielles', 'casielles');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Raquel', 'Salcedo', 15, 0,
'salcedo', 'salcedo');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Osmani', 'García', 27, 0,
'garcia', 'garcia');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('German', 'Vidal', 40, 1,
'vidal', 'vidal');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Michael', 'Van Cassidy', 21, 0,
'cassidy', 'cassidy');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Luis', 'Juan', 57, 1, 'juan',
'juan');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Corinna', 'Randazzo', 23, 2,
'randazzo', 'randazzo');

INSERT INTO `usuario`(`nombre`, `apellidos`, `edad`, `rol`,
`username`, `password`) VALUES ('Axel', 'Hedfords', 34, 0,
'hedfords', 'hedfords');
```

### 9.1.6.2.- DOCUMENTOS

```
INSERT INTO `documentos`(`titulo`, `autor`, `ano`, `editorial`,
`paginas`, `tipo`, `lengua`, `isbn`, `sinopsis`, `puntos`,
`votos`, `valoracion`, `imagen`, `online`, `iduser`,
`fechaInsert`) VALUES ('El libro 1', 'Jack Ryan', '1995',
'casieles', 34, 'Libro', 'Español', 'lkjke', 'documentol', 0, 0,
0, ' ', ' ', ' ', '2015/04/25');

INSERT INTO `documentos`(`titulo`, `autor`, `ano`, `editorial`,
`paginas`, `tipo`, `lengua`, `isbn`, `sinopsis`, `puntos`,
`votos`, `valoracion`, `imagen`, `online`, `iduser`,
`fechaInsert`) VALUES ('On my way', 'Axwell /\ Ingresso', '2015',
'Def jam Records', 0, 'Audio', 'Inglés', '0', 'Best song of 2015',
10, 1, 10, '
', 'https://dl.dropboxusercontent.com/u/49796845/axwell--ingrosso-
on-my-way_mp3cofe.com.mp3', '2015/04/26');
```



```
INSERT
  INTO `documentos` ( `titulo` , `autor` , `ano` , `editorial` , `p
  aginas` , `tipo` , `lengua` , `isbn` , `sinopsis` , `puntos` , `vo
  tos` , `valoracion` , `imagen` , `online` , `iduser` , `fechaInser
  t` )
VALUES ('Dont you worry child', 'Swedish House Mafia', '2012',
'EMI', 0, 'Audio', 'Inglés', '0', 'Best song of 2012', 10, 1, 10,
'images/documentos/dontworrychild.jpg', 'https://dl.dropboxusercont
ent.com/u/49796845/Swedish%20House%20Mafia%20%E2%80%93%20Don%27t%2
0You%20Worry%20Child%20feat.%20John%20Martin%20%28Pete%20Tong%20Ra
dio%20Exclusive%2010.08.12%29%20-
%20officialswedishhousemafia.mp3', '2015/05/02'
);
```

```
INSERT INTO `documentos`(`titulo`, `autor`, `ano`, `editorial`,
`paginas`, `tipo`, `lengua`, `isbn`, `sinopsis`, `puntos`,
`votos`, `valoracion`, `imagen`, `online`, `iduser`,
`fechaInsert`) VALUES ('If I lose myself tonight','Alesso ft One
Republic' , '2013' , 'EMI' , 0, 'Audio', 'Inglés' , '0' , 'Best song of
2013' , 10, 1, 10, 'images/documentos/OneRepublic_Vs_Alesso.jpg'
, 'https://dl.dropboxusercontent.com/u/49796845/If_I_lose_my_self_t
onight.mp3', '2015/05/02');
```

```
INSERT
  INTO `documentos` ( `codd` , `titulo` , `autor` , `ano` , `editorial` ,
  `paginas` , `tipo` , `lengua` , `isbn` , `sinopsis` , `puntos` , `votos`
  , `valoracion` , `imagen` , `online` , `iduser` , `fechaInsert` )
VALUES (NULL , 'El código da Vinci', 'Dan Brown' , '2013'
, 'DoubleDay', '300' , 'Libro' , 'Castellano' , '84-95618-60-5'
, NULL, '7' , '3' , '7' , NULL , NULL , '2015-05-12');
```

```
INSERT
  INTO `casiteka_com`.`documentos` ( `codd` , `titulo` , `autor` , `ano`
  , `editorial` , `paginas` , `tipo` , `lengua` , `isbn` , `sinopsis` , `
  puntos` , `votos` , `valoracion` , `imagen` , `online` , `iduser` , `fec
  haInsert` ) VALUES (NULL , 'Los pilares de la Tierra', 'Ken
  Follet', '1989', 'Macmillan ' , '1300', 'Libro', 'Castellano', '0-
  333-51983-3', NULL , '9' , '2' , '9', NULL , NULL , '2015-05-12');
```

### 9.1.6.3.- COMENTARIOS

```
INSERT INTO `comentarios`(`codc`, `iddoc`, `iduser`, `comentario`)
VALUES (null,'2','1','Gran regreso de Axwell /\ Ingrosso como
duo');
```

```
INSERT INTO `comentarios`(`codc`, `iddoc`, `iduser`, `comentario`)
VALUES (null,'5','2','La mejor canción de Swedish House Mafia y un
gran colofón a toda su carrera');
```

```
INSERT INTO `comentarios`(`codc`, `iddoc`, `iduser`, `comentario`)
VALUES (null,'2','5','Troleito guapo');
```

#### 9.1.6.4.- PRESTAMOS

```
INSERT INTO `prestamos` (`codp`, `iddoc`, `iduser`, `fechaini`,  
`fechafin`) VALUES (NULL, '1', '1', '2015-06-03', '2015-06-25');
```

#### 9.1.6.5.- BIBLIOTECAS

```
INSERT INTO `bibliotecas` (`codb`, `nombre`, `horarios`, `imagen`,  
`descripcion`, `telefono`, `direccion`) VALUES (null,'Biblioteca central','Lunes a  
Viernes: de 8:00 a 21:45 - Sábados y domingos : de 8:30 a  
20:45',null,'Situada en Valencia, en el Campus de Vera, la Biblioteca Central es un  
edificio de 5.762 m2 repartidos en tres plantas, con más de 1.300 puestos de  
estudio. La Biblioteca Central alberga la mayor colección bibliográfica de la UPV.  
Su amplio horario e instalaciones ofrecen los servicios propios de un CRAI  
(Centro de Recursos para el Aprendizaje y la  
Investigación).','963877084','Edificio 4L Camino de Vera, s/n 46022 Valencia');
```

```
INSERT INTO `bibliotecas` (`codb`, `nombre`, `horarios`, `imagen`,  
`descripcion`, `telefono`, `direccion`) VALUES (null,'Biblioteca de ADE-  
Topografía','Lunes a Viernes: de 8:00 a 21:45- Sábados y domingos : de  
8:30 a 20:45',null,'La Biblioteca de Facultad de ADE - ETS Ingeniería Geodésica,  
Cartográfica y Topográfica forma parte de la red de bibliotecas de la UPV  
teniendo como objetivo facilitar el acceso a la información científica y cultural, la  
conservación, incremento y difusión de los fondos bibliográficos, documentales y  
audiovisuales de la Universidad.','963879926','Edificio 7J Camino de Vera, s/n  
46022 Valencia (Facultad de ADE)');
```

```
INSERT INTO `bibliotecas` (`codb`, `nombre`, `horarios`, `imagen`,  
`descripcion`, `telefono`, `direccion`) VALUES (null,'Biblioteca de  
Agroingeniería','Lunes a Viernes: de 8:00 a 21:45 - Sábados y domingos :  
de 8:30 a 20:45',null,'La biblioteca de la Escuela Técnica Superior de Ingeniería  
Agronómica y del Medio Natural, forma parte de la red de bibliotecas de la UPV, y  
tiene como objetivo facilitar el acceso a la información científica y cultural,  
mediante la conservación, incremento y difusión de los fondos bibliográficos,  
documentales y audiovisuales de la Universidad.','963877007','Edificio 3P  
Camino de Vera, s/n 46022 Valencia');
```

```
INSERT INTO `bibliotecas` (`codb`, `nombre`, `horarios`, `imagen`,  
`descripcion`, `telefono`, `direccion`) VALUES (null,'Biblioteca de  
Informática','Lunes a Viernes: de 8:00 a 21:45 - Sábados y domingos :  
de 8:30 a 20:45',null,'La función de la Biblioteca de Informática y  
Documentación, como parte integrante de la red de bibliotecas de la UPV, es  
facilitar el acceso a todos los medios de información científica o cultural; para  
ello, conserva, incrementa y difunde los fondos bibliográficos, documentales y  
audiovisuales de la Universidad.','963877007-78826','Edificio 1H Camino de  
Vera, s/n 46022 Valencia (Edificio Multiusos de Informática)');
```

```
INSERT INTO `bibliotecas` (`codb`, `nombre`, `horarios`, `imagen`,  
`descripcion`, `telefono`, `direccion`) VALUES (null,'Biblioteca de
```

Edificación', 'Lunes a Viernes: de 8:00 a 21:00', null, 'La biblioteca de la Escuela Técnica Superior de Ingeniería de Edificación forma parte de la red de bibliotecas de la UPV, y tiene como objetivo facilitar el acceso a la información científica y cultural', '963877120', 'Edificio 1B Camino de Vera, s/n 46022 Valencia');

```
INSERT INTO `bibliotecas`(`codb`, `nombre`, `horarios`, `imagen`, `descripcion`, `telefono`, `direccion`) VALUES (null, 'Biblioteca de Diseño', 'Lunes a Viernes: de 8:00 a 21:00', null, 'La biblioteca de la Escuela Técnica Superior de Ingeniería del Diseño, como parte integrante de la red de bibliotecas de la UPV, tiene como fin facilitar el acceso a todos los medios de información científica o cultural; para ello, conserva, incrementa y difunde los fondos bibliográficos, documentales y audiovisuales de la Universidad.', '963877000', 'Edificio 7B Camino de Vera, s/n 46022 Valencia (ETSID)');
```

```
INSERT INTO `bibliotecas`(`codb`, `nombre`, `horarios`, `imagen`, `descripcion`, `telefono`, `direccion`) VALUES (null, 'Biblioteca de Caminos', 'Lunes a Viernes: de 8:00 a 21:00', null, 'La Biblioteca de la Escuela Técnica Superior de Caminos Canales y Puertos forma parte de la red de bibliotecas de la UPV y tiene como objetivo facilitar el acceso a la información científica y cultural mediante la conservación, incremento y difusión de los fondos bibliográficos, documentales y audiovisuales de la Universidad.', '963877155', 'Edificio 4Q Camino de Vera, s/n 46022 Valencia');
```

```
INSERT INTO `bibliotecas`(`codb`, `nombre`, `horarios`, `imagen`, `descripcion`, `telefono`, `direccion`) VALUES (null, 'Biblioteca de Industriales', 'Lunes a Viernes: de 8:00 a 21:00', null, 'La Biblioteca de la Escuela Técnica Superior de Ingenieros Industriales, forma parte de la red de bibliotecas de la UPV y tiene como objetivo facilitar el acceso a la información científica y cultural, mediante la conservación, incremento y difusión de los fondos bibliográficos documentales y audiovisuales de la Universidad.', '963877170', 'Edificio 5F Camino de Vera, s/n 46022 Valencia');
```

```
INSERT INTO `bibliotecas`(`codb`, `nombre`, `horarios`, `imagen`, `descripcion`, `telefono`, `direccion`) VALUES (null, 'Biblioteca de Bellas Artes', 'Lunes a Viernes: de 8:00 a 21:00', null, 'La biblioteca de la Facultad de Bellas Artes tiene como objetivo facilitar el acceso a la información científica y cultural, mediante la conservación, incremento y difusión de sus fondos bibliográficos, documentales y audiovisuales.', '963877000', 'Edificio 3M-3N Camino de Vera, s/n 46022 Valencia');
```

```
INSERT INTO `bibliotecas`(`codb`, `nombre`, `horarios`, `imagen`, `descripcion`, `telefono`, `direccion`) VALUES (null, 'Biblioteca de Campus de Alcoy', 'Lunes a Viernes: de 8:00 a 21:00', 'images/bibliotecas/plazaCampusAlcoi.jpg', 'Biblioteca del campus de Alcoy donde se estudian diversas ingenierías como ingeniería informática o de diseño', '966528530', 'Plaza de Ferrándiz y Carbonell, s/n. Ed. Carbonell, 03801 Alcoy (Alicante - España).');
```

```
INSERT INTO `bibliotecas`(`codb`, `nombre`, `horarios`, `imagen`, `descripcion`, `telefono`, `direccion`) VALUES (null, 'Biblioteca de Campus de Gandía - CRAI', 'Lunes a Viernes: de 8:00 a 21:00', null, 'La Biblioteca Campus Gandía CRAI es el conjunto de espacios, colecciones y servicios que dan soporte al aprendizaje, la docencia, la investigación y la gestión del Campus de Gandía, Universitat Politècnica de Valencia.', '962849304', 'Edificio H – Escola Politècnica Superior de Gandía C/ Paranimf, 1 46730 GANDIA (Valencia)');
```

