



Universidad  
de La Laguna

Escuela Superior de  
Ingeniería y Tecnología  
Sección de Ingeniería Informática

# Trabajo de Fin de Grado

---

Creación de una Red Social Feedback para el  
intercambio de opiniones y experiencias

*Creation of a Social Network Feedback for opinion and  
experience sharing*

Mahrach Mahrach, Mohammed

---

La Laguna, 5 de septiembre de 2016

D. Jonas Philipp Lüke, con N.I.F. X0581666L profesor ayudante adscrito al Departamento de Ingeniería Industrial de la Universidad de La Laguna, como tutor

## **C E R T I F I C A**

Que la presente memoria titulada:

*“Creación de una Red Social Feedback para el intercambio de opiniones y experiencias.”*

ha sido realizada bajo su dirección por D. **Mohammed Mahrach Mahrach**, con N.I.F. 43844985-Q.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firma la presente en La Laguna a 5 de septiembre de 2016.

## Agradecimientos

En primer lugar, le agradezco a Dios por darme la vida y por haberme acompañado en ella en general y a lo largo de mi carrera en especial.

Agradezco a mis padres Abdellatif y Naima por su inestimable amor y apoyo incondicional.

A mi mujer Elena por quererme y apoyarme a lo largo de estos años con todo lo que conlleva de dificultad, lucha y sacrificios.

Y gracias a mi hija Maryam por hacernos los padres más felices del mundo.

A todos y cada uno de mis profesores que han sido durante este tiempo el apoyo para alcanzar este objetivo, les agradezco toda su dedicación, profesionalismo que ha contribuido de manera importante a mi crecimiento personal y profesional.

Especial mención para Jonás por su guía y consejo y por aceptar tutorizar este trabajo.

A mi amigo Andrés, todos estos años desde que empezamos la universidad e incluso antes, no habría sido lo mismo sin tu amistad.

Y por último a todos los compañeros y compañeras de la carrera de los que he aprendido mucho profesionalmente y personalmente.

Gracias a todos, de verdad.

## Resumen

Este documento describe las distintas fases de desarrollo del sistema FeedBack: Análisis de requisitos, Diseño, Implementación y Análisis de seguridad. Además, cubrirá el desarrollo de los tres componentes fundamentales del proyecto: el modelo de datos, el acceso a los datos e interfaz de usuario.

Haciendo uso de este sistema, se puede compartir experiencias y opiniones. La idea viene motivada por la necesidad de las personas de pedir consejo a la hora de comprar un producto, contratar un servicio, realizar un viaje o vivir una nueva experiencia personal, económica, turística, etc. Otras personas que tienen una experiencia previa en lo que estamos afrontando, pueden aportar su opinión, que sería muy útil para quien la necesite. Con la aplicación desarrollada un usuario dispone de una base de datos dinámica y que se puede consultar online de forma que cualquiera puede compartir sus experiencias online y recibir feedback a través de comentarios opinando sobre el mismo, también puede consultar experiencias de otros usuarios y retroalimentarlas en base a sus propias experiencias.

Este proyecto ofrece una herramienta mediante la cual se pueden crear post relacionado con alguna experiencia y publicarlos en el tablón de usuario, realizar búsquedas de experiencias publicadas según una necesidad, consultar el perfil del autor y seguirle si parece interesante y si desea recibir sus futuros post.

El documento comenzará introduciendo al lector en el proyecto que se va a realizar.

Después de marcar los objetivos de este trabajo se realizará un análisis de la posible competencia con el fin de asegurar en la medida de lo posible diferenciarnos de lo que existe actualmente. Seguidamente se explicarán las tecnologías utilizadas y los detalles tenidos en cuenta para resolver el problema planteado. Tras ello, se describe el análisis detallado de los requisitos software. En la fase de diseño, se explica la arquitectura utilizada y se describen sus componentes. Por último, se explica la implementación de dichos componentes según la arquitectura elegida junto con los resultados obtenidos, finalizando con las conclusiones obtenidas y posibles requisitos futuros.

**Palabras clave:** Aplicación web, análisis, diseño, implementación, Red Social, Xampp, PHP, MySQL.

## **Abstract**

This document describes the different stages of development of the the “Feedback” system: requirement analysis, design, implementation and security analysis. In addition, it will cover the development of the three main components of the project: the data model, data access and user interface.

The system allows to share experiences and opinions. The idea is motivated by the need of people to ask for advice when buying a product, a contract a service, when taking a trip or living a new personal, economic or touristic experience, etc. Other people who have prior experience in what we are facing, can contribute with their opinion, which would be very useful for those who need it. With the application developed, the user has a dynamic database that is available online so that anyone can share his experiences and receive feedback through comments. You can also check experiences of other users online and feedback them based on your own experience.

This project provides a tool by which you can create posts related with some experience that will be posted on the user wall, search for experiences published according to a need, check the author's profile and follow him if it looks interesting and you want to receive his future posts.

The document begins by introducing the reader to the project to be carried out.

After showing the goals of this work, an analysis of potential competitors will be done, in order to ensure as far as possible differentiate ourselves from what currently exists. Then, the technologies used and details considered for solving the problem raised will be explained. Thereafter, software requirements analysis is described in detail. In the design phase, the architecture used is explained and its components are described. Finally, the implementation of these components is explained according to the chosen architecture with the results obtained, ending with the conclusions and possible future requirements.

**Keywords:** Web Application, analysis, design, implementation, Social Network, Xampp, PHP, MySQL.

# Índice General

<b>Capítulo 1. Introducción</b>	<b>5</b>
1.1 Objetivos	6
1.2 Análisis de la competencia	7
1.3 Metodologías y Herramientas.	7
<b>Capítulo 2. Análisis de Requisitos</b>	<b>10</b>
2.1 Requisitos funcionales	11
2.2 Requisitos no funcionales.	12
2.3 Requisitos de información.	14
2.4 Modelo Conceptual OOP.	14
<b>Capítulo 3. Diseño del sistema</b>	<b>31</b>
3.1 Arquitectura de Software	31
3.2 Diagrama de clases	33
3.3 Diagrama de Componentes	36
3.4 Diseño de la base de datos	37
<b>Capítulo 4. Implementación</b>	<b>40</b>
4.1 Modelo	40
4.2 Vista	41
4.1 Controlador	42
<b>Capítulo 5. Resultados</b>	<b>46</b>
5.1 Registro y Acceso	46
5.2 Navegador de FeedBack	47
5.3 Crear post de Experiencia	48
5.4 Buscar una Experiencia	49
5.5 Comentar una Experiencia	50
<b>Capítulo 6. Resumen y conclusiones</b>	<b>53</b>
<b>Apéndice A. Competencia</b>	<b>55</b>
<b>Apéndice B. Protección de Datos Personales</b>	<b>56</b>
B.1. Protección de los derechos al honor, propia imagen, intimidad y privacidad	56
B.2. Protección de Datos de Carácter Personal	56
B.3. Protección de menores e incapaces	56
B.4. Aspectos relativos a la seguridad de la información	56
B.5. Ley de Servicios de la Sociedad de la información y el Comercio electrónico.	57
B.6. Ley General de Publicidad	57
<b>Apéndice C. Glosario de términos</b>	<b>58</b>
<b>Apéndice D. Presupuesto</b>	<b>59</b>
<b>BIBLIOGRAFÍA</b>	<b>60</b>

# Índice de figuras

Figura 1: Atom (IDE) .....	9
Figura 2: Historial de versiones de los componentes de XAMPP [5] .....	9
Figura 3: diagrama de flujo de proceso .....	10
Figura 4: Diagrama de paquetes .....	15
Figura 5: Diagrama del paquete Account.....	15
Figura 6: Diagrama del paquete Profile .....	20
Figura 7: Diagrama del paquete Experience .....	27
Figura 8: Arquitectura MVC .....	32
Figura 9: Diagrama de clases .....	35
Figura 10: Diagrama de Componentes.....	37
Figura 11: Modelo entidad-relación (Diagrama ER) .....	38
Figura 12: Modelo Relacional (Diagrama EER).....	39
Figura 13: Responsive Web Design .....	41
Figura 14: Maquetado de la plantilla en el diseño .....	42
Figura 15: Crear post de experiencia en MVC.....	42
Figura 16: Metodo addFeedback de la clase Experience .....	43
Figura 17: Barra superior de navegación .....	46
Figura 18: Formularios de registro y acceso al sistema .....	47
Figura 19: Navegador de Feedback.....	47
Figura 20: Creación de un post con adjuntos .....	48
Figura 21: Post con adjunto publicado .....	48
Figura 22: Buscar Experiencias relacionados con la Gomera.....	49
Figura 23: Comprobación del número de los registros en la base de datos .....	49
Figura 24: Resultado de búsqueda de post en Feedback.....	49
Figura 25: Tipos de feedback (Agree/Disagree) .....	50
Figura 26: Los comentarios y sub-comentarios de un post.....	50
Figura 27: Acciones de edición de un post .....	51
Figura 28: Página personal (perfil) del usuario .....	51
Figura 29: Página personal (perfil) de otro usuario .....	52
Figura 30: Mis favoritos .....	52

# Índice de tablas

Tabla 1: Gestión de acceso.....	11
Tabla 2: Gestión de usuarios .....	11
Tabla 3: Gestión de perfiles.....	12
Tabla 4: Gestión de las publicaciones .....	12
Tabla 5: Gestión de los muros de usuarios.....	12
Tabla 6: Disponibilidad y estabilidad del sistema.....	13
Tabla 7: Interfaz y diseño .....	13
Tabla 8: Usabilidad y accesibilidad web.....	13
Tabla 9: Concurrencia optimizada .....	14
Tabla 10: Gestión de información de usuario .....	14
Tabla 11: Gestión de información del sistema.....	14
Tabla 12: Caso de uso “sign up” .....	17
Tabla 13: Caso de uso “Sign out” .....	18
Tabla 14: Caso de uso “Login” .....	19
Tabla 15: Caso de uso “Logout” .....	19
Tabla 16: Caso de uso “View profile” .....	22
Tabla 17: Caso de uso “View Personal data”.....	23
Tabla 18: Caso de uso “Edit personal data”.....	24
Tabla 19: Caso de uso “Visit other profiles” .....	25
Tabla 20: Caso de uso “Set favorite” .....	25
Tabla 21: Caso de uso “Set unfavorite” .....	26
Tabla 22: Caso de uso “Show experience” .....	28
Tabla 23: Caso de uso “Create experience” .....	29
Tabla 24: Caso de uso “Search experience”.....	30
Tabla 25: Casos de uso “Agree/disagree experience”.....	31
Tabla 26: Control de acceso por sesión.....	45
Tabla 27: Competencia.....	55
Tabla 28: Presupuesto .....	59



# Capítulo 1.

## Introducción

Las redes sociales [1] son estructuras que permiten la comunicación entre grupos de personas. Este concepto ha existido desde el principio de los tiempos. Sin embargo, la digitalización de este fenómeno no es muy antigua, En poco tiempo, se han convertido en un medio de comunicación de gran envergadura. Las redes sociales se pueden clasificar basándose en la temática, teniendo en cuenta si el objetivo es congregarse en torno a una temática definida a un colectivo concreto, o basándose en el tipo de relación social que generan, si centran su atención en fomentar las relaciones humanas utilizando enlaces o tipos de relaciones como amistad, parentesco, intereses comunes o perfil social.

Algunas de estas plataformas impactan directamente en la forma de operar de la economía, los usuarios han descubierto el beneficio de pertenecer a una comunidad en la que pueden ayudar y pedir ayuda sin tener que conocerse y establecer relaciones basadas en la confianza. Este nuevo modelo se denomina economía colaborativa. Se pueden identificar una serie de categorías que permiten clasificar las plataformas de redes sociales digitales más significativas. La primera categoría se centra en los bienes, englobando tanto el intercambio y/o compraventa de objetos usados, como la venta de productos personalizados. Algunos ejemplos son, Etsy, Quirky o FabLabs. La segunda categoría son las redes sociales centradas en servicios, que permiten compartir trabajo o tiempo, como, por ejemplo, Elance o TaskRabbit. Otra categoría sería la de los transportes, como es el caso de las plataformas de alquiler de vehículos particulares, como, por ejemplo, RelayRides o servicios de transporte, como, por ejemplo, Blablacar, que facilita compartir coches particulares. También existen plataformas centradas en compartir espacios. Por ejemplo, Zipcar se centra en compartir espacio de oficina, mientras que AirBnB, se centra en el alquiler de habitaciones en casas de particulares. La siguiente categoría es la basada en el dinero, incluyendo sistemas de crowdfunding o los famosos préstamos P2P a través de plataformas como Comunitae, Kickstarter o KissKissBankBank. Finalmente, se puede distinguir la categoría de redes sociales basadas en compartir experiencias y conocimiento, donde el intercambio de datos, información y conocimiento útil son el objeto principal, pudiendo generar beneficios económicos en forma de ahorro de tiempo, dinero o evitando el gasto de dinero en servicios inútiles.

La idea que se desarrolla en este Trabajo de Fin de Grado, se encuadra en la última categoría, basándose en el intercambio de opiniones y experiencias, como, por ejemplo, zoover o TripAdvisor que se centra en recomendaciones sobre lugares y establecimientos turísticos. Concretamente, lo que se pretende hacer es una plataforma para una red social que permita compartir experiencias de todo tipo y luego categorizarlas y poder alimentarlas por comentarios, críticas y recomendaciones de otros usuarios. Esto puede ser muy beneficioso para otros usuarios que podrían estar afrontando situaciones parecidas o que necesitan consultar otras opiniones antes de comprar productos, contratar servicios, visitar lugares, etc. La red social que se pretende formar alrededor de esta aplicación la denominaremos Feedback.

Este documento contiene la descripción del trabajo realizado durante el Trabajo de Fin de Grado y se organiza como sigue: Primero se encuentra este capítulo de introducción que contiene los objetivos, el análisis de la competencia, así como las metodologías y herramientas empleadas en el desarrollo. Seguidamente, en el capítulo 2 se realiza el análisis de requisitos, mientras que en el capítulo 3 se describen los aspectos más relevantes del sistema desarrollado. El capítulo 4, describe la implementación realizada. Finalmente, en el capítulo 5 se explican los resultados obtenidos mediante este desarrollo, haciendo hincapié en las principales funcionalidades que se han desarrollado y el capítulo 6 contiene las conclusiones y las líneas de trabajo futuras.

## **1.1 Objetivos**

### **1.1.1 Objetivo General**

El objetivo de FeedBack no es sólo ayudar al usuario a localizar experiencias y consejos sobre algún tema que les interese. También pretende convertirse en una herramienta de publicidad transparente por la que se puede dar a conocer cualquier producto y dejar que los mismos usuarios opinen y aconsejen sus productos a través de la red. Esto constituye una gran ventaja para las empresas que están seguras de la calidad de sus productos obteniendo más credibilidad con los apoyos de los usuarios, pero también para poder mejorar la calidad de sus productos o servicios a través de las críticas recibidas por parte de los usuarios, que pueden no estar satisfechos de algún aspecto de lo que ofrecen.

Lo que se pretende es crear un encuentro social virtual que cambia el sentido de redes sociales de comunicación y ocio, siendo una herramienta beneficiosa para la comunidad compartiendo experiencias y opiniones.

### **1.1.2 Objetivos Específicos**

Se proponen una serie de objetivos que se deberían alcanzar al finalizar del proyecto y que se encuentran detallados a continuación:

- Compartir experiencias de todo tipo, poder buscarlas y opinar sobre las mismas, para que incluso la peor experiencia para alguien pueda ser algo útil para otros.
- Opinar sobre las experiencias de los demás, retroalimentándolas por comentarios o adjuntando nuestras propias experiencias relacionadas, valorándolas positivamente o negativamente.
- Crear una herramienta de publicidad transparente basada en las opiniones reales de los usuarios, evitando así la publicidad ingeniosa y tener un feedback más fiable.
- Comparar en base a las opiniones reales de los diferentes usuarios de la comunidad, la calidad de un servicio, producto o una experiencia para sacar nuestras propias conclusiones que pueden ser negativas o positivas según los criterios que busca cada uno.

- Crear un perfil de usuario donde aparece un historial de sus experiencias en forma de una página personal pública, donde otros usuarios puede consultar, seguir y opinar.

## 1.2 Análisis de la competencia

La realización del análisis de la competencia es un paso previo indispensable que influye de forma directa en las decisiones de la estrategia de negocio de este proyecto, sobre todo si se trata, como es nuestro caso, de una red social que aspira a convertirse en una herramienta realmente útil e indispensable en la vida diaria de los usuarios en un mercado casi agotado de redes sociales.

Para saber quién es realmente competencia se he llevado a cabo la investigación de identificación de la competencia. Esta investigación ha dado como resultado la ausencia de una aplicación que cumpla todos los requisitos finales sobre todo convertir la temática general de la plataforma, un objeto abstracto, es decir, lo especial en la red social FeedBack es que no se especializa en ningún sector, tema o tipo de objetos.

Sin embargo, existe una cierta similitud en algunos aspectos en cada competidor de forma que, analizados sus aspectos por separado podrían constituir una competencia parcial.

Esta investigación de identificación de la competencia se resume en la Tabla 27, que se encuentra en el apéndice A, destacando los aspectos de similitud y diferencias de cada entidad y su aspecto competitivo con el proyecto FeedBack.

## 1.3 Metodologías y Herramientas.

En este apartado se detallan las herramientas empleadas para alcanzar los objetivos planteados en este Trabajo de Fin de Grado. La elección de las herramientas adecuadas es fundamental para el correcto desarrollo del proyecto, por ello se ha realizado un estudio de distintas opciones, adoptando finalmente las que se detallan a continuación.

### 1.3.1 Herramientas de desarrollo

Como la mayoría de las aplicaciones que soportan redes sociales, Feedback se basará en tecnologías web y se distinguirá entre la parte de front-end y la parte de back-end.

#### Front-end<sup>1</sup>

Debido a que FeedBack debe tener una interfaz web, las tecnologías Front-end serán HTML5, CSS3 y JavaScript, además de algunas framework como bootstrap para css y JQuery para javascript.

1. **HTML5** (HyperText Markup Language 5): Es la nueva versión del lenguaje HTML.
2. **CSS3** (Cascading Style Sheets Level 3): Es un lenguaje para definir el estilo o la apariencia de las páginas web.

---

<sup>1</sup> Tecnologías que corren del lado del cliente como (HTML5, CSS3, Javascript, jQuery)

3. **JavaScript:** Implementado como parte de un navegador web permite crear diferentes efectos e interactuar los usuarios y mejorar la interfaz web.
4. **Librerías y framework:** También se utiliza bootstrap<sup>2</sup> como framework de CSS y librerías de javascript como JQuery<sup>3</sup>.

## Back-end<sup>4</sup>

Existen multitud de lenguajes para el Back-End, algunos con más éxito que otros, entre ellos PHP, Ruby con Rails, Python con Django y muchos otros, aunque la elección final es el resultado de la comparación entre un lenguaje tradicional como PHP y otro más reciente que utilizan Javascript llamado Node.JS.

Tanto PHP como Node.JS funcionan del lado servidor y son opciones viables para el desarrollo de sitios y aplicaciones web.

- **Node.js** [2]: Es un framework open source que se apoya en el motor V8<sup>5</sup> para permitir la ejecución de programas hechos en Javascript del lado del servidor. Por lo tanto, permite que el servidor y las aplicaciones de escritorio se comuniquen por medio de Javascript de una forma extremadamente rápida. A veces se hace referencia a NodeJS como “es Javascript del lado del servidor”, pero es mucho más.
- **PHP** [3] (acrónimo recursivo de PHP: Hypertext Preprocessor): Es un lenguaje de programación de uso general, que funciona del lado de servidor embebidos dentro de HTML.

Aunque se dice que **Node.js** es un lenguaje con más futuro, finalmente, la elección ha sido PHP. Esta elección se basa principalmente en que el lenguaje es de programación de uso general, además de los siguientes motivos:

1. Lenguaje totalmente libre, abierto y muy bien documentado.
2. Curva de aprendizaje muy baja al contrario de Node.JS.
3. Los entornos de desarrollo son de rápida y fácil configuración
4. Fácil despliegue: paquetes totalmente autoinstalables que integran PHP
5. Fácil acceso a bases de datos, totalmente compatible con MySQL.
6. Comunidad muy grande y de rápida solución de problemas, mientras para Node.JS es muy escasa.

### 1.3.2 Entorno de desarrollo

Como entorno de desarrollo se ha optado por Atom ya que es un editor de código libre y de código abierto para OS X, Linux y Windows. Además, tiene soporte para múltiples plug-ins. En la Figura 1, se puede observar una captura de pantalla de este entorno de desarrollo

---

<sup>2</sup> Web oficial en <http://getbootstrap.com>

<sup>3</sup> Web oficial en <https://jquery.com>

<sup>4</sup> Tecnologías que corren del lado del servidor como (PHP, Ruby on Rails, Django, Node.js, .NET)

<sup>5</sup> Un motor de código abierto para JavaScript creado por Google, siendo su programador jefe Lars Bak. Está escrito en C++ y es usado en Google Chrome y en el navegador de internet del sistema operativo Android 2.2. [13]

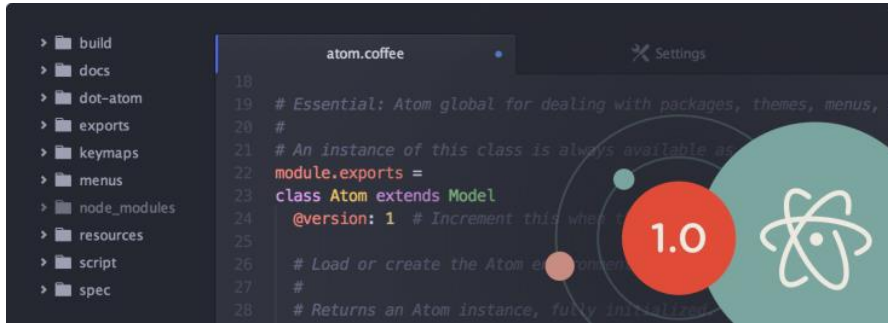


Figura 1: Atom (IDE)

### 1.3.3 Servidor web

Para montar una aplicación web, es necesario disponer tanto de un servidor de páginas web, como un lenguaje de programación y una base de datos.

**XAMPP** [4]: Es un acrónimo que denomina las primeras letras de un conjunto de componentes que forman parte de una solución. Tres proyectos de software libre han proporcionado esas herramientas con una calidad sobresaliente.

Generalmente, es utilizado para construir sitios web dinámicos sometidos a gran cantidad de peticiones. Sus componentes son:

- **X**: para cualquiera de los diferentes sistemas operativos
- **A**: de Apache ha sido desde prácticamente el principio de la web el servidor de páginas web más utilizado (como lo atestigua mes a mes la empresa Netcraft en su Web Server Survey).
- **M**: de MySQL, es un sistema gestor de bases de datos libre creado en 1995. MySQL se convirtió rápidamente en la base de datos libre más popular, desde la versión "5.6.15", XAMPP cambió la base de datos de MySQL A MariaDB. El cual es un fork de MySQL con licencia GPL.
- **P** (de PHP): Es el lenguaje de referencia para el desarrollo web.
- **P** (de Perl): Es un lenguaje de programación muy utilizado para construir aplicaciones CGI para el web.

Todos estos componentes destacan también por ser de código abierto aumentando su atractivo al no depender de ningún vendedor concreto.

La imagen que aparece en la Figura 2 muestra la historia de las últimas versiones de los componentes de XAMPP.

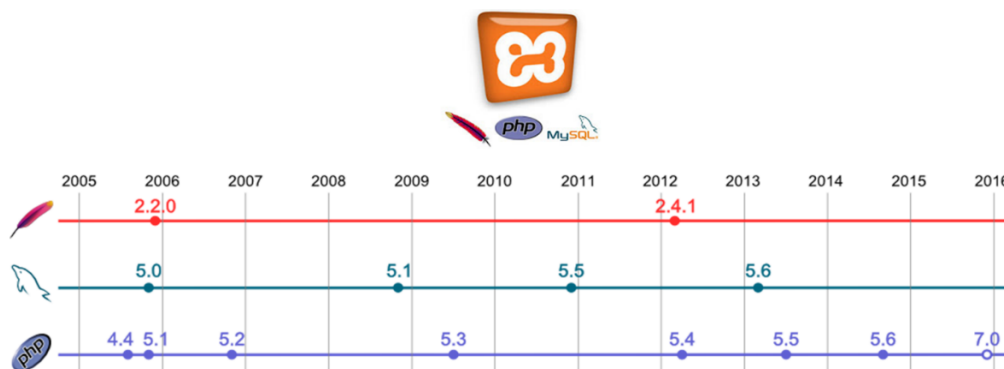


Figura 2: Historial de versiones de los componentes de XAMPP [5]

# Capítulo 2.

## Análisis de Requisitos

Para especificar las características operacionales, las interfaces entre los elementos del sistema y las restricciones que debe cumplir el software, es necesario realizar un análisis de requisitos<sup>6</sup>. La primera tarea que nos permitirá entender bien los mecanismos del funcionamiento básico requeridos del sistema y el flujo de datos entre el sistema y el usuario a través de su interfaz web, es realizar un diagrama que describe el de flujo de datos y procesos se ha realizado un diagrama de flujo. Este diagrama se muestra en la Figura 3 y que se describirá a continuación.

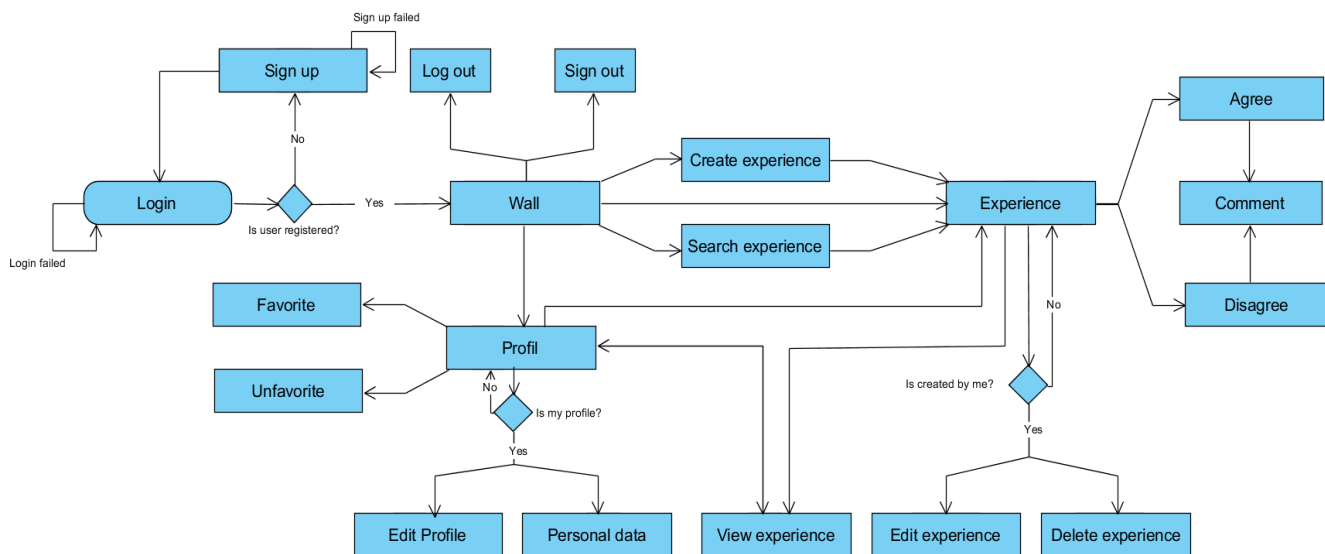


Figura 3: diagrama de flujo de proceso

El usuario debe poder ver su perfil y en cualquier momento debe poder ampliar, modificar o personalizar la configuración de su perfil. Además, un usuario debe poder visualizar los perfiles de otros usuarios, siempre con las configuraciones de visibilidad permitidas para proteger la privacidad de datos de estos usuarios.

También deben poderse crear nuevos post y que aparezcan en el perfil según su orden cronológico. Estos post pueden ser una simple foto y una breve descripción o un video que describe una experiencia, un enlace externo o un artículo escrito, siempre pudiendo añadir ficheros multimedia al post que describiría una experiencia personal, turística, económica o un lugar donde se ha comprado o contratado un servicio. Obviamente, el usuario debe poder ver sus propios post, poder modificarlos o eliminarlos.

Buscar y visitar las experiencias publicadas por otros usuarios es otra de las funcionalidades que permitirá aprovechar experiencias de otro usuario. Una vez localizado el post de interés puede también recomendar o no recomendar (Agree/Disagree) la experiencia descrita. En tal caso, habrá que argumentar su opinión con un comentario o describir su propia experiencia

<sup>6</sup> Es el conjunto de técnicas y procedimientos que nos permiten conocer los elementos necesarios para definir un proyecto de software. [14]

relacionada con el mismo tema. De esta forma, el beneficio es compartido, dejando que los mismos usuarios de la plataforma valoren la experiencia y las opiniones acerca de ella.

También se debería poder seguir a otros usuarios para recibir sus experiencias si encajan con su perfil y que sus experiencias le pueden aportar beneficios y dejar de seguirlos en caso contrario.

Por otra parte, se ha calculado que el presupuesto inicial para llevar a cabo este trabajo y que se estima en 6670 euros, este presupuesto se muestra detallado en la Tabla 28 que se encuentra en el apéndice D.

## 2.1 Requisitos funcionales

Los Requisitos funcionales describen el funcionamiento del sistema y los servicios que hay que proporcionar. En este apartado, se definirán este tipo de requisitos indicando cómo debe reaccionar el sistema ante una entrada particular y cómo se debe comportar ante diferentes situaciones.

Las siguientes tablas definen los requisitos funcionales básicos que tiene que cumplir la plataforma Feedback.

### 2.1.1 Gestión de acceso

RF-01	Gestionar el acceso al sistema
Descripción	El sistema deberá gestionar el acceso al sistema tomando en cuenta la seguridad de los usuarios y del mismo sistema.
Estabilidad	Alta
Comentarios	La gestión del acceso al sistema se controla mediante cuentas de usuario y control de sesiones.

Tabla 1: Gestión de acceso

### 2.1.2 Gestión de usuarios

RF-02	Gestionar los usuarios
Descripción	El sistema deberá gestionar las cuentas de usuarios, registros, carpetas personales, post propios, visitar post de otros usuarios y post en los que participa, valorarlas y comentarlas.
Estabilidad	Alta
Comentarios	Un usuario puede tener su cuenta de usuario una vez registrado en el sistema.

Tabla 2: Gestión de usuarios

### 2.1.3 Gestión de perfiles

RF-03	Gestionar de perfiles
Descripción	El sistema deberá gestionar los perfiles de los usuarios, poder consular la información personal y editarla. Además, un usuario puede visualizar un perfil de otro usuario con la restricción de no poder modificar datos.
Estabilidad	Alta
Comentarios	El perfil es la página personal del usuario, ahí un usuario puede ver sus propias experiencias, su información personal la que puede editar o añadir o ampliar.

Tabla 3: Gestión de perfiles

### 2.1.4 Gestión de las publicaciones

RF-04	Gestionar los Post
Descripción	El sistema deberá gestionar publicación de los post, poder editarlos y permitir su eliminación por parte de sus autores, visitar post de otros usuarios, valorarlos y comentarlos.
Estabilidad	Alta
Comentarios	Un post describe una experiencia de un usuario, se publica y es visible para todos los usuarios.

Tabla 4: Gestión de las publicaciones

### 2.1.5 Gestión de los muros de usuarios

RF-05	Gestionar los Muros
Descripción	El sistema deberá gestionar los muros de los usuarios, gestión y orden de visualización de los post en el muro, experiencias en los que participan, valora o comenta.
Estabilidad	Alta
Comentarios	El muro es la página principal de la página, ahí un usuario puede editar una experiencia o buscar también.

Tabla 5: Gestión de los muros de usuarios

## 2.2 Requisitos no funcionales.

Los requisitos no funcionales o atributos de calidad como indica su nombre son, requisitos que especifican criterios para juzgar la calidad de una operación de un sistema. Por tanto, se refieren a todos los requisitos que no describen información, ni funciones, sino características de funcionamiento como la disponibilidad, la estabilidad, usabilidad y accesibilidad.

Las siguientes tablas definen los requisitos no funcionales que tiene que cumplir la plataforma Feedback.



## Disponibilidad y estabilidad del sistema

RNF-01	Disponibilidad y estabilidad
Descripción	El sistema debe contar con una interfaz de administración que incluya: Administración de perfil de usuarios, Administración de datos. El sistema debe tener capacidad de permitir su fácil mantenimiento con respecto a los posibles errores que se puedan presentar durante la operación del sistema.
Estabilidad	Alta
Comentarios	Todo el sistema deberá estar complemente documentado, cada uno de los componentes de software que forman parte de la solución propuesta deberán estar debidamente documentados tanto en el código fuente como en los manuales de administración y de usuario.

Tabla 6: Disponibilidad y estabilidad del sistema

### 2.2.1 Interfaz y diseño

RNF-02	Diseño amigable y responsivo
Descripción	El sistema deberá de estar desarrollado con un diseño web responsive o bien web adaptive para que la web sea adecuada a cualquier dispositivo cambia su aspecto automáticamente para adaptarse al dispositivo que el usuario está utilizando.
Estabilidad	Media
Comentarios	La diferencia entre Responsive y Adaptive es que en el diseño responsive, permite que las páginas de la web se ajusten automáticamente al navegador con el mismo contenido en cada dispositivo, mientras el diseño adaptive, a través del navegador, es el servidor quien detecta el dispositivo del usuario y carga el contenido al que está optimizado con contenido personalizado para cada dispositivo [6].

Tabla 7: Interfaz y diseño

## Usabilidad y accesibilidad web

RNF-03	Usabilidad y accesibilidad
Descripción	El sistema deberá hacer que los objetos, acciones y opciones sean visibles. El usuario no tiene por qué recordar información. El usuario siempre debe saber exactamente qué es lo que el sistema está haciendo. El sistema debe hablar el lenguaje del usuario, Las frases, palabras y conceptos deben de ser familiares para el usuario, haciendo que la información aparezca en un orden lógico y natural [7].
Estabilidad	Media
Comentarios	Ninguno

Tabla 8: Usabilidad y accesibilidad web

## 2.2.2 Gestión de concurrencia optimizada

RNF-04	Acceso concurrente a bases de datos
Descripción	El sistema deberá asegurar que múltiples accesos se puedan realizar concurrentemente sin causar lentitud notable en el sistema. Los procesos deben progresar constantemente y sin interrupción, así como las acciones de intercambio de información.
Estabilidad	Media
Comentarios	Ninguno

Tabla 9: Concurrencia optimizada

## 2.3 Requisitos de información.

Los requisitos de información describen qué información debe almacenar el sistema para poder ofrecer los servicios necesarios. Deben identificar el concepto relevante sobre el que se debe guardar información, así como qué datos específicos relativos al concepto son importantes para cumplir los objetivos del sistema.

### 2.3.1 Gestión de información de usuario

RI-01	información de usuario
Descripción	El sistema deberá almacenar, gestionar y proteger la información de los usuarios y sus cuentas, así como datos personales requeridos para el sistema.
Estabilidad	Alta
Datos	Datos personales, contraseñas, correos, publicaciones.

Tabla 10: Gestión de información de usuario

### Gestión de información del sistema

RI-02	información del sistema
Descripción	El sistema deberá almacenar y proteger la información del sistema, así como la delegación a los administradores de la gestión y actualización de los mismos
Estabilidad	Media
Comentarios	Datos y procedimientos del sistema

Tabla 11: Gestión de información del sistema

## 2.4 Modelo Conceptual OOP.

A continuación, se detallan las distintas agrupaciones lógicas en las que se divide el sistema y las dependencias entre las mismas. Esto se muestra en el diagrama de paquetes de la Figura 4 y seguidamente se detallan los casos de uso contenidos en cada uno de los tres paquetes de “application logic”. Los demás paquetes son paquetes auxiliares.

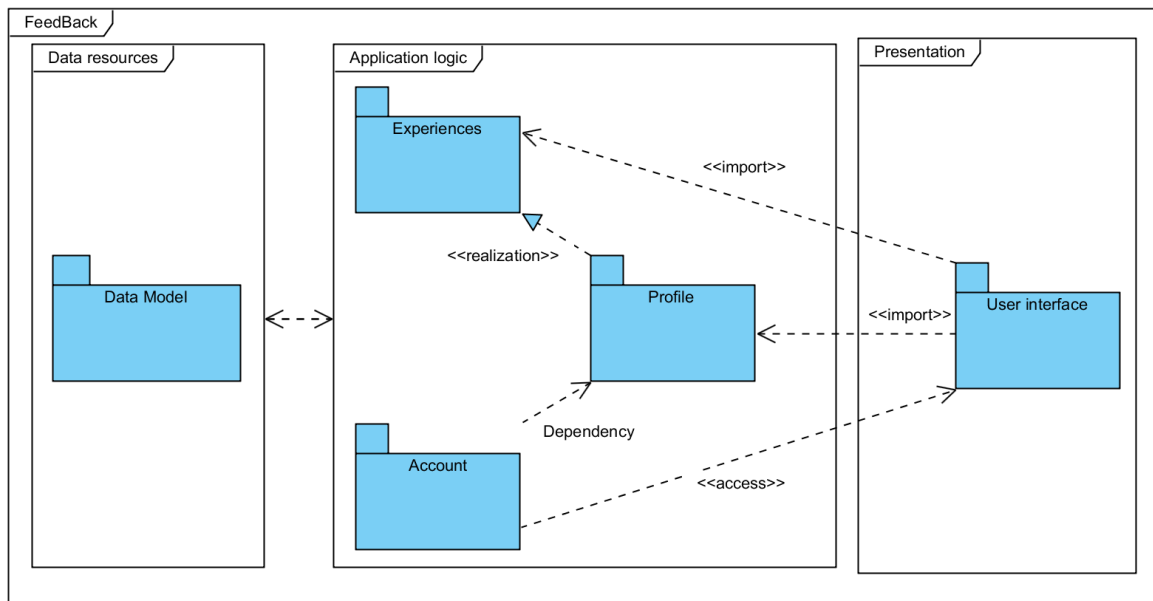


Figura 4: Diagrama de paquetes

### 2.4.1 Casos de Uso del paquete Access.

El paquete inicio contiene todas aquellas funcionalidades relacionadas con el registro de usuarios y el acceso de los mismos a la aplicación. El diagrama de casos de uso se muestra en la Figura 5, distinguiéndose cuatro casos de uso principales: Login, Logout, Sign up y Sign out, el diagrama incluye otros casos de uso realizados automáticamente por el sistema. A continuación, se detallan los casos de uso del paquete Account. Una descripción más detallada de los casos de uso del Paquete Account, se detalla en la Tabla 12 a Tabla 15.

#### Diagrama de casos de uso Account

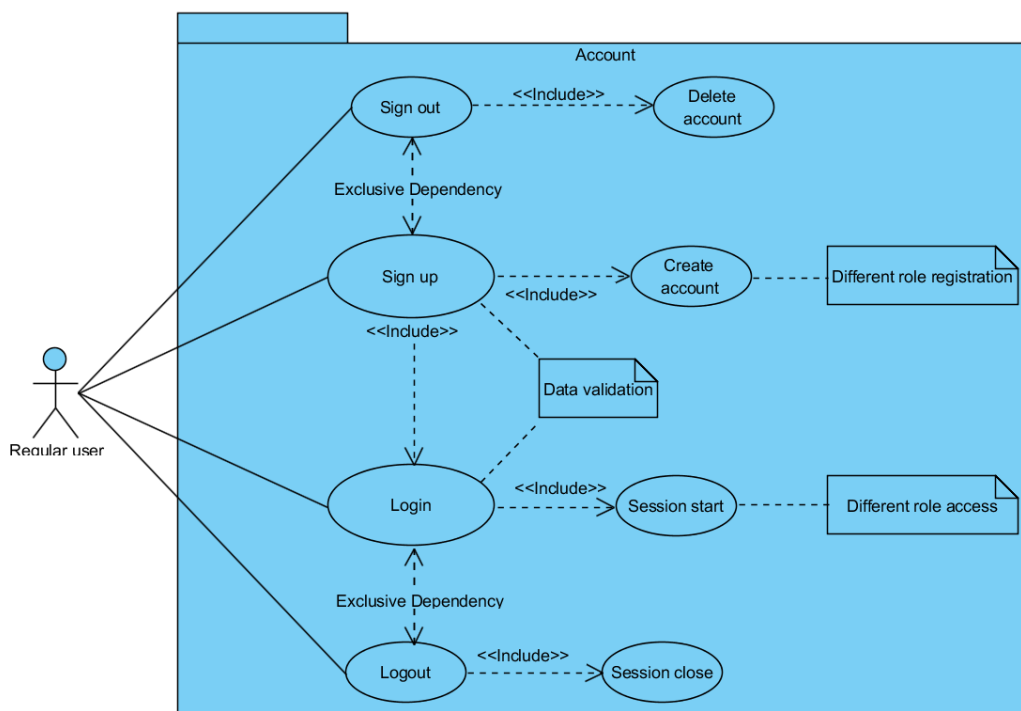


Figura 5: Diagrama del paquete Account

### *Especificación del caso “Sign up”*

Para poder tener acceso a todos los servicios que ofrece la plataforma Feedback es necesario registrarse, para ellos el usuario debe seleccionar la opción “Sign up” en la barra superior, rellenar todos los campos del formulario obligatorios y presiona el botón enviar, el sistema así valida los datos ingresados del sistema y crea una cuenta para el usuario queda registrado y almacenado en la base de datos, en caso de que algún dato sea erróneo o no válido el sistema avisa al usuario mediante una notificación del error y le permitirá volver a intentarlo de nuevo.

### *Especificación del caso “Sign out”*

La plataforma Feedback permite a los usuarios darse de baja del sistema, para ello el usuario debe seleccionar la opción “Sign out” en la página personal del usuario, introducir su contraseña, clic en el botón enviar y confirmar el aviso final de que la cuenta de usuario quedará eliminada del sistema, el sistema valida la contraseña y la sesión para proceder a la eliminación de la cuenta de usuario, en caso de que algún dato sea erróneo o no válido el sistema avisa al usuario mediante una notificación del error y le permitirá volver a intentarlo de nuevo.

### *Especificación del caso “Login”*

Para iniciar una sesión en el sistema, el usuario debe validarse con sus credenciales, para esto es necesario estar registrado previamente en el sistema, el usuario debe seleccionar la opción “Login” en la barra superior, introducir su usuario y contraseña, luego presiona el botón “Login”, el sistema valida los datos ingresados, permitirá al usuario el acceso al sistema y abrirá una nueva sesión, en caso de que algún dato sea erróneo o no válido el sistema avisa al usuario mediante una notificación del error y le permitirá volver a intentarlo de nuevo.

### *Especificación del caso “Logout”*

El usuario puede terminar la sesión cuando termina su navegación y salir del sistema. Para ello el usuario debe seleccionar opciones de usuario en la barra superior y elegir “Logout”, el sistema cierra la sesión en el servidor y redirige el usuario a la página inicial de la web.

Ficha de caso de uso "Sign up"			
CU-1	Sign up		
Fecha	10/10/2015		
Objetivos asociados	Registrarse en el sistema		
Descripción	El usuario se registra en el sistema para poder tener acceso a la aplicación.		
Precondición	El usuario debe aceptar la normas y términos de contrato además tener la edad mínima para poder inscribirse en la aplicación		
Secuencia normal	Paso	Acción	Descripción
	1	Ejecutar Aplicación.	El caso de uso comienza cuando el usuario inicia la aplicación web en el su ordenador, tableta o móvil.
	2	Seleccionar opción Registrar.	El usuario selecciona la opción "SignUp" en la barra superior.
	3	Formulario de registro del usuario.	El sistema muestra el formulario de registro de nuevos usuarios para que el usuario introduzca los datos solicitados y presione el botón enviar.
	4	Procesamiento de datos.	El sistema valida los datos ingresados y crea una cuenta con los datos proporcionados por el usuario.
	5	Fin de caso de uso.	El usuario queda registrado y almacenado en la base de datos.
Post-condición	El sistema crea una cuenta nueva con los datos proporcionados por el usuario.		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/flujo Alternativos
	Paso 4 (caso 1)	Excepción en la validación de datos del formulario: El nombre de usuario o el correo ya se han utilizado previamente.	El sistema notifica que el nombre de usuario o el correo ya se han utilizado previamente y pedirá al usuario que vuelva a intentarlo. Volver al paso 3 del flujo base.
	Paso 4 (caso 2)	Excepción en la validación de datos del formulario: Los datos son incorrectos o incompletos.	El sistema notifica que los datos son incorrectos o incompletos. Volver al paso 3 del flujo base.

Tabla 12: Caso de uso "sign up"

Ficha de caso de uso "Sign out"			
CU-2	Sign out		
Fecha	10/10/2015		
Objetivos asociados	Darse de baja en el sistema.		
Descripción	El usuario cancela su registro en el sistema y eliminar su cuenta.		
Precondición	Es necesario estar ingresado en el sistema con sesión abierta.		
Secuencia normal	Paso	Acción	Descripción
	1	Página personal del usuario.	El caso de uso comienza cuando el usuario tenga su sesión abierta en el sistema y tenga en su página personal abierta.
	2	Seleccionar la pestaña "Information".	El usuario selecciona la pestaña "Information" en la página personal.
	3	Seleccionar opción "Sign out".	El usuario selecciona la opción "Sign out" en la parte inferior de la página de información privada.
	4	Formulario de baja del usuario.	El sistema muestra el formulario de baja, donde el usuario proporcionará su contraseña y presiona el botón enviar.
	5	Aviso y confirmación.	El usuario confirma el aviso final de que su cuenta quedará eliminada del sistema junto a todos sus datos relacionados.
	6	Procesamiento de datos.	el sistema valida los datos ingresados y los datos de sesión para proceder a la eliminación de la cuenta de usuario.
	7	Fin de caso de uso.	Se cierra la sesión y el usuario quedará eliminado del sistema junto a todos sus datos relacionados.
Post-condición	La cuenta se eliminará de forma definitiva y los datos del usuario serán inaccesibles.		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/flujo Alternativos
	paso 4	Excepción en la validación de datos del formulario: La contraseña es incorrecta.	El sistema notifica que la contraseña es incorrecta. Volver al paso 3 del flujo base.
	Paso 5	Cancelación de la operación: Usuario no acepta el aviso	El sistema no procede a la eliminación de la cuenta. Volver al paso 2 del flujo base.

Tabla 13: Caso de uso "Sign out"

Ficha de caso de uso "Login"			
CU-3	Login		
Fecha	10/09/2015		
Objetivos asociados	Iniciar sesión		
Descripción	El usuario inicia sesión en el sistema para navegar en el sistema.		
Precondición	El usuario deberá estar registrado en el sistema.		
Secuencia normal	Paso	Acción	Descripción
	1	Ejecutar Aplicación	El caso de uso comienza cuando el usuario inicia la aplicación web en el su ordenador, tableta o móvil.
	2	Llenar formulario	El usuario introduce su usuario y contraseña y presiona el botón enviar.
	3	Validación y acceso.	El sistema valida correctamente los datos y muestra la pantalla principal.
	4	Fin de caso de uso	Una vez que se abra la sesión y el usuario puede navegar en el sistema termina el caso de uso.
Post-condición	Se mantiene en el servidor una sesión única con los datos del usuario, hasta que el usuario decida cerrar sesión.		
Excepciones de la aplicación o flujos alternativos	Paso	Descripción	Manejo de la excepción/flujo Alternativos
	3	Excepción en la validación de datos del formulario: Los datos son incorrectos o incompletos.	El sistema notifica que los datos de acceso son incorrectos o incompletos. Seguir con el paso 4 del flujo base.
	3	Excepción en la sesión de usuario: Otra sesión abierta con las mismas credenciales.	El sistema notifica que existe otra sesión abierta con el mismo usuario, piderá al usuario de cerrar sesión y volver a intentarlo. Volver al paso 1.

Tabla 14: Caso de uso "Login"

Ficha de caso de uso "Logout"			
CU-4	Logout		
Fecha	15/09/2015		
Objetivos asociados	Cerrar sesión		
Descripción	El usuario cierra la sesión para salir del sistema.		
Precondición	El usuario deberá estar dentro del sistema con la sesión abierta.		
Secuencia normal	Paso	Acción	Descripción
	1	Seleccionar opciones de usuario	El caso de uso comienza cuando el usuario decide cerrar la sesión y seleccionar opciones de usuario en la barra superior.
	2	Seleccionar "Logout"	El usuario elegir la opción "Logout".
	3	Validación y cierre	El sistema valida los datos de sesión antes cerrar la sesión en el servidor.
	4	Fin de caso de uso	Finalmente, el sistema redirige el usuario a la página inicial de la web y termina el caso de uso.
Post-condición	Se eliminan los datos de la sesión en el servidor.		

Tabla 15: Caso de uso "Logout"

## 2.4.2 Casos de Uso del Paquete Profile

En la Figura 6 se muestra el diagrama de Casos de uso del Paquete Profile, donde se muestra la interacción del usuario con los diferentes casos de uso en este paquete. El paquete se encarga de los casos de uso relacionados con la página personal del usuario donde un usuario puede consultar editar o eliminar tanto sus experiencias a lo largo del tiempo, como sus datos personales privados.

A continuación, se detallan los casos de uso del paquete Profile. Una descripción más detallada de los casos de uso del Paquete Access, se detalla en Tabla 16 a Tabla 21.

### Especificación del caso de uso “View profile”

El usuario pueda acceder a su página personal, para esto es necesario estar ingresado en el sistema, el usuario debe seleccionar la opción “Profile” en el menú lateral, el sistema comprueba los datos de sesión actuales y selecciona los permisos de acceso adecuados al perfil, permitirá al usuario el acceso a la página de perfil, donde por defecto se mostrarán sus experiencias personales, podrá consultar sus datos personales “View personal data” y realizar cambios en dichos datos “Edit personal data”. En caso de que la sesión haya caducado no se mostrarán los datos personales y solo podrá ver las experiencias al usuario visitado.

### Diagrama de casos de uso Profile

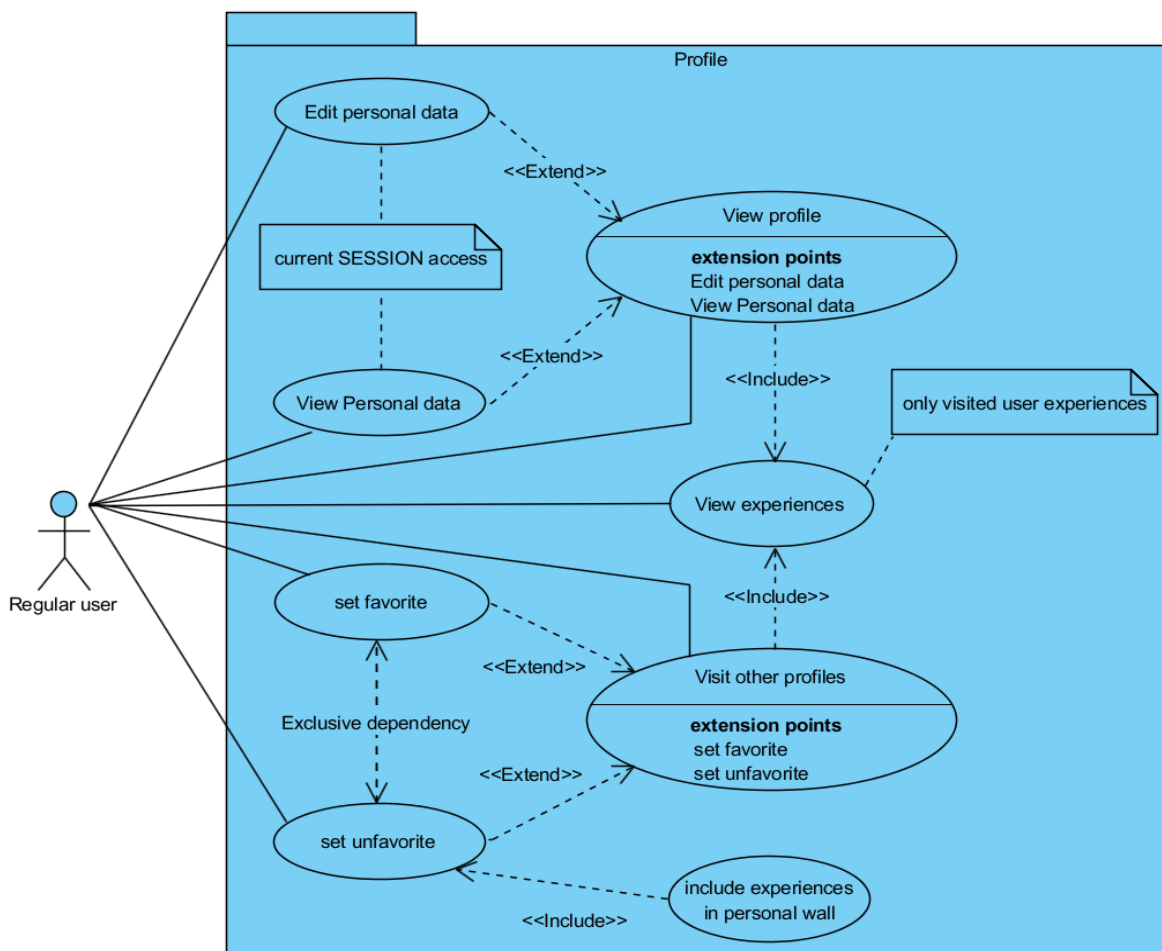


Figura 6: Diagrama del paquete Profile



### *Especificación del caso de uso “View Personal data”*

El caso de uso *View Personal data* describe el proceso de consultar los datos personales de un usuario. Una vez que el usuario tenga su sesión abierta en el sistema y tenga en su página personal abierta, podrá consultar sus datos personales seleccionando la pestaña *Information* en la misma página personal.

### *Especificación de casos de uso “Edit personal data”*

Además de poder consultar los datos personales un usuario podrá realizar cambios en sus datos. El caso de uso *Edit personal data* describe el proceso de edición, eliminación y ampliación de los datos personales por parte del usuario propietario de la cuenta. Una vez que el usuario tenga su sesión abierta en el sistema y tenga en su página personal abierta, podrá consultar sus datos personales seleccionando la pestaña *Edit profile* en la misma página personal.

### *Especificación del caso de uso “Visit other profiles”*

Además de permitir explorar experiencias de otros usuarios, Feedback debe permitir visitar la página personal del autor de una experiencia y ver otras experiencias compartidas por este usuario, haciendo clic en el nombre del autor en la cabecera de una experiencia o en su foto, el sistema nos lleva a visitar su página personal, mostrando sus experiencias en su orden cronológico.

### *Especificación de casos de uso “Set favorite”*

Cuando estamos visitando la página personal del autor de una experiencia, puede que nos parezca interesante su perfil o que el tipo de sus experiencias pueden ser útiles para nosotros y si deseamos recibir sus experiencias en el futuro, podemos seguirle, dando clic en el botón “Favorite” en la página de su perfil, el sistema registra este usuario como favorito y lo tendrá en cuenta en el futuro mostrando sus experiencias en nuestro muro.

### *Especificación de casos de uso “Set unfavorite”*

Si el contenido de las experiencias de la persona seguida deja de ser interesante o por cualquier otro motivo no deseamos seguir recibiendo experiencias de este contacto, podemos dejar de seguir a este usuario, dando clic en el botón “Unfavorite”, el sistema eliminará este usuario de nuestros favoritos.

Ficha de caso de uso "View profile"			
CU-5	View profile		
Fecha	15/09/2015		
Objetivos asociados	Acceder a la página personal.		
Descripción	El usuario accede a su página personal donde puede consultar los datos de su perfil y sus experiencias creadas.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación.		
Secuencia normal	Paso	Acción	Descripción
	1	Sesión abierta.	El caso de uso comienza cuando el usuario tenga su sesión abierta en el sistema.
	2	Presionar en menú "Profile"	El usuario seleccionar la opción "Profile" en el menú lateral, también puede acceder haciendo clic en la cabecera de cualquier post creado por él.
	3	Comprobación de permisos	El sistema comprueba los datos de sesión actuales y selecciona los permisos de acceso adecuados al perfil.
	4	Acceso a la página personal	El sistema permitirá al usuario el acceso a la página de perfil, donde por defecto se mostrarán sus experiencias personales y otras dos pestañas, "Information" para acceder los datos personales y "Edit personal data" para edición de perfil.
	5	Fin de caso de uso	Una vez mostrados los datos termina el caso de uso.
Post-condición	Ninguna		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/flujo Alternativos
	Paso 3	Flujo alternativo de visualización de perfil.	Si el perfil es de otro usuario, no aparecerá la pestaña "Information" para acceder los datos personales ni la pestaña "Edit personal data" para edición de perfil.

Tabla 16: Caso de uso "View profile"

Ficha de caso de uso "View personal data"			
CU-6	View personal data		
Fecha	15/09/2015		
Objetivos asociados	consultar los datos personales de su perfil.		
Descripción	El usuario accede a sus datos personales en su página personal.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación y tener la página personal abierta.		
Secuencia normal	Paso	Acción	Descripción
	1	Página personal del usuario.	El caso de uso comienza cuando el usuario tenga su sesión abierta en el sistema y tenga en su página personal abierta.
	2	Presionar "Information"	El usuario seleccionar la pestaña "Information" en la misma página personal.
	3	Validación y acceso	El sistema comprueba los datos de sesión actuales para comprobar si hay permiso de visualizar datos personales.
4	Fin de caso de uso	Una vez mostrados los datos personales del usuario termina el caso de uso.	
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/flujo Alternativos
	Paso 3	Flujos alternativos en la sesión de usuario: La sesión ha caducado o se está visitando el perfil de otro usuario.	Caso1: No aparecerá la pestaña "Información" para acceder los datos personales. Caso 2: Si la página estaba precargada antes de que se cierre la sesión, al hacer clic en la pestaña "Información" se recarga la página de perfil. Volver al paso 1.

Tabla 17: Caso de uso "View Personal data"

Ficha de caso de uso "Edit personal data"			
CU-7	Edit personal data		
Fecha	15/09/2015		
Objetivos asociados	Editar los datos personales del usuario		
Descripción	El usuario podrá actualizar o cambiar sus datos personales con los que cuenta en ese momento.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación y tener la página personal abierta.		
Secuencia normal	Paso	Acción	Descripción
	1	Página personal del usuario.	El caso de uso comienza cuando el usuario tenga su sesión abierta en el sistema y tenga en su página personal abierta.
	2	Presionar "Edit profile"	El usuario seleccionar la pestaña "Edit profile" en la misma página personal.
	3	Formulario para editar datos personales.	El sistema muestra el formulario "Edit profile" mostrando los datos actuales para poder editarlos.
	4	Editar datos	El usuario puede realizar los cambios en los campos que desee y presionar el botón "enviar" o "cancelar" si desea deshacer dichos cambios.
	5	Procesamiento de datos.	Una vez los cambios se han enviado, el sistema valida los datos ingresados y los datos de sesión para proceder a la validación y actualización de datos.
	6	Fin de caso de uso	Una vez mostrados los datos ya actualizados en el mismo formulario termina el caso de uso.
Post-condición	Los datos personales, se quedarán modificados en la base de datos y mostrados al usuario al final del proceso.		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/Flujo alternativos
	Paso 4	Cancelación de la operación: Usuario no confirma el aviso y cancela la operación	El sistema no procede a la eliminación de la cuenta. Volver al paso 2 del flujo base.
	paso 5	Excepción en la validación de datos del formulario: Algún dato no tiene un formato válido.	El sistema notifica el dato y el formato incorrecto. Volver al paso 4 del flujo base.

Tabla 18: Caso de uso "Edit personal data"

Ficha de caso de uso "Visit other profiles"			
CU-8	Visit other profiles		
Fecha	06/10/2015		
Objetivos asociados	Visitar el perfil de otro usuario		
Descripción	Visitar la página personal del autor de una experiencia y consultar otras experiencias compartidas en su página personal.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación.		
Secuencia normal	Paso	Acción	Descripción
	1	Iniciar Aplicación	El caso de uso comienza cuando el usuario inicia la aplicación web en el su ordenador, tableta o móvil. No es necesario tener sesión abierta.
	2	Hacer clic en el nombre del usuario a visitar	El usuario hace clic en el nombre o foto del autor en la cabecera de una experiencia.
	3	Fin de caso de uso.	Una vez el sistema nos lleva a visitar la página personal del autor, mostrando sus experiencias en su orden cronológico, termina el caso de uso.
Post-condición	En la pantalla aparece la página personal del usuario visitado.		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/Flujo alternativos
	Paso 3	Flujos alternativos en la sesión de usuario: El usuario visita su propia página.	Si el usuario hace clic en el nombre o foto de un post creado por el mismo, tendrá acceso a su propio perfil con todos los permisos, datos personales y edición de perfil.

Tabla 19: Caso de uso "Visit other profiles"

Ficha de caso de uso "Set favorite"			
CU-9	Set favorite		
Fecha	07/10/2015		
Objetivos asociados	Seguir a otro usuario.		
Descripción	Seguir a otro usuario convierte su perfil en favorito y permite recibir sus experiencias en el futuro.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación, no puede ser seguidor de esta persona en este momento.		
Secuencia normal	Paso	Acción	Descripción
	1	Iniciar Aplicación	El caso de uso comienza cuando el usuario tenga su sesión abierta en el sistema y tenga en la página personal del usuario a seguir abierta.
	2	Clic en "Favorite"	El usuario hace clic en el botón "Favorite" en la página de su perfil.
	3	Procesamiento de datos.	El sistema registra este usuario como favorito para tenerlo en cuenta en el futuro, mostrando sus experiencias en nuestro muro.
	4	Fin de caso de uso	Una vez cambia el aspecto del botón "Favorite" a "Unfavorite" se ha llegado al fin del caso de uso.
Post-condición	Una relación que se crea entre los dos usuarios y se almacena en la base de datos, hace que el usuario visitante recibirá las próximas experiencias del usuario visitado en su muro.		

Tabla 20: Caso de uso "Set favorite"

Ficha de caso de uso “Set unfavorite”			
CU-10	Set unfavorite		
Fecha	07/10/2015		
Objetivos asociados	Dejar de seguir a otro usuario.		
Descripción	Eliminar a un usuario como favorito y dejar de recibir sus experiencias en nuestro muro.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación y tiene que ser seguidor de esta persona en este momento.		
Secuencia normal	Paso	Acción	Descripción
	1	Iniciar Aplicación	El caso de uso comienza cuando el usuario tenga su sesión abierta en el sistema y tenga en la página personal del usuario al que hay que dejar de seguir.
	2	Clic en “Unfavorite”	El usuario hace clic en el botón “Unfavorite” en la página de su perfil.
	3	Procesamiento de datos.	El sistema elimina este usuario como favorito y dejaremos de recibir sus experiencias en nuestro muro.
	4	Fin de caso de uso	Una vez cambia el aspecto del botón “Unfavorite” a “Favorite” se ha llegado al fin del caso de uso.
Post-condición	Una relación entre los dos usuarios quedará eliminada desde la base de datos y el usuario visitante dejará de recibir experiencias del usuario visitado.		

Tabla 21: Caso de uso “Set unfavorite”

### 2.4.3 Casos de Uso del Paquete Experience

En la Figura 7, se muestra el siguiente diagrama del Paquete Experience donde un usuario puede crear y añadir post de una nueva experiencia u opinión sobre algún objeto, una vez agregado un post es visible para todos los usuarios permitidos. Además, se puede buscar un post, comentarlo recomendarlo o criticarlo.

Los casos de uso del Paquete Experience se describen a continuación. Una descripción más detallada de los casos de uso de este diagrama, se detalla en la Tabla 22, Tabla 23 y la Tabla 24.

#### *Especificación del caso de uso “Show experience”*

El caso de uso “Show experience” es el caso de uso al más alto nivel de abstracción en este diagrama, ya que algunos casos de uso lo incluyen por defecto como *Search experience* y *Create experience* y otros extienden de él como *Agree/disagree experience* o *Edit/Delete experience*, todos son más específicos y dependen de una acción concreta del usuario.

Además, *Show experience* es un caso de uso que no tiene ninguna restricción, ya que se puede ver una experiencia no siendo un usuario registrado, sin embargo, los casos de uso que extienden de él como son restringidos para usuarios registrados.

Así que *Show experience* es un caso de uso generado por el sistema y no depende de ninguna acción directa del usuario. Por lo cual su descripción se concreta más en la descripción de cada uno de los casos de uso relacionados con él.

## Diagrama de casos de uso Experience

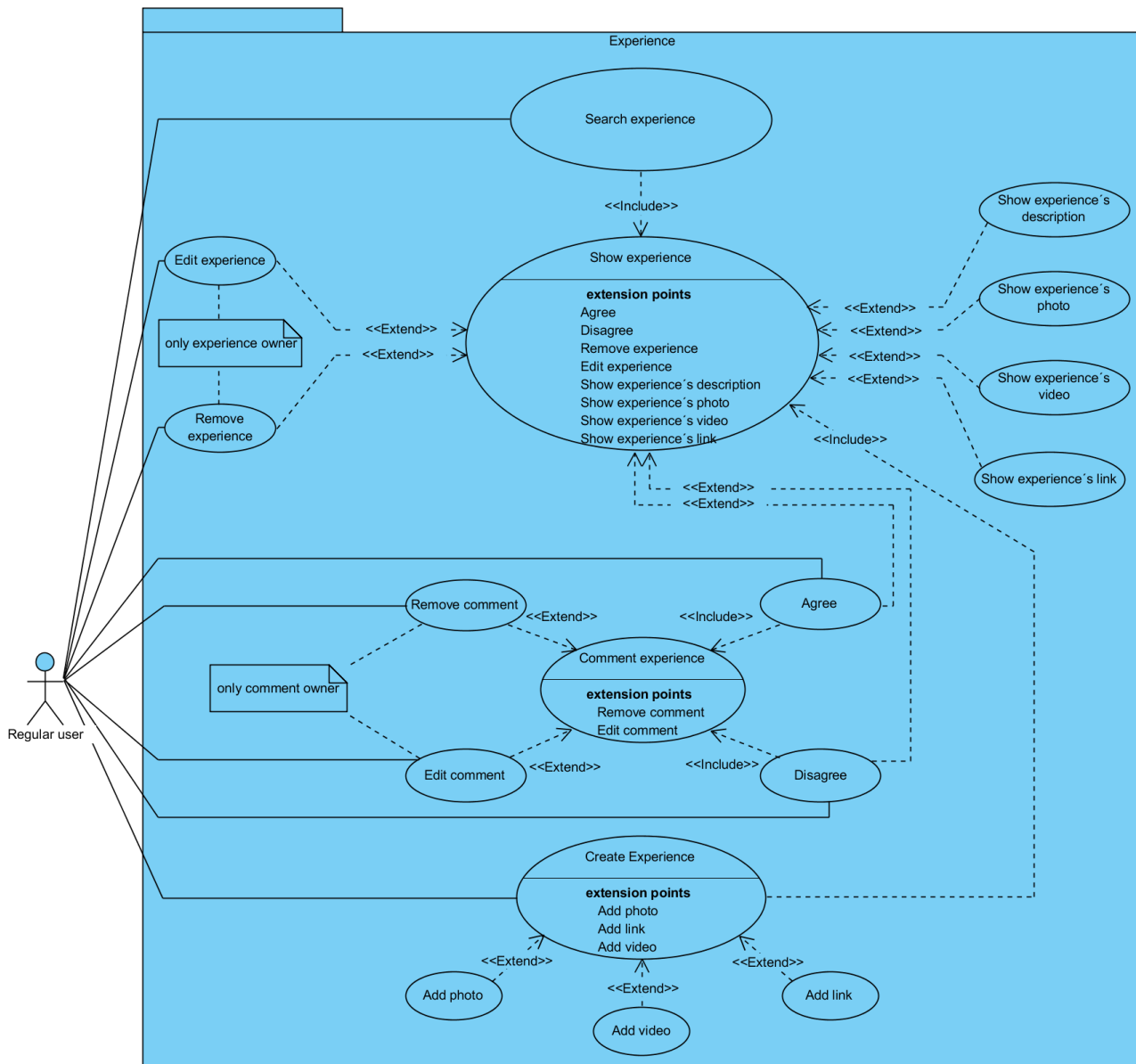


Figura 7: Diagrama del paquete Experience

### Especificación del caso de uso "Create experience"

El usuario dispone de una herramienta para compartir sus experiencias u opinión sobre algún objeto, servicio o una experiencia personal. La herramienta permite no solo redactar la experiencia con una descripción en formato texto, sino documentar la experiencia capturando una foto o video en tiempo real o compartiendo algún fichero multimedia guardado en su dispositivo. O simplemente compartir algún enlace de un video de Youtube. Una vez el usuario termina de formular la experiencia, presiona el botón enviar, pero antes de que se comparta la experiencia en la red, el sistema cuenta con sistemas de validación de los datos ingresados y los datos de sesión. Además de directivas de seguridad para evitar ataques como incrustación de código. Luego el sistema proceder a almacenarlo el contenido en la base de datos y al final publicarlo. El caso de uso termina cuando se mostrará la experiencia en el muro del usuario.

### Especificación del caso de uso “Search experience”

El caso de uso “**Search experience**” describe como un usuario puede buscar experiencia que puedan aclarar alguna duda que tenga o consejo para una nueva experiencia que este afrontando. El resultado de la búsqueda es una lista de uno o más post ordenados según la búsqueda realizada.

### Especificación de los casos de uso “Agree/disagree experience”

Además, de que un usuario pueda consultar el contenido una experiencia, sería útil saber su valoración acerca de experiencia. Los casos de uso “Agree experience”, “DisAgree experience” describen el proceso de valoración de una experiencia. Para valorar una experiencia, el usuario podrá elegir si recomienda el post o lo critica describiendo con un comentario su opinión en la entrada de texto correspondiente y presiona el botón “Enter” del teclado. El sistema recibirá el texto, comprobará los datos de sesión y aplicará las directivas de seguridad para asegurar el contenido del comentario y almacenarlo en la base de datos y finalmente publicarlo.

Ficha de caso de uso “Show experience”			
CU-11	Show experience		
Fecha	10/10/2015		
Objetivos asociados	Mostrar una experiencia		
Descripción	Mostrar un Post que describe una experiencia por parte del sistema, tanto para usuarios registrados como para no registrados.		
Precondición	Ninguna.		
Secuencia normal	Paso	Acción	Descripción
	1	Iniciar Aplicación	El caso de uso comienza cuando el usuario inicia la aplicación web en el su ordenador, tableta o móvil.
	2	La página principal.	[Usuario no registrado]: podrá ver algunos post de experiencias por defecto entre los últimos creados. [Usuario registrado]: puede ver los post relacionados con su perfil o sus favoritos. Además del contenido, el usuario puede consultar el número de votos (Agree/disagree) con sus comentarios, el nombre del usuario quien lo creo, su foto de perfil, fecha y hora de creación de un post. También se puede consultar los post de otro usuario.
	3	Fin de caso de uso	Una vez aparece el post de experiencia se ha llegado al fin del caso de uso.
Post-condición	En la pantalla aparece la experiencia con los permisos de visualización edición dependiendo del propietario del post y las variables de sesión.		

Tabla 22: Caso de uso “Show experience”



Ficha de caso de uso "Create experience"			
CU-12	Create experience		
Fecha	10/10/2015		
Objetivos asociados	Crear post de una experiencia		
Descripción	Crear y compartir un post que describa una nueva experiencia.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación.		
Secuencia normal	Paso	Acción	Descripción
	1	Redactar la experiencia	El usuario dispone de una herramienta en la cabecera del muro donde puede redactar describiendo la experiencia.
	2	[Opcional] Adjuntar ficheros.	Se añadir enlaces o subir contenido multimedia.
	3	Presionar el botón "Post"	El usuario presiona el botón enviar para compartir la experiencia en la red.
	4	Procesamiento de datos.	El sistema valida los datos formulados y los datos de sesión, aplicando directivas de seguridad y almacenarlo en la base de datos
	5	Fin de caso de uso	Una vez creado el post se mostrará la experiencia encabezando el muro del usuario y termina el caso de uso.
Post-condición	En la pantalla aparece la experiencia creada con los permisos de visualización edición del propietario.		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/Flujo alternativos
	Paso 2	Un adjunto es corrupto, no tiene un formato o tamaño permitido y no se adjuntó correctamente.	El sistema notifica al usuario la excepción, y se les preguntará si desea cancelar el post o crearlo sin este adjunto.
	Paso 3	La sesión ha caducado o se está visitando el perfil de otro usuario.	El sistema notifica que la sesión ha caducado. Volver a la página principal el usuario donde podrá iniciar sesión para crear su post de experiencia.

Tabla 23: Caso de uso "Create experience"

Ficha de caso de uso "Search experience"			
CU-13	Search experience		
Fecha	10/10/2015		
Objetivos asociados	Buscar una experiencia		
Descripción	Cualquier usuario, puede realizar una búsqueda de experiencia en sistema y obtener un resultado de post que corresponden con la búsqueda.		
Precondición	Ninguna		
Secuencia normal	Paso	Acción	Descripción
	1	Presionar el botón buscar.	El usuario dispone de una herramienta en la cabecera de la página, para desplegarla, el usuario tiene que hacer clic en el botón con icono de buscar.
	2	Redactar la búsqueda.	Aparece una entrada de texto donde el usuario puede redactar su búsqueda utilizando el lenguaje natural o escribiendo palabras clave.
	3	Presionar el botón "Enter" en el teclado.	El usuario presiona el botón "Enter" en el teclado.
	4	Validación	El sistema valida texto ingresado, los datos de sesión, además de las directivas de seguridad antes de proceder a consultar el contenido correspondiente a la búsqueda en la base de datos.
	5	Procesamiento de datos.	El sistema ha procesado la búsqueda y selecciona la lista de post de experiencias encontradas.
	6	Fin de caso de uso	El caso de uso termina una vez se han mostrado las experiencias en la pantalla.
Post-condición	Se mostrarán las experiencias ordenadas según su correspondencia con la búsqueda.		
Excepciones de la aplicación o flujos alternativos	Acción	Descripción	Manejo de la excepción/Flujo alternativos
	Paso 2	Ninguna experiencia coincide con la búsqueda realizada.	El sistema notifica al usuario que ninguna experiencia coincide con la búsqueda realizada, y se le proporcionará consejos para mejorar los resultados de su búsqueda. Volver al paso 1.

Tabla 24: Caso de uso "Search experience"

Ficha de caso de uso “Agree/disagree experience”			
CU-14	Agree/disagree experience		
Fecha	10/10/2015		
Objetivos asociados	Comentar una experiencia expresado una opinión positiva o bien negativa.		
Descripción	Además de consultar una experiencia, un usuario podrá comentar su opinión de forma apoyando o criticando el post consultado.		
Precondición	El usuario debe iniciar sesión al comenzar la aplicación.		
Secuencia normal	Paso	Acción	Descripción
	1	Iniciar Aplicación	El caso de uso comienza cuando el usuario inicia la aplicación web en el su ordenador, tableta o móvil.
	2	Mostrar Post	El usuario podrá consulta la experiencia que desea opinar. además del contenido, el número de votos (Agree/disagree) con sus comentarios, el nombre del usuario quien lo creo, fecha y hora de creación de un post, tipo y categoría.
	3	Calificar Post	El usuario podrá seleccionar si recomienda el post o lo critica describiendo su opinión con un comentario
	4	Presionar el botón “Enter” en el teclado.	Una vez que se haya valorado el post se presiona el botón “enter” en el teclado
	5	Procesamiento de datos.	El sistema valida texto ingresado, los datos de sesión, además de las directivas de seguridad para proceder a almacenar el contenido del comentario en la base de datos.
	6	Fin de caso de uso	El caso de uso termina una vez se ha publicado el comentario en la correspondiente experiencia.
Post-condición	Se mostrarán las experiencias ordenadas según su correspondencia con la búsqueda.		

Tabla 25: Casos de uso “Agree/disagree experience”

## Capítulo 3.

# Diseño del sistema

### 3.1 Arquitectura de Software

Los patrones de diseño son herramientas de modelado para solucionar problemas de diseño de software. Algunos sólo son aplicables a determinados objetivos y su finalidad es permitirnos afrontar problemas de diseño encajando nuestro proyecto en un modelo. En este trabajo la arquitectura del sistema está basada en el patrón de diseño MVC (Modelo-Vista-Controlador). El MVC [8] es un patrón de diseño que nos permite identificar tres grandes componentes independientes en nuestro sistema: datos (modelo), funciones de acceso a datos (controlador) y presentación de los mismos al usuario del software (vista).

En la Figura 8, se muestran los componentes del patrón de diseño MVC de un sistema, así como la secuencia de acciones que se producirán en una interacción entre el usuario y el software.

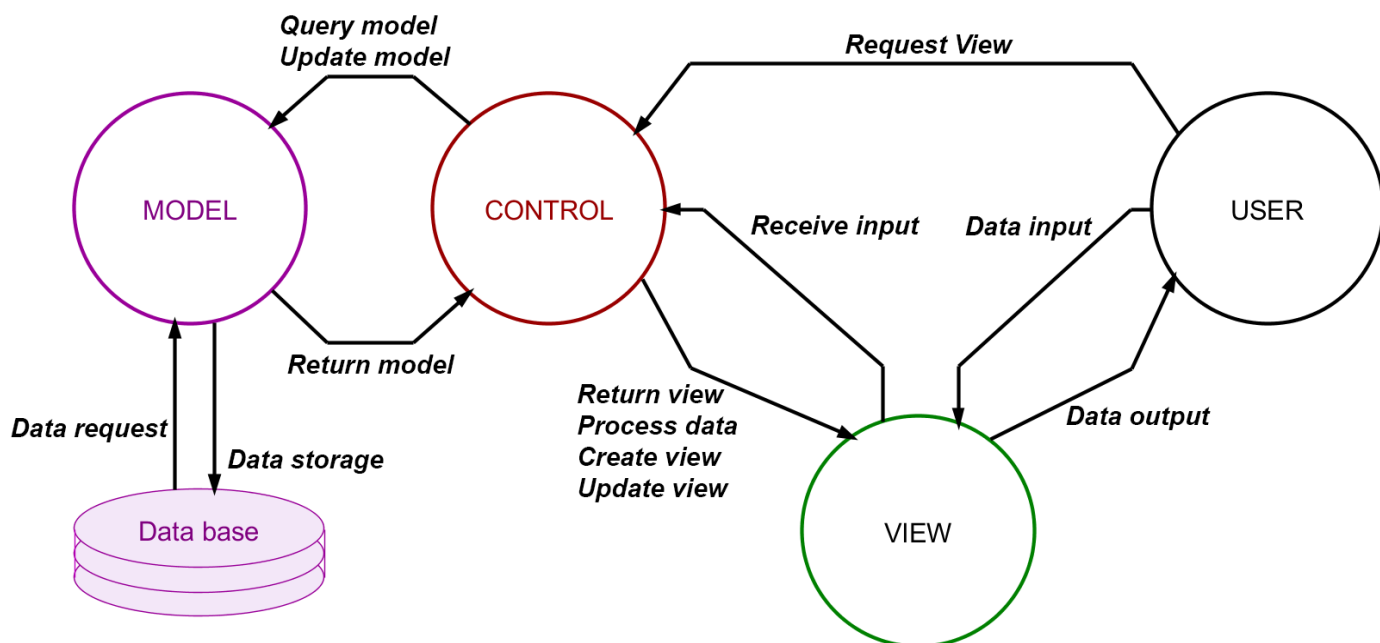


Figura 8: Arquitectura MVC

El proceso de funcionamiento del sistema es el siguiente:

En primer lugar, el cliente solicita la primera vista de usuario, una vez se le ofrece la vista solicitada, la vista recogerá los datos de la interfaz, que serán preparados y enviados al controlador (servidor) mediante el protocolo HTTP. Por su parte, el controlador se encargará de realizar las operaciones de inserción y consulta al modelo de datos y generará una respuesta que será devuelta a la vista para el usuario. Como vemos el único componente que interactúa con el modelo es el controlador, y la vista interactúa con el usuario final.

Con un esquema como éste, conseguimos no sólo separar componentes de la aplicación, lo que facilita el mantenimiento del sistema, sino permitir que cada componente pueda residir en lugares físicos distintos y simplificar el reemplazamiento de una componente funcional por otra creando un diseño mantenible, escalable y modular. Además, se podrían implementar distintas vistas de acceso, por ejemplo, vista web, móvil u otros.

### 3.1.1 Modelo

El modelo de datos es un componente funcional e independiente del sistema.

Los datos constituyen la información que manejará el software y se debe establecer un modelo que formalice correctamente sus características y atributos. El modelo de datos debe garantizar total independencia y abstracción a componentes funcionales externos de cómo se gestiona la información almacenada. En nuestro caso, el modelo de datos albergará toda la información de la plataforma que ha sido expuesta en la fase de análisis de requisitos.

El modelo de datos resulta ser el almacén de datos que típicamente acaba diseñándose sobre una base de datos. En nuestro caso será así y se procederá a describir su funcionamiento con detalle en el capítulo de implementación.

### **3.1.2 Vista**

La vista de los datos básicamente constituirá la interfaz gráfica de usuario y utilizará el controlador para acceder a los datos y mostrarlos al usuario.

La función principal de esta segunda capa es la de presentar los datos de la aplicación al usuario y ofrecer elementos de interfaz de entrada y salida para que el usuario pueda interactuar con el sistema. A su vez este conjunto de funcionalidades nunca accederá a los datos directamente y en su lugar será el controlador quien haga de capa intermediaria.

Para el caso de este sistema la vista será el sitio web de la aplicación FeedBack. En el capítulo de implementación, veremos el diseño de esta capa y la implementación de la misma.

### **3.1.3 Controlador**

Por último, tenemos el controlador, que es el conjunto de funciones de acceso a los datos, que nos permitirán leerlos, escribirlos y modificarlos. Estas funciones en ningún momento manipularán los datos directamente, esa tarea está asignada al propio modelo de datos, que deberá proveer a su vez de funciones para la manipulación de datos.

El diseño del modelo y el controlador deben ser tales que el sistema pueda acoplarse a distintas interfaces sin modificar ninguna de estas dos capas. En el siguiente apartado veremos cómo interactúa el controlador y el modelo de datos y las clases de cada uno con sus atributos y funciones.

## **3.2 Diagrama de clases**

Para llevar a cabo la implementación, desarrollaremos diagramas que permitirán visualizar los distintos ámbitos que esperamos analizar con respecto a funcionalidad, procesos y estructura. En la Figura 9, tenemos el diagrama de clases que es el diagrama de más alto nivel con el que describiremos la estructura del sistema mostrando sus clases, sus atributos, métodos (incluyendo herencia, agregación, asociación entre las clases).

El controlador junto con los datos constituye el núcleo del sistema, pues podemos decir que la interfaz gráfica es una parte independiente. Por lo cual, el diagrama se divide en dos partes. Por una parte, las clases del controlador que se encargan de comunicar con el modelo, procesar los datos y generar las vistas para el usuario y por otra parte las clases del modelo que describen el proceso de recibir peticiones del controlador y el acceso a los datos.

### **3.2.1 Descripción de las clases del Modelo**

#### *DB*

La clase DB es la clase más básica del modelo, es necesaria para establecer cualquier comunicación con la base de datos, ya que en su constructor es el único que tiene la configuración del servidor y que la utiliza para crear la conexión con la base de datos, esta

clase contiene 2 métodos para realizar los diferentes tipos de consultas de acceso a la base de datos (select, insert, delete, update).

### *Query*

La clase Query contiene todos los métodos necesarios para realizar las consultas de extracción de datos del tipo (SELECT) necesarias para el modelo, esta clase incluye la clase DB que le permite acceder a la base de datos. Además, es necesario instanciar un objeto db en cada método para enviarle el mensaje de la consulta correspondiente.

### *Exec*

La clase Exec contiene todos los métodos necesarios para realizar las consultas de acceso a datos del tipo (INSERT, UPDATE y DELETE) necesarias para el modelo, esta clase también incluye la clase DB que le permite acceder a la base de datos. Además, es necesario instanciar un objeto db de la misma forma que los métodos de la clase anterior.

## **3.2.2 Descripción de las clases del Controlador**

### *Account*

La clase Account representa los datos de acceso al sistema de cada usuario, un usuario podrá obtener estas credenciales de forma única una vez registrado en el sistema, con las cuales podrá abrir sesión en la plataforma Feedback y navegar en el sistema.

### *Profile*

La clase Profile representa el resto de los datos y configuración personal del usuario, estos incluyen sus datos personales y las experiencias que ha realizado, se guardan además en el perfil los contactos preferidos del usuario.

### *Experience*

La clase Experience es la más importante de todas, pues todo gira en torno a un post de experiencia, ya que estará relacionada con un usuario (el autor del post), los comentarios, los adjuntos y además una experiencia se muestra en los muros de los perfiles de los usuarios que tienen como preferido el autor de esta experiencia. Los métodos de esta clase permiten crear, editar incluso eliminar una experiencia por parte del autor.

### *Wall*

La clase Wall es la clase de más alto nivel del controlado ya que maneja los tres métodos más utilizados por las vistas que interactúan con el usuario, searchExperience(), createExperience() y publishExperience(). Además de manejar los algoritmos de búsqueda y creación de experiencias. Organiza el muro del usuario mostrando los post de experiencias según su correspondencia con el ámbito del perfil del usuario, variable sesión y sus contactos preferido.

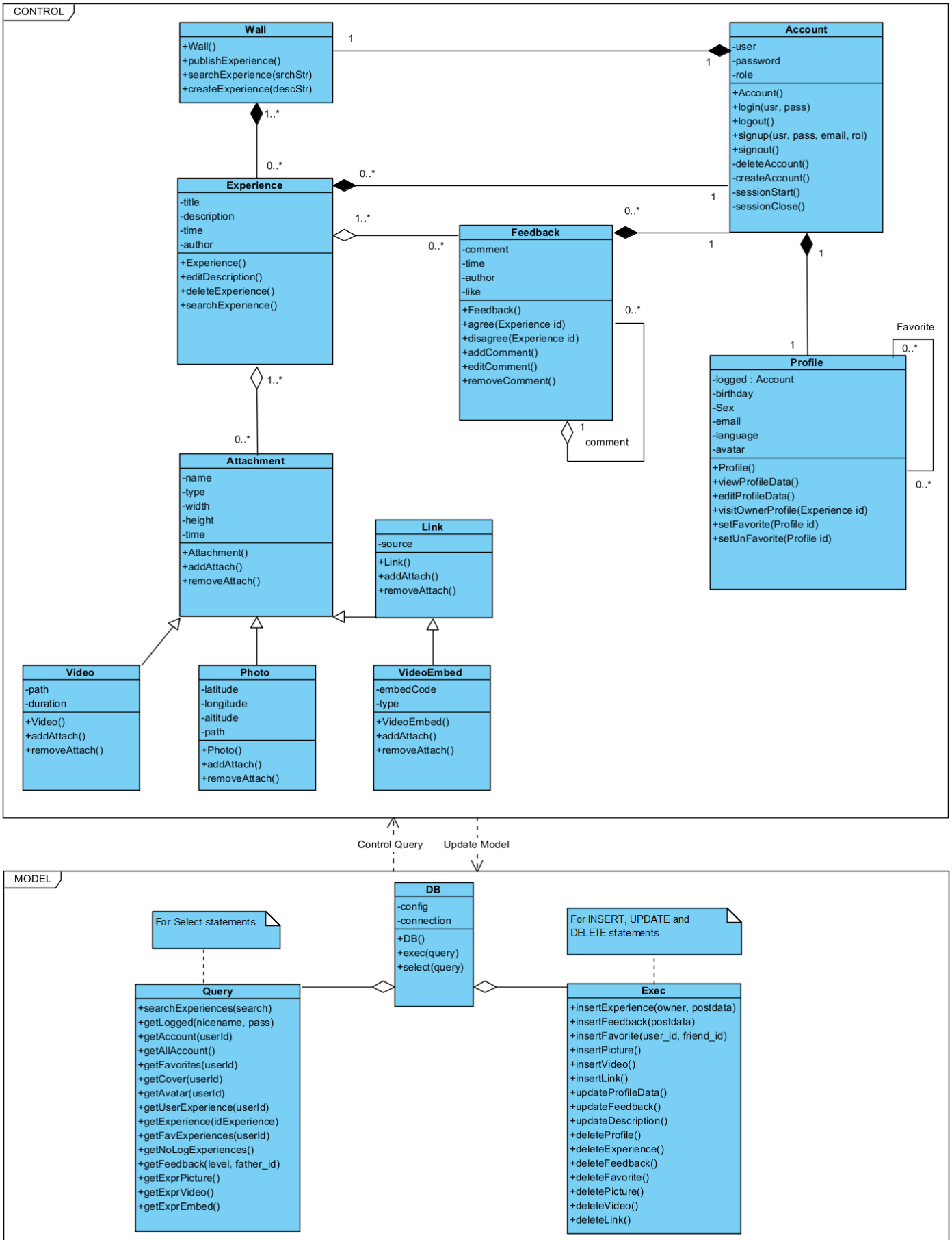


Figura 9: Diagrama de clases

### *Attachment*

Esta clase representa el adjunto o los adjuntos que puede tener una experiencia. Las clases que heredan de ella serán las que pueden instanciar adjuntos como una foto, video o un enlace externo. Sus atributos y métodos serán reutilizados por las clases herederas.

### *Photo*

La clase photo es una clase que hereda de la clase *Attachment*, instancia un objeto foto que tendrá como atributos además de la ruta de acceso a la imagen, toda la información acerca de esta foto está representada por los atributos que aparecen en el diagrama, además de los atributos heredados por la clase madre.

### *Video*

La clase video, igual que la clase photo, hereda de la clase *Attachment*, y a partir de ella se instanciarán los objeto video que tendrá como atributos además de la ruta de acceso al video, toda la información acerca de este video representada por los atributos que aparecen en el diagrama, además de los atributos y métodos heredados por la clase madre.

### *Link*

La clase link es una clase que hereda de la clase *Attachment*. Esta clase se ha desarrollado para futuras extensiones por ejemplo visualizar enlaces externos en la experiencia u cualquier otro tipo de enlaces. Por ahora solo tiene la clase *Embed* que hereda de ella y que se describe a continuación.

### *Embed*

Esta clase es para los enlaces de tipo Youtube que ofrecen un servicio para empotrar sus videos en otra página con un código relaciona con el video. Además de los atributos y métodos heredados por la clase madre, tiene dos atributos propios (type y embedCode) y son para representar por ejemplo (Youtube, código\_del\_video).

### *Feedback*

Esta clase representa un comentario sobre una experiencia, con dos posibles categorías (*Agree/disagree*) para poder categorizar los comentarios de forma que la persona que consulta la experiencia puede tener una idea rápida de cómo es la reacción en general hacia esta experiencia. Estos comentarios están relacionados con la experiencia y además con la persona que ha realizado el comentario.

## **3.3 Diagrama de Componentes**

A continuación, se modela la vista estática y dinámica de un sistema mediante el diagrama de componentes que se muestra en la Figura 10. Esta figura muestra la división del sistema en varios componentes y las dependencias entre estos.



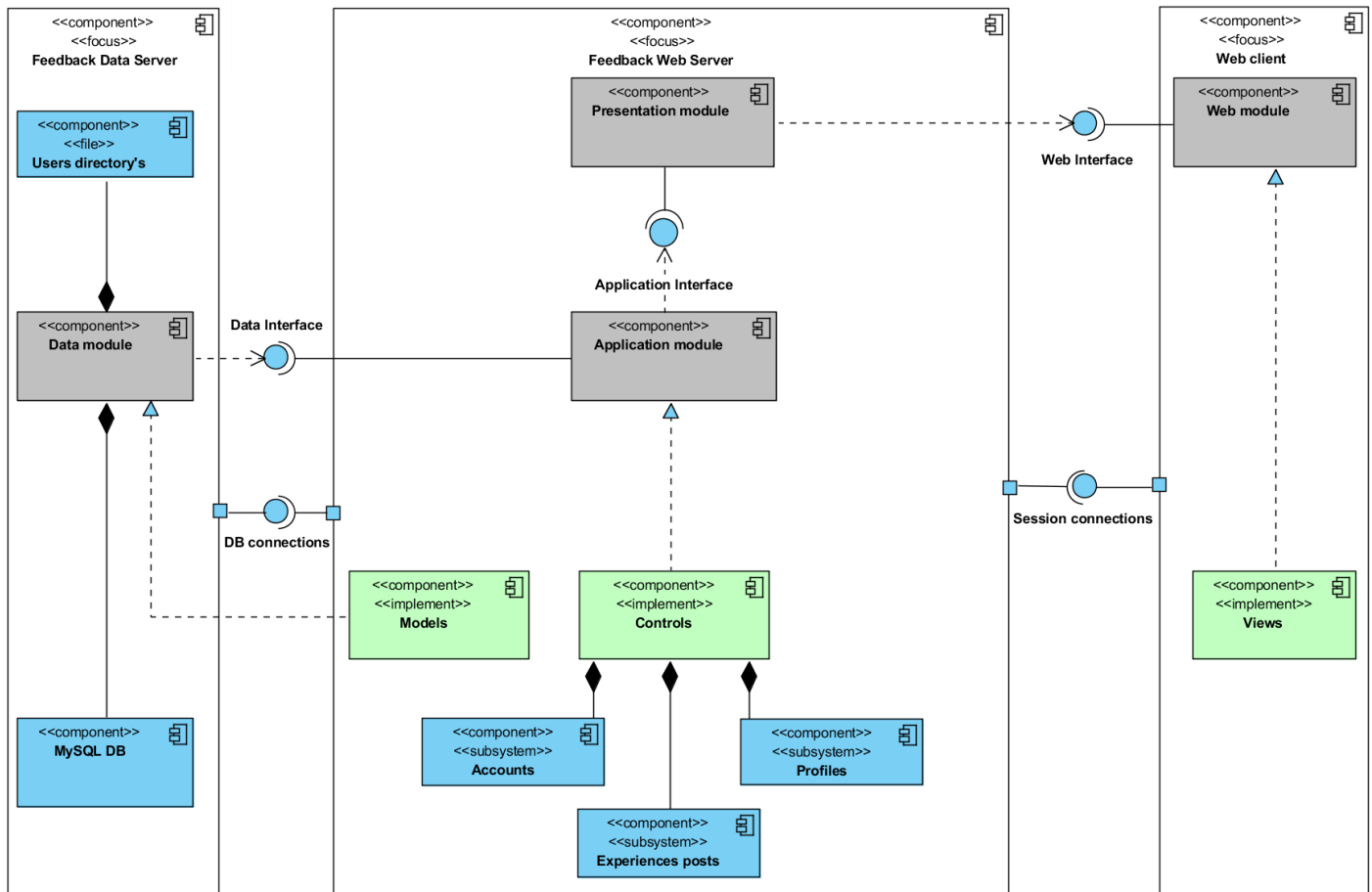


Figura 10: Diagrama de Componentes

Con el diagrama de componentes se pretende visualizar la estructura de alto nivel del sistema y su comportamiento a través de sus interfaces.

El diagrama se ha dividido en el nivel más abstracto en tres ámbitos (Data Server, Web Server y Web client) ya que son los ámbitos físicos donde operan los diferentes componentes. Las interfaces entre estos ámbitos son “Session interface” y “Data interface”.

En un nivel más concreto vemos que sistema se divide en cuatro componentes principales y son los siguientes (Web module, Presentation module, Application module, Data module), con los interfaces que los interconectan (Web interface, Application interface y Data interface). Estos componentes se implementan o se componen de módulos de más bajo nivel como los paquetes que contienen las clases con líneas de código, la base de datos o directorios de usuarios.

### 3.4 Diseño de la base de datos

El estudio y el diseño de la estructura de la base de datos de experiencias se llevarán a cabo en tres etapas, definiendo un diseño conceptual y un diseño lógico, que luego se implementa en el modelo físico usando para ello el gestor de bases de datos de nuestra elección (MySQL [9] en nuestro caso).

### 3.4.1 Diseño conceptual

En esta etapa se obtiene una estructura de la información de la futura BD independiente de la tecnología que hay que emplear y el resultado de la etapa del diseño conceptual se expresa mediante el modelo ER. En la Figura 11, se muestra el diagrama ER obtenido en esta etapa, donde se muestran las entidades relevantes del sistema de información, así como sus interrelaciones.

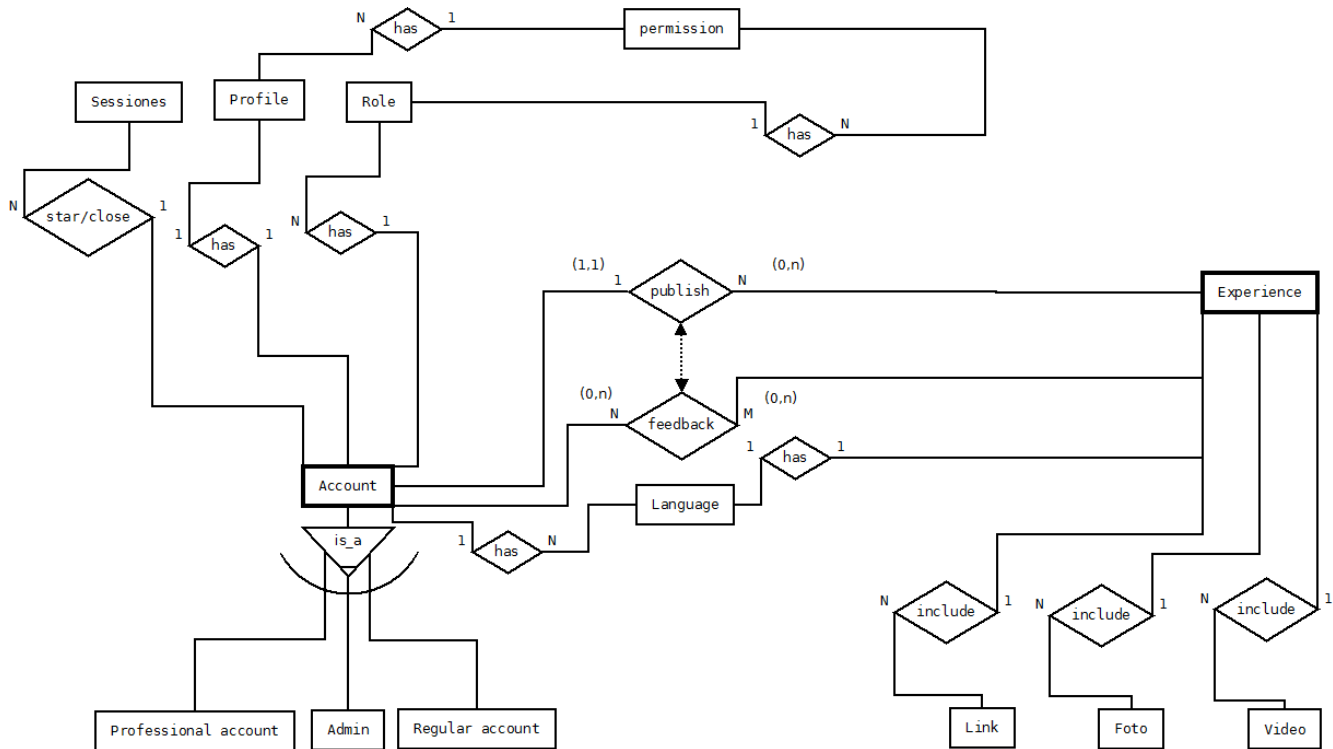


Figura 11: Modelo entidad-relación (Diagrama ER)

### 3.4.2 Diseño lógico

Una vez tenemos claro el modelo ER lo traducimos al modelo lógico, como vemos en la Figura 12. Es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de SGBD.

Con la conversión del modelo conceptual al modelo lógico. Además de obtener el modelo lógico con las entidades definitivas (tablas en el modelo físico), se obtendrá un conjunto de relaciones con sus atributos, claves primarias y claves foráneas ya que es un modelo relacional. La mayoría de las herramientas profesionales de bases de datos permiten generar automáticamente las instrucciones necesarias para crear la base de datos. (Modelo físico).

### 3.4.3 Implementación del diseño físico

El diseño físico por regla general recae en la propia base de datos, a partir del diseño lógico. Se ha transformado la estructura obtenida en la etapa del diseño lógico a un modelo físico, con el objetivo de conseguir una mayor eficiencia, además, se completa con aspectos de implementación física que dependerán del SGBD, en nuestro caso es MySQL. La base de datos final se crea con las tablas y las relaciones tal como aparecen las entidades y las relaciones del modelo lógico en la Figura 12.

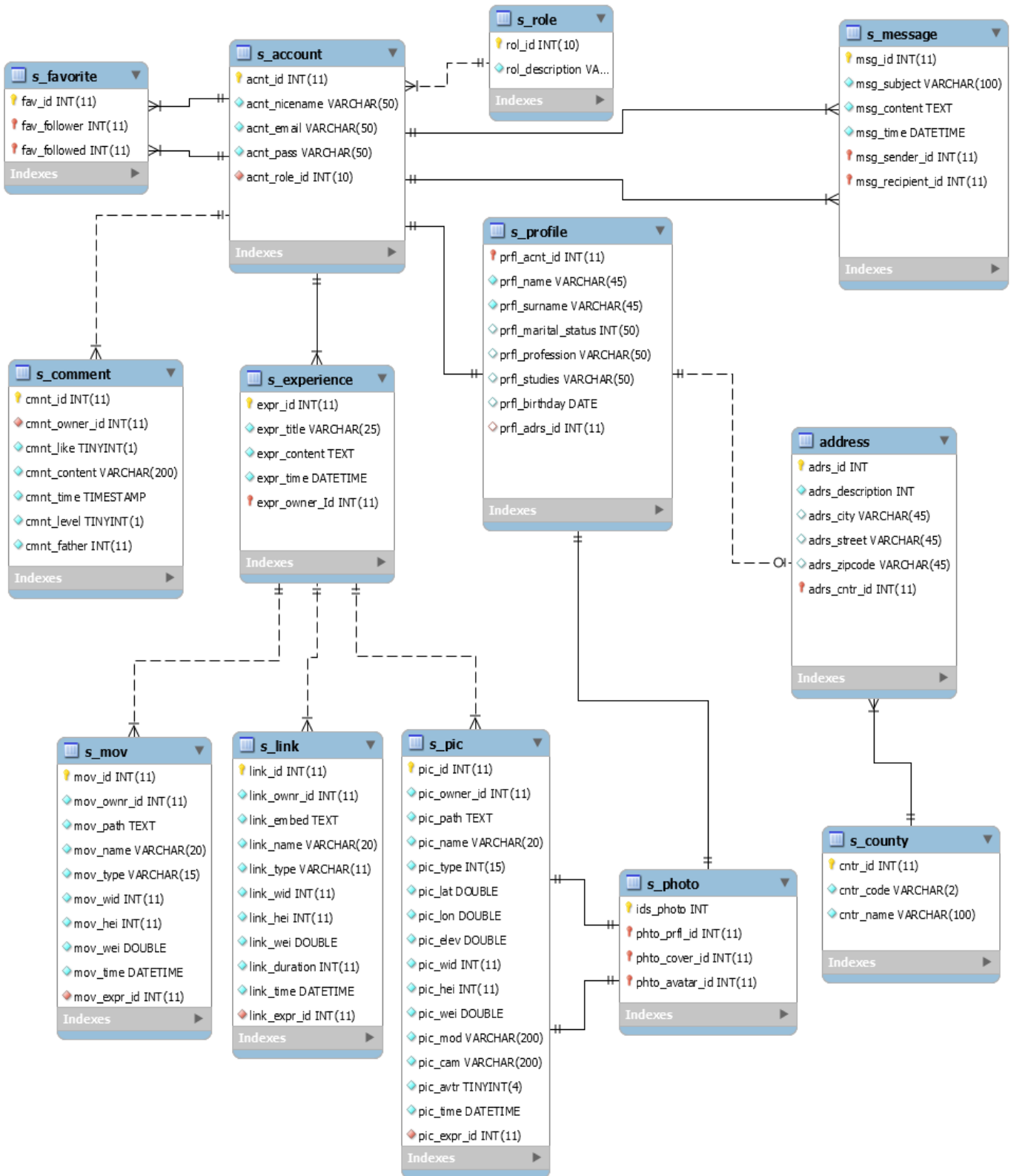


Figura 12: Modelo Relacional (Diagrama EER)

# Capítulo 4.

## Implementación

Una vez acabado el diseño, es posible implementar todas las partes del sistema. La implementación se basa principalmente en el patrón de diseño MVC que divide las aplicaciones en tres niveles de abstracción (modelo, vista y controlador). Durante este capítulo se explicará cómo se ha llevado a cabo la implementación en cada uno de los niveles por separado, que tecnología se ha utilizado y cómo se ha estructurado el código en cada una.

### 4.1 Modelo

El modelo es el componente que realiza comunicación entre el controlador y los datos. Este componente estará compuesto por dos módulos ‘class-exec’ y ‘class-query’ que requieren el módulo ‘DB’. Estos módulos contienen los métodos PHP que incluirán la funcionalidad necesaria para recibir parámetros de entrada por el método POST de HTTP, preparar las estructuras y tipos de datos necesarios y enviarlos al controlador mediante un mensaje HTTP. La estructura estos módulos y sus métodos se puede ver claramente en el diagrama de clases Figura 9.

La estructura básica del método de estos dos módulos sigue las siguientes sintaxis:

#### 4.1.1 Sintaxis de los métodos de consulta en la clase “class-query”

```
public function funcionDeConsulta($parametro, $parametro2,..) {
    global $db;
    $table = 's_tabla';
    $query = "QUERY SQL TIPO SELECT;";
    $obj = $db->select($query);
    if (!$obj) {
        return "No object found";
    }
    return $obj;
}
```

#### 4.1.2 Sintaxis de los métodos de cambio en la clase class-exec’

```
public function funcionDeCambio($parametro, $parametro2,..) {
    global $db;
    $table = 's_status';
    $query = "QUERY SQL TIPO INSERT, UPDATE o DELETE;";
    return $db->exec($query);
}
```

Este módulo no se encarga de generar ninguna vista ni crea ningún código HTML.

## 4.2 Vista

La vista, en este caso es una interfaz web que se utiliza desde un navegador web. La tecnología empleada para su implementación PHP para la generación de las vistas. Se han utilizado el conjunto Lenguaje de Marcas HTML, CSS y Javascript, que construcción de las vistas.

Por otro lado, se ha hecho especial énfasis en separar la declaración de componentes de la vista de sus estilos asociados. Así el generador de páginas tan solo producirá páginas HTML, cuyos estilos estarán contenidos a parte en una hoja de estilos CSS y su framework bootstrap. Por otra parte, tenemos la inclusión del lenguaje Javascript, desde algunas sencillas funciones que cambian dinámicamente algún componente de la vista.

El hecho de que la aplicación sea accesible vía web, se ha tenido en cuenta a la hora de diseñar todas las posibles vistas para los diferentes dispositivos móviles de forma que la aplicación permita la creación de múltiples vistas independientes a partir del mismo contenido [Responsive-Web-Design] [6]. Esto es, diseñar la web de forma que una misma vista sea visualizada de forma óptima en dispositivos diferentes sin necesidad de hacer desarrollos individuales para cada tamaño de pantalla o dispositivo, en la Figura 13 muestra cómo se adapta el diseño de la aplicación Feedback en cuatro tamaños diferentes utilizando el mismo diseño web.

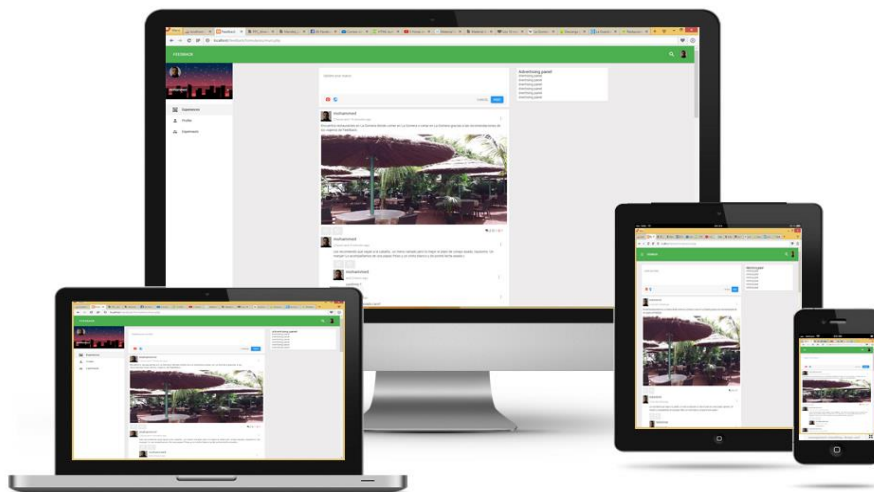


Figura 13: Responsive Web Design

Por otra parte, se ha creado una plantilla con las partes estáticas del código HTML y sus correspondientes hojas de estilo CSS, evitando así códigos redundantes en varias páginas. La Figura 14 muestra la maquetación de la página *inicio* utilizando la plantilla. La única parte que cambia es contenido dinámico que la parte central que interactúa con las entradas del usuario y su estado de sesión.

La vista se ha diseñado para ser adaptada una aplicación móvil, para no tener que diseñar desde cero el diseño para la aplicación móvil. Con PhoneGap se podría para convertir aplicaciones desarrolladas con HTML+Javascript+CSS en aplicaciones nativas para dispositivos móviles. Finalmente, no se hizo porque los resultados obtenidos con la interfaz web son satisfactorios, pero también porque se ha dado más prioridad en invertir el tiempo en terminar aspectos más importantes del sistema.

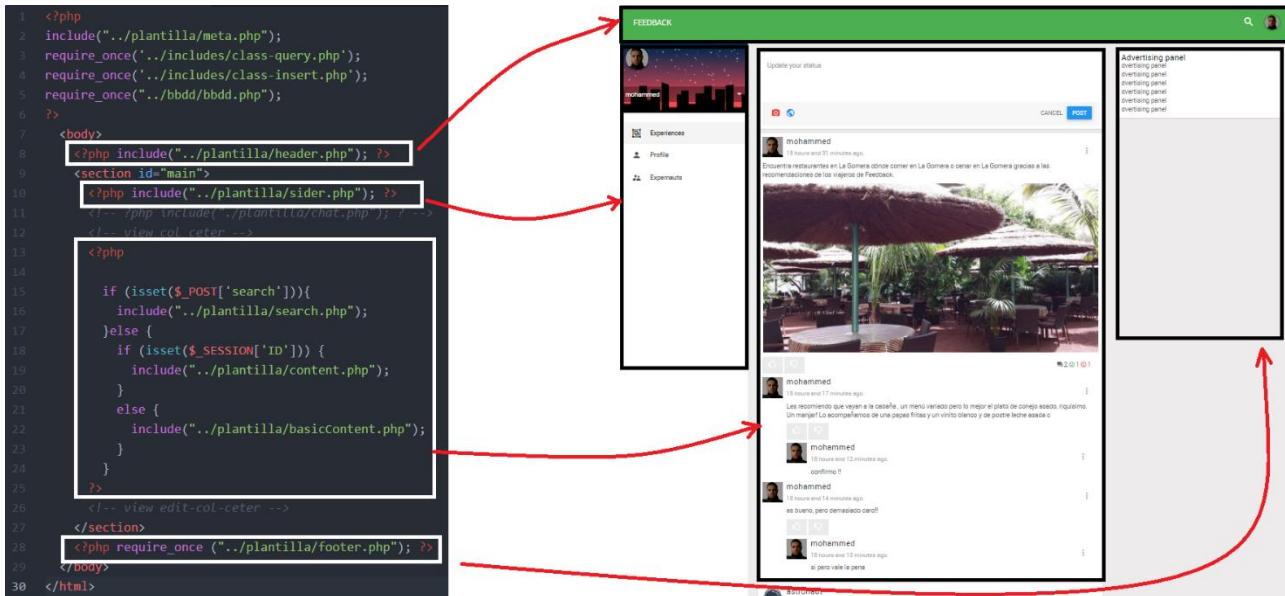


Figura 14: Maquetado de la plantilla en el diseño

## 4.1 Controlador

El controlador es el conjunto de clases que por una parte se encargan de comunicar entre sí y con el modelo para obtener o insertar datos y por otra parte son los responsables de enviar los datos a los módulos de las vistas para la generación de las vistas utilizando las funciones para la visualización de interfaces presentadas al usuario, que serán rellenas con los datos pertinentes obtenidos del Modelo.

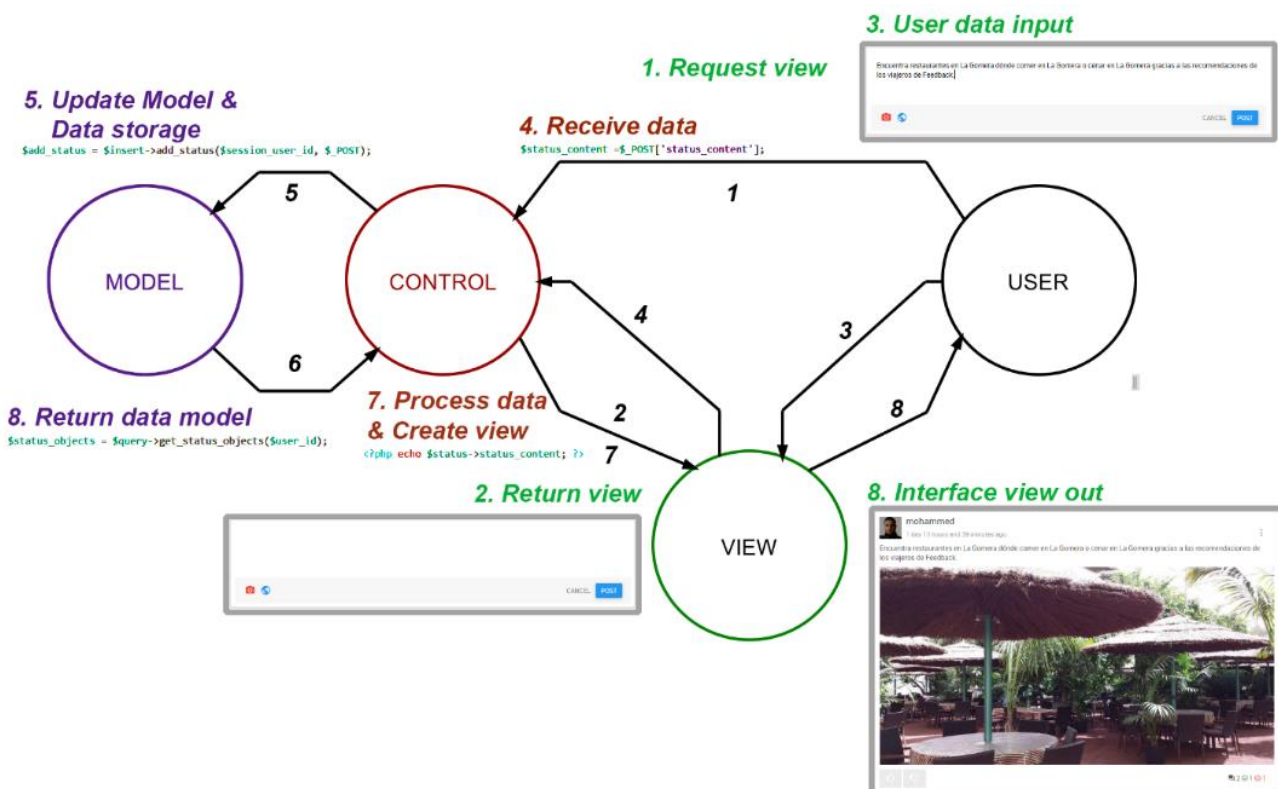


Figura 15: Crear post de experiencia en MVC

El proceso de implementación de las clases del controlador es prácticamente implementar las funciones de las clases descritas en el diagrama de clases en la Figura 9.

Estas clases contienen métodos que reciben datos como argumentos desde la vista, se relaciona con las funciones del modelo descritas en el apartado anterior 40, para responder a las peticiones del usuario desde la vista.

Un ejemplo muy claro es entre las clases *Experience*, *Feedback* (comentario) y cómo comunican con el modelo, que se describe a continuación.

## Implementación de feedback de experiencias

La clase *Experience* es una clase que permite instanciar objetos de experiencias. Su constructor es el responsable de su creación y necesita los siguientes atributos (title, description, time, author) para la creación del objeto.

```
public function addFeedback($_POST){
    $level = $_POST['$level'];
    $father_id = $_POST['father_id'];
    $comment = $_POST['Feedback_down']; //If the feedback is positive
    $author = $_SESSION['ID'];
    $fatherID = $_POST['$fatherID']; // $fatherID is the objectID to be commented (experience or other comment)

    if (empty($comment)) {
        $like = true; //like=true, when the comment is positive.
        $level = 1; //level1 comment is a comment of a post.
        $comment = $_POST['Feedback_up'];
    }

    $_cmntObj = new Feedback($comment,$time,$author,$like,$level,$fatherID); //Feedback is the comments class.
    $cmntID = $exec->insertFeedback($_cmntObj); //request model to insert this feedback(comment)
    $comments[] = $cmntID;
}
```

Figura 16: Metodo addFeedback de la clase Experience

Además, la clase dispone de una serie de métodos, algunos son para realizar cambios en atributos del objeto, y otros son para añadir atributos como adjuntos o comentarios. El método *addFeedback(\$\_POST)* recibe el objeto POST que contiene los datos enviados por POST desde la vista. En base a las variables recibidas, primero se crea un objeto *Feedback* (un comentario) y luego invoca el método *InsertFeedback* de la clase *Exec*. Cuando se recibe el objeto (de la clase *feedback*), se crea una consulta de tipo INSERT e inserta los atributos del objeto como valores en una nueva línea en la base de datos, esta línea es un comentario nuevo de esta experiencia, en la Figura 16 se muestra el método *addFeedback*. Otro ejemplo es el proceso de implementación del buscador que a continuación se describe la evolución de este algoritmo.

## Implementación del buscador de experiencias

En SQL las búsquedas normalmente se realizan mediante el comando LIKE introduciendo una cadena de texto y recibir como resultado los registros coincidentes, un ejemplo sería el siguiente:

```
SELECT * FROM s_experience WHERE expr_title LIKE '%$busqueda%' OR TITULO LIKE '%$busqueda%';
```

Pues esta consulta mostrará todos los artículos que en su título o en su contenido aparezca la cadena de búsqueda tal y como la introducimos. Cualquier cambio en el orden de las palabras o separación no mostrará los resultados deseados. En nuestro caso esto supondrá una limitación, por lo cual necesitamos una solución más eficiente. Utilizaremos los índices FULLTEXT específicamente indicados para búsquedas más complejas. Pues bien, esto implica crear un índice FULLTEXT [9] con todos los campos de la tabla que deseamos incluir en nuestra búsqueda. la instrucción es la siguiente:

```
ALTER TABLE s_experience ADD FULLTEXT search (expr_title, expr_content);
```

Una vez creado el índice, la instrucción SQL para buscar será:

```
SELECT * FROM s_experience WHERE MATCH(expr_title, expr_content) AGAINST ('$busqueda');
```

De esta manera, el buscador de experiencias es capaz de buscar en la base de datos Experiencia por una, o más palabras utilizando la función MATCH ... AGAINST ... que encuentra el texto buscado, usando consultas en lenguaje natural parecido a como lo hacen los motores de búsqueda. Para perfeccionar la búsqueda podemos utilizar la puntuación y limitar el número de registros a devolver, sentencia SQL sería la siguiente:

```
SELECT *, MATCH(expr_title, expr_content) AGAINST('$busqueda') AS score FROM s_experience WHERE MATCH(expr_title, expr_content) AGAINST('$busqueda') ORDER BY SCORE DESC LIMIT 10";
```

Internamente, se calcula una puntuación en función de cómo aparecen los términos buscados dentro del título o la descripción del post de experiencia. La sentencia SQL anterior, devuelve los 50 primeros resultados encontrados ordenados de más a menos puntuación. El valor de la puntuación es un número decimal comprendido entre 0 y 1 por cada ocurrencia del patrón de búsqueda. El cálculo de la puntuación es un algoritmo interno del sistema gestor de base de datos. Las búsquedas realizadas con MATCH ...AGAINST fallan cuando la búsqueda contiene una sola palabra. Por el contrario, son unas búsquedas rapidísimas que producen mejores resultados que otros métodos más rudimentarios como el LIKE cuando se trata de varias palabras o frases. La solución fue utilizando una búsqueda simple con LIKE en el caso de una sola palabra, y el método MATCH...AGAINST en el caso de varias. A continuación se muestra la implementación del algoritmo de esta solución.

```
public function searchExperiences($search) {
    global $db;
    $table = 's_account';
    $search = mysql_real_escape_string($search);
    if($search<>''){
        $token=explode(" ", $search)
        $i=count($token);
        if ($num == 1) {
            $query = " SELECT * FROM s_experience WHERE expr_title
            LIKE '%$search%' OR expr_content LIKE '%$search%' LIMIT 10;";
        } elseif ($num>1) {
            $query = " SELECT *, MATCH(expr_title, expr_content)
            AGAINST('$search') AS score FROM s_experience
            WHERE MATCH(expr_title, expr_content) AGAINST('$search')
            ORDER BY SCORE DESC LIMIT 10;";
        }
        $status_objects = $db->select($query);
        return $status_objects;
    }
    else {
        return false;
    }
}
```



### 4.1.1 Seguridad del sistema

PHP es un potente lenguaje y su intérprete puede acceder a ficheros, ejecutar comandos o abrir conexiones de red desde el servidor. Sin tomar ninguna medida, estas propiedades hacen que sea inseguro todo lo que se ejecute en un servidor web. Partiendo de un correcto ajuste de opciones de configuración y de prácticas de programación apropiadas, es posible obtener la combinación de libertad y de seguridad necesaria.

Se han aplicado una serie de consideraciones básicas de seguridad y a continuación describiremos su implementación:

**El módulo session:** Este módulo ara el control de acceso por sesión en el servidor, este módulo sólo acepta session ID válidas generadas por él mismo, rechazando cualquier session ID proporcionada por usuarios, en la Tabla 26 se muestra el acceso a la página de perfil de un usuario, en el primer caso a la izquierda, el usuario está visitando su propio perfil, el segundo a la derecha, el usuario está visitando el perfil de otro usuario:

Un usuario visitando su propio perfil	Un usuario visitando un perfil de otro usuario.
	
<p>El usuario puede acceder su página personal completa. Podrá acceder tanto a sus experiencias como la información personal, además de poder realizar cambios en la opción Edit profile.</p>	<p>Sólo se puede visitar el perfil de un usuario sin tener acceso a su información personal. Si el usuario copia y pega el enlace del perfil de otro usuario en la barra de direcciones, para acceder a sus datos malintencionadamente tendrá el mismo resultado de esta figura.</p>

Tabla 26: Control de acceso por sesión

**Otros módulos de seguridad:** El módulo de sesión no puede garantizar que la información que se almacena en una sesión sea vista sólo por el usuario que creó la sesión. Se necesita tomar medidas adicionales para proteger activamente la confidencialidad de la sesión:

- Establecer el modo de id de sesión estricto habilitando el módulo `session.use_strict_mode` en el fichero `php.ini` en el servidor `php`.

- Utilizar cookies para almacenar el id de sesión para previene ataques de pasar el id de sesión en la URL habilitando el módulo `session.use_only_cookies` en el fichero `php.ini` en el servidor `php`.

**Regeneración del id de sesión:** Utilizar la función `session_regenerate_id()` para reemplazará el id de sesión actual por uno nuevo que mantendrá la información de la sesión actual [10].

**Inyección de código SQL:** Para evitar la inyección de código SQL, PHP provee de una serie de funciones, como `mysql_real_escape_string()`, esta función antepone barras invertidas a los siguientes caracteres: `\x00`, `\n`, `\r`, `\`, `'`, `"` y `\x1a`, un ejemplo de su utilización es en el buscador de experiencias explicado anteriormente.

**Ubicación segura de documentos personales:** El sistema Feedback crea una carpeta para cada usuario donde se almacenan todos sus ficheros relacionados con sus datos personales y experiencias, una medida segura es ubicar la carpeta de los usuarios fuera del directorio del sistema. De esta forma será más difícil acceder a cualquier fichero desde el directorio del sistema.

## Capítulo 5.

# Resultados

Después de implementar todos componentes del sistema, se muestran los resultados obtenidos al final del trabajo. Para comprobar las funcionalidades, había que revisar la lista extendida de los requisitos y comprobar todos y cada uno de los requerimientos. Para poder comprobar los resultados se crearon algunos usuarios ficticios y se simularon diferentes situaciones normales, pero también anómalas que podrían darse en la realidad. En este capítulo se recogerán algunos de estas simulaciones y se mostrarán aquellos los resultados obtenidos.

### 5.1 Registro y Acceso

Feedback no obliga a los usuarios a estar ser registrados para poder consultar experiencias incluso buscar experiencias con el buscador de la aplicación, pero hasta que no acceda al sistema no podrá crear post de experiencia ni comentar experiencia de otros usuarios.

En la barra superior de navegacion en la Figura 17, el usuario dispone además del buscador, de la opción acceder con sus credenciales o registrarse en caso no dispone de un usuario.



Figura 17: Barra superior de navegación

La parte izquierda de la Figura 18 muestra el formulario de registro en el sistema, este proceso requiere que el usuario introduzca todos los datos solicitados, una vez rellenado el formulario

el usuario podrá acceder al sistema introduciendo sus credenciales del usuario (usuario y contraseña) como se muestra en la parte derecha de la Figura 18

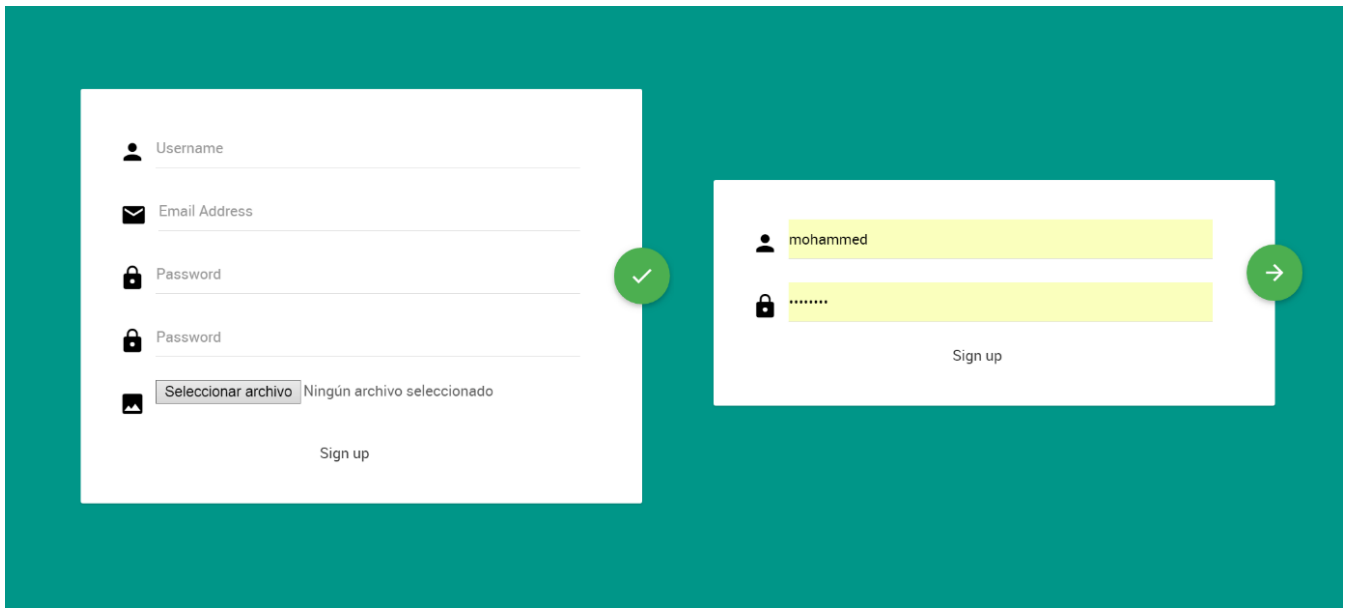


Figura 18: Formularios de registro y acceso al sistema

## 5.2 Navegador de FeedBack

Una vez el usuario ha accedido al sistema, algunas opciones del navegador cambian para ayudar al usuario a navegar entre las distintas funciones del sistema que se muestran en la Figura 19.

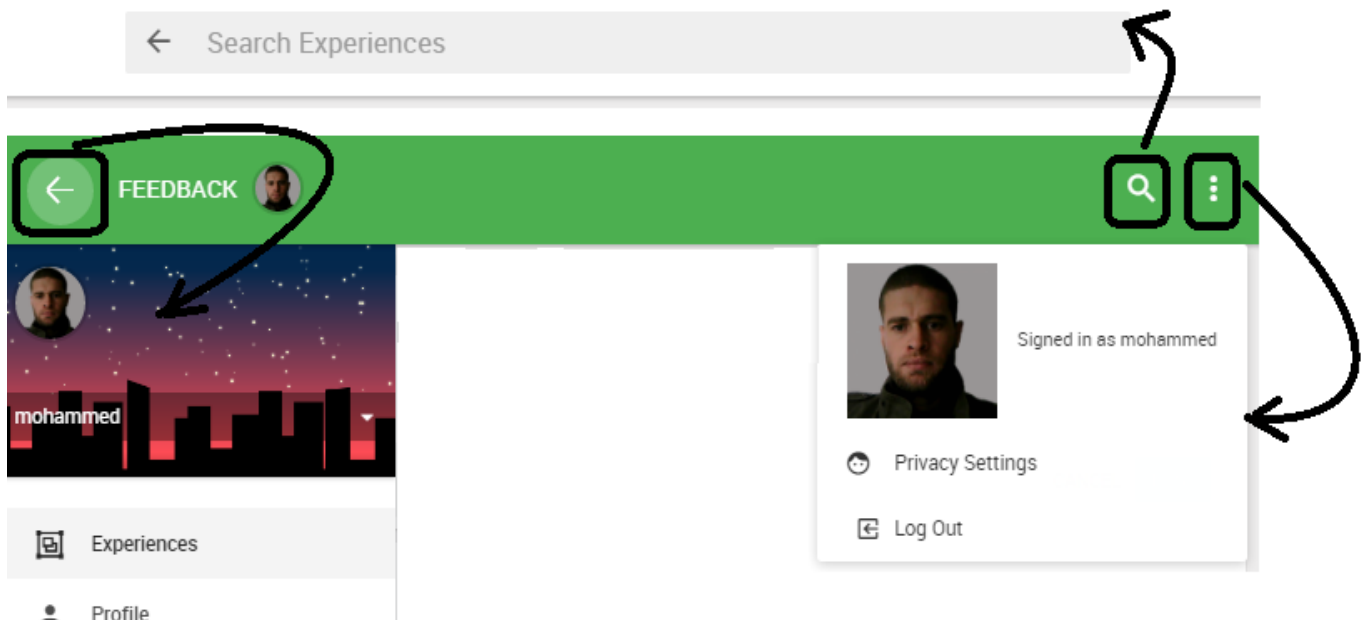


Figura 19: Navegador de FeedBack

## 5.3 Crear post de Experiencia

En la parte superior podemos documentar nuestra experiencia con texto y además podemos adjuntar ficheros multimediales para detallar más nuestra experiencia, el adjunto puede ser una imagen o video almacenados localment. También puede ser un enlace Embeded de un video externo Youtube. Se ha añadido un pre-visualizador que nos permite ver una miniatura del vídeo antes de subirlo, como se muestra en la Figura 20.

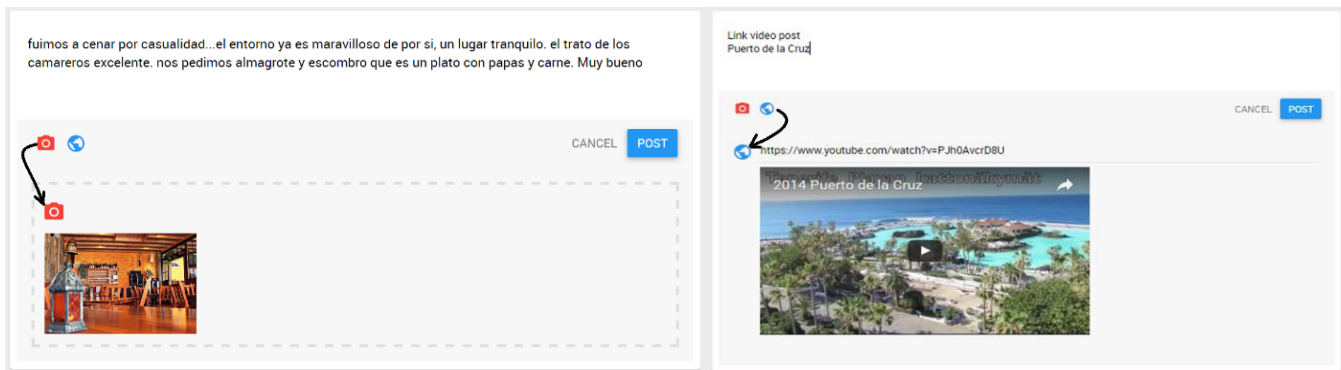


Figura 20: Creación de un post con adjuntos

Una vez tengamos el post redactado con sus correspondiente adjuntos (imagen, video, enlace,..), se publicar pulsando en el botón Post. en la Figura 21 podemos ver la experiencia creada con el adjunto (imagen) correspondiente.

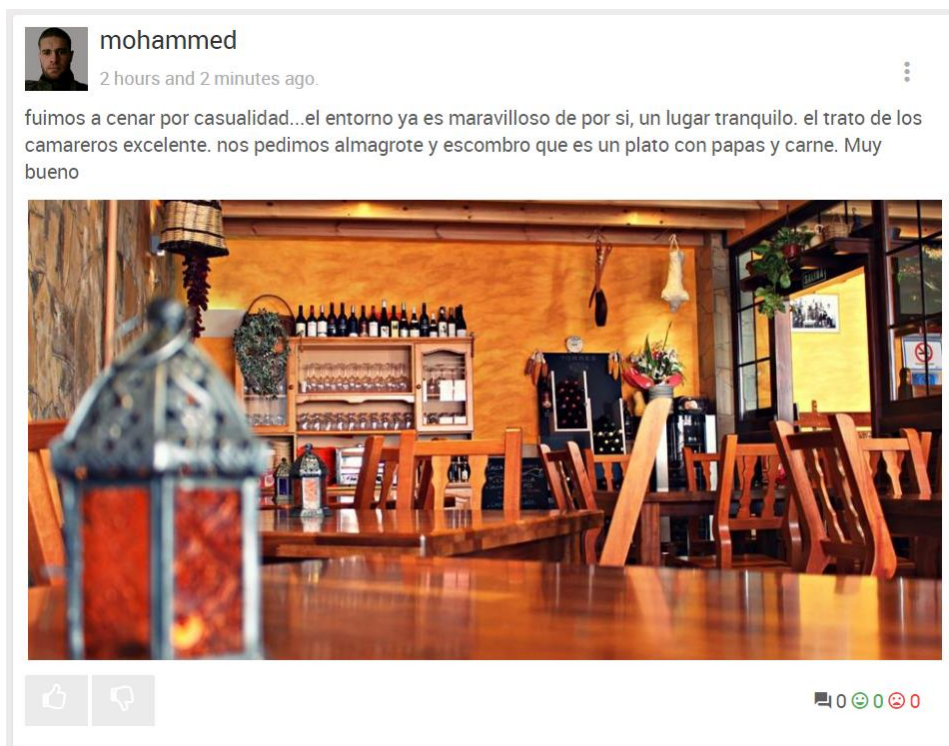


Figura 21: Post con adjunto publicado

## 5.4 Buscar una Experiencia

El buscador de experiencias es el campo de texto situado en la parte superior del navegador del FeedBack, capaz de buscar en la base de datos. Un ejemplo es el que aparece en la Figura 22, donde buscamos post de experiencias relacionadas con *la Gomera*.

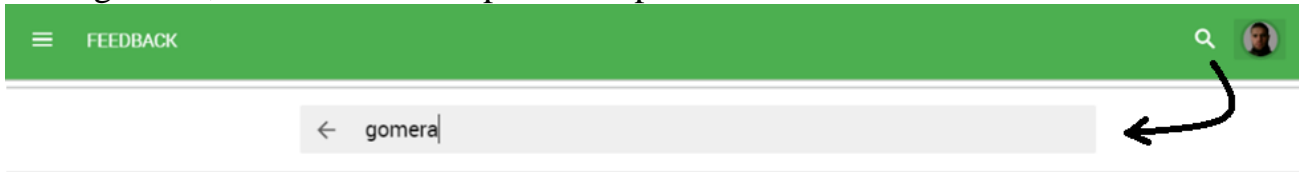


Figura 22: Buscar Experiencias relacionados con la Gomera

Mediante phpMyAdmin<sup>7</sup> vamos a asegurarnos de que en la base de datos halla los registros post relacionados con la gomera. En la Figura 23, aparecen dos registros relacionados con la Gomera.

+ Opciones		ID	user_id	status_title	status_content	status_time
<input type="checkbox"/>	Editar	96	38			2016-07-24 01:57:50
<input type="checkbox"/>	Editar	102	39	La Guardia Urbana de Barcelona está tratando de l...		2016-07-25 23:42:16
<input type="checkbox"/>	Editar	103	39	La <b>Gomera</b> es una de las siete islas principales de...		2016-07-25 23:43:29
<input type="checkbox"/>	Editar	105	30	Encuentra restaurantes en La <b>Gomera</b> donde comer en...		2016-07-25 23:59:07

Figura 23: Comprobación del número de los registros en la base de datos

Efectivamente, el buscador encuentra estos dos post y los visualiza en el muro del usuario como muestra en la Figura 24.

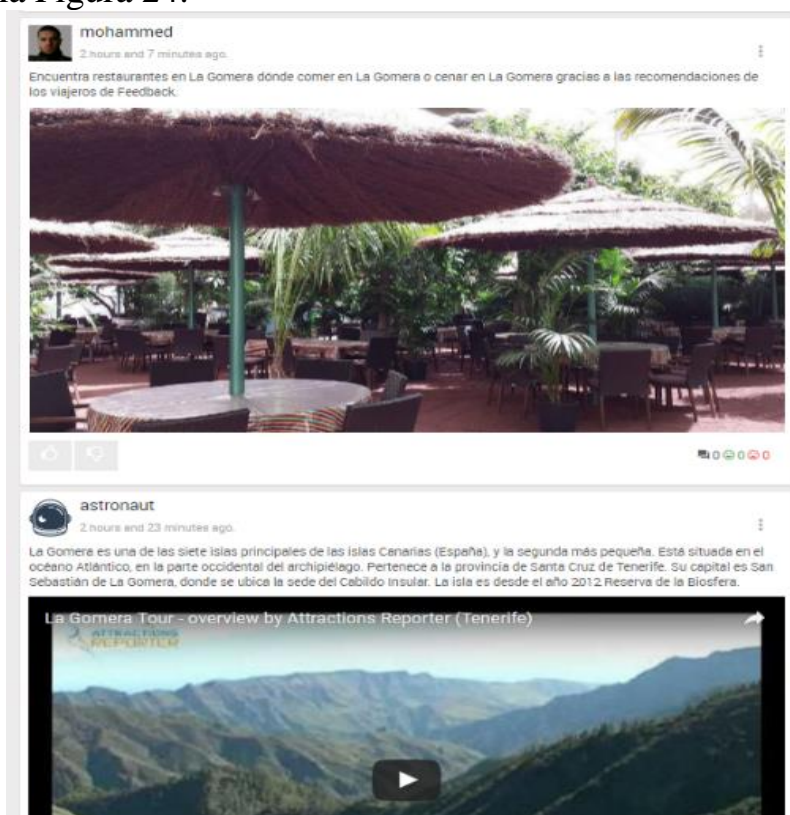


Figura 24: Resultado de búsqueda de post en Feedback

<sup>7</sup> Una aplicación web que crea y maneja bases de datos MySQL.

## 5.5 Comentar una Experiencia

Un post publicado puede ser recomendado o criticado a través de los comentarios positivos y negativos. De forma que el (Agree/disagree) es argumentado con una opinión en el comentario.

### 5.5.1 Feedbacks (Recomendar y Criticar)

Para recomendar una experiencia. Habrá que realizar un comentario positivo pulsando en el botón Agree(verde), este a su vez despliega una entrada de texto para argumentar la recomendación. Lo mismo sería para criticar, habrá que realizar un comentario negativo pulsando en este caso el botón disAgree (rojo), en la entrada de texto habrá que argumentar la crítica.

En la Figura 25, aparecen dos ejemplos de feedback en los dos casos.

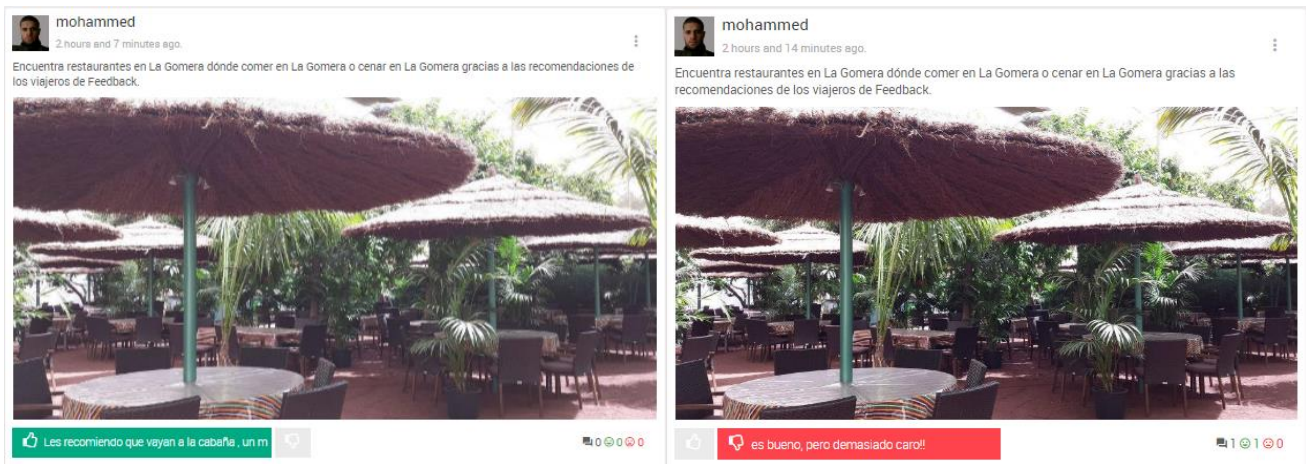


Figura 25:Tipos de feedback (Agree/Disagree)

### 5.5.2 Comentarios y sub-comentarios

Los comentarios pueden ser tratados a su vez como post en el sentido de que pueden ser criticados o recomendados de la misma forma que los post. La Figura 26 muestra la jerarquía de comentarios y sub-comentarios de un post.

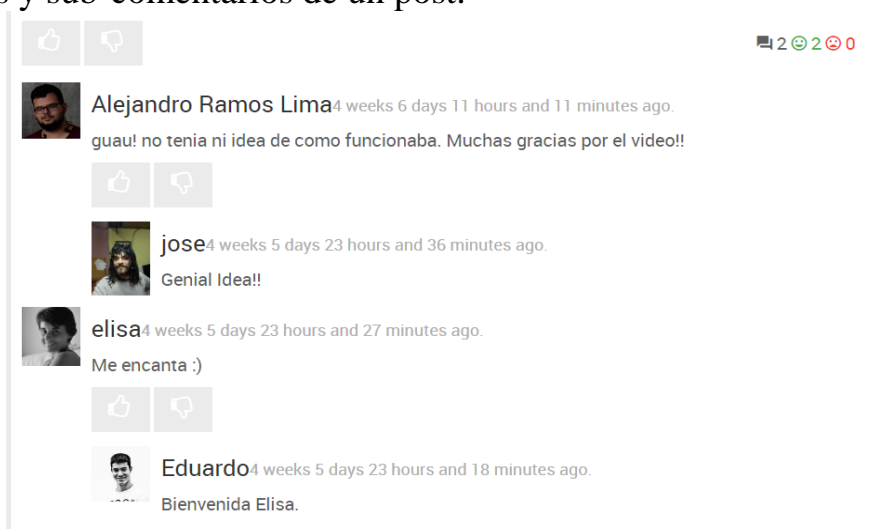


Figura 26: Los comentarios y sub-comentarios de un post

### 5.5.3 Edición de una experiencia

Una vez creada una experiencia tenemos la posibilidad de editarla cambiando su contenido o eliminando la experiencia por completo.

A continuación, en Figura 27 la muestra el menú donde editar/borrar un post.

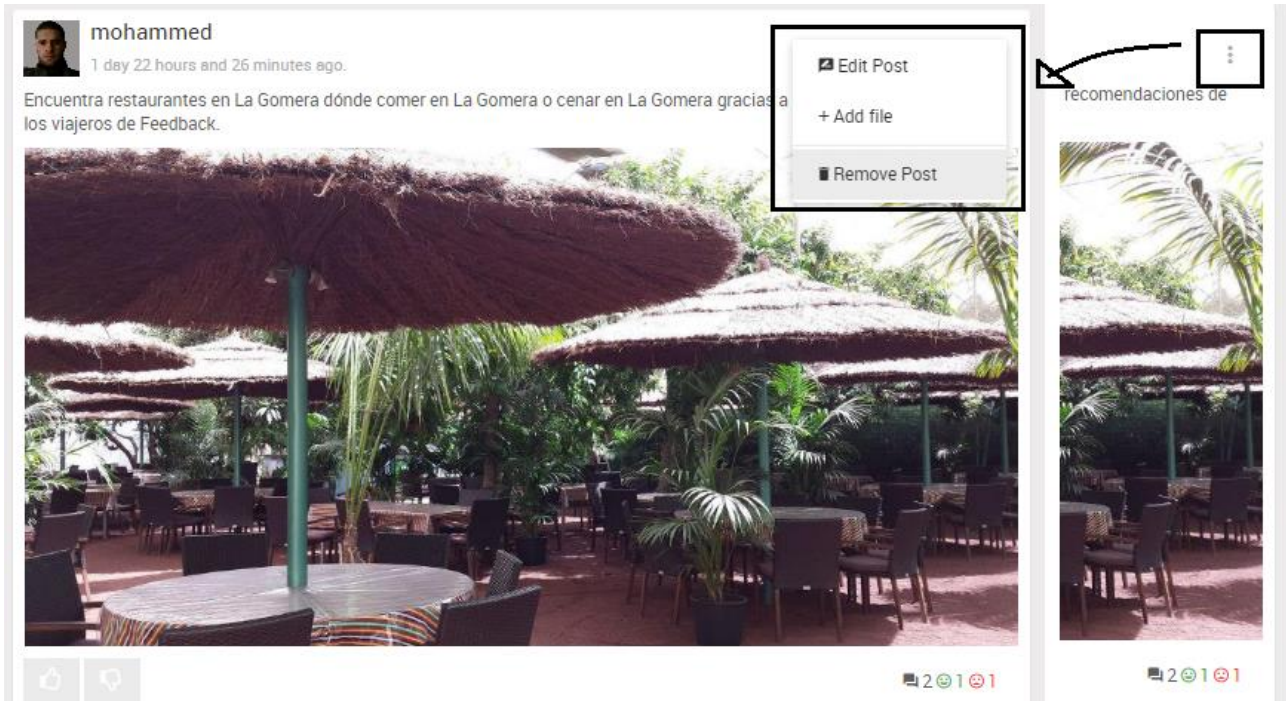


Figura 27: Acciones de edición de un post

### 5.5.4 Perfil del usuario

El perfil es la página personal del usuario, que además de visualizan los post de experiencia personales, un usuario puede consultar los datos personales en la pestaña "Information" y editarlos si fuese necesario en la pestaña "Edit profile", en la Figura 28 se puede ver el resultado.

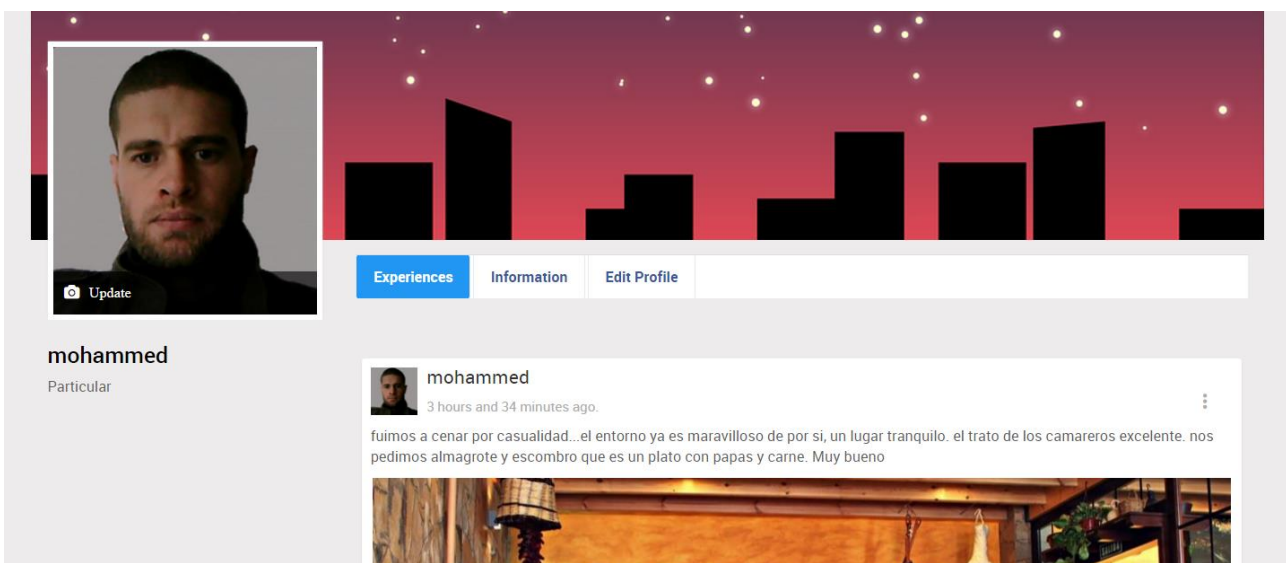


Figura 28: Página personal (perfil) del usuario

### 5.5.5 Visitar el perfil de otro usuario

Cuando visitamos la página personal de otro usuario observamos en la Figura 29 a diferencia de la Figura 28, solo podemos ver una pestaña “Experiences” y no tenemos acceso al resto de funcionalidad como podemos ver en la Figura 29. Además, en esta figura aparece en perfil visitado en dos modalidades, antes y después de seleccionarlo como favorito.

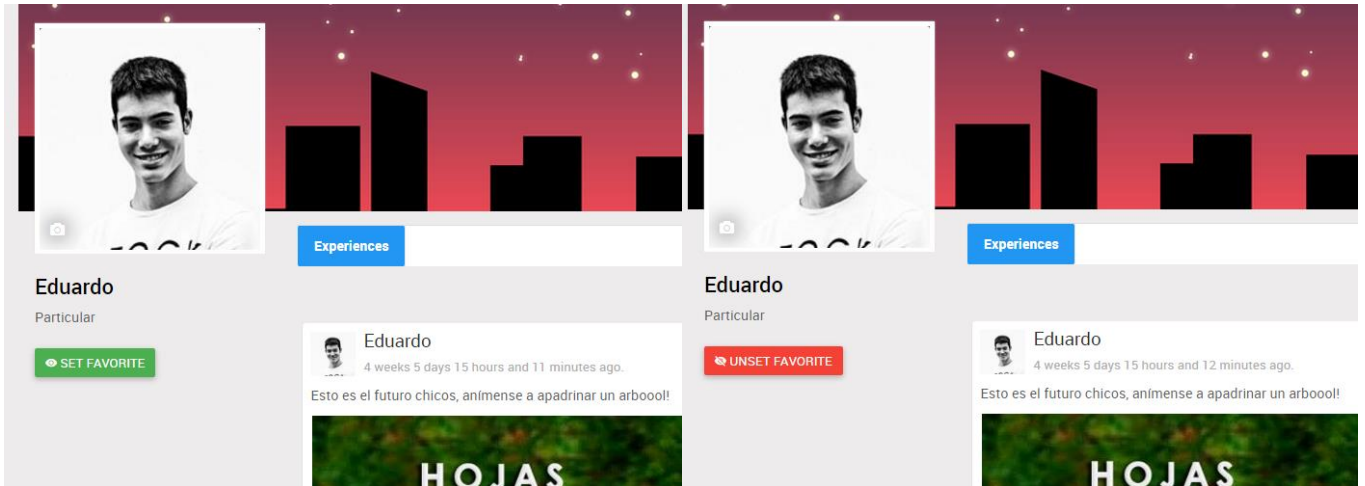


Figura 29: Página personal (perfil) de otro usuario

Por último, podemos ver en la Figura 30 que el perfil visitado se ha añadido a la lista de mis favoritos, después de seleccionarlo como favorito. Podemos acceder a los favoritos desde el menú izquierda como se ve en la misma figura.

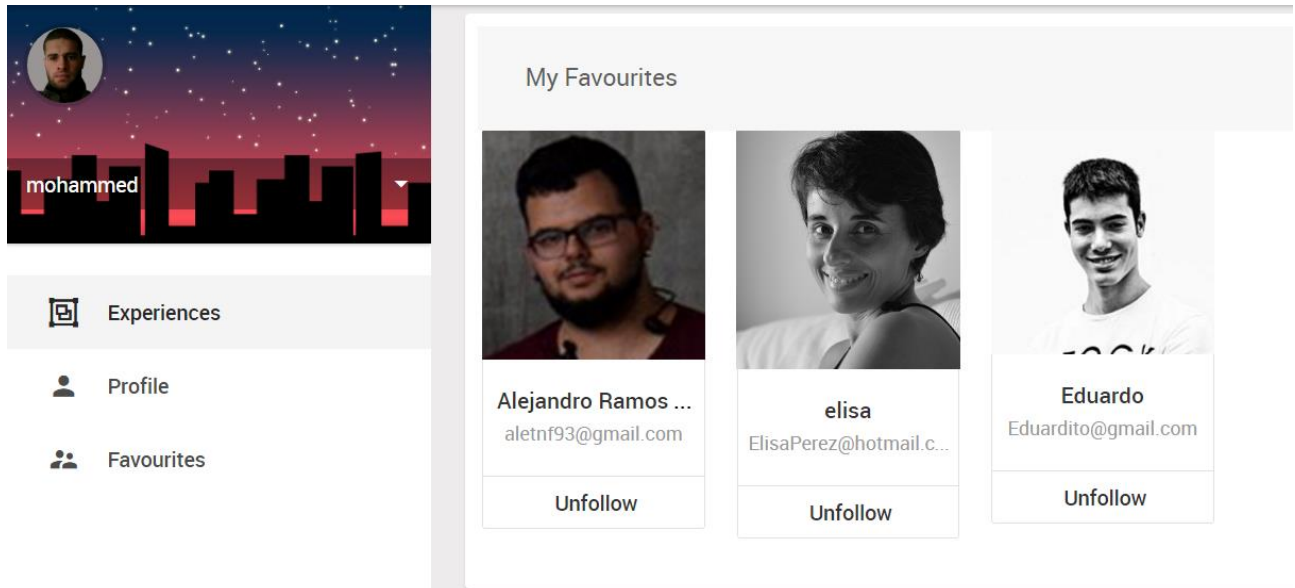


Figura 30: Mis favoritos



# Capítulo 6.

## Resumen y conclusiones

Este trabajo ha consistido en la realización de una aplicación web para una red social, FeedBack, dirigida a las opiniones y experiencias, que nos permita ayudar a gestionar de la mejor manera nuestras propias experiencias y aprovechar las experiencias de los demás ahorrándonos así, tiempo, dinero además de evitar pasar por malas experiencias.

El proceso seguido para el desarrollo de FeedBack ha sido en primer lugar realizar un análisis de los requisitos, donde se han identificado las funcionalidades y objetivos del sistema. Después se ha realizado un diseño de la aplicación, empleando un diseño Responsive-Design, tras lo cual se ha llevado a cabo la implementación de la aplicación, donde se han estudiado las distintas soluciones técnicas adaptables al diseño previo y donde también se han construido todos los componentes del sistema para su despliegue y utilización.

Para afrontar el diseño del sistema, primero se ha descrito la arquitectura de software, que está basada en el patrón de diseño MVC. Para ello, se han identificado los distintos componentes que conforman el sistema, constituyendo el modelo de datos, el controlador para el acceso al modelo y el procesamiento de datos y la vista para interactuar el usuario final.

Las tecnologías utilizadas han sido SGBD relacional en MySQL para el modelo de datos, PHP en el controlador de acceso a la base de datos y la generación de vistas dinámicas y HTML, CSS, Javascript con sus Framework para la presentación de vistas del lado del usuario. Para la comunicación entre la vista y el controlador se ha utilizado el protocolo HTTP.

Los usuarios son capaces de interactuar con las publicaciones, propias y de otros usuarios y realizar todas las acciones detalladas en los requerimientos. La estructura de red social también está reflejada en la aplicación, donde los usuarios pueden seguir y dejar de seguir a otros y visualizar su contenido publicado.

Este trabajo ha permitido crear una aplicación web escalable<sup>8</sup> y lista para su puesta en producción en Internet. Además, ha sido una oportunidad para reforzar el aprendizaje de ingeniería adquirido a lo largo de la carrera y ponerlos a la práctica, así como descubrir los beneficios aportados por la programación web y sobre todo con el lenguaje PHP. La fase de diseño ha sido la más desafiante e importante del proyecto, ya que ha requerido analizar con detalle las distintas soluciones posibles. Las tareas más laboriosas han sido las relacionadas con la fase de implementación, en particular validar el correcto funcionamiento de funciones como la de búsquedas de Experiencias para poder proveer un filtrado completo y consistente.

---

<sup>8</sup> La escalabilidad es la capacidad de mejorar recursos para ofrecer una mejora en la capacidad de servicio, para poder adaptarse al aumento de peticiones o el incremento de datos, se pueden diferenciar dos soluciones: escalabilidad vertical aumentando la potencia de los recursos y la escalabilidad horizontal que consiste en aumentar el número de los recursos, aunque no necesariamente potentes.

# Summary and Conclusions

This work has involved the realization of a web application for the social network “FeedBack”, dedicated to opinions and experiences users, allowing us to help manage in the best way our own experiences and benefit from the experiences of others, saving us time, money, in addition, to avoid going through bad experiences. The process followed for the development of “FeedBack” has been first of all an analysis of the requirements, where the functions and objectives of the system have been identified. After that, the application design was done using Responsive-Design. Then, the application implementation has been conducted, where we have studied the various solutions for the design and where also all system components have been built, allowing its deployment and use.

To deal with the design of the system, first, the software architecture, based on the MVC design pattern, has been described. For this purpose, we identified the different components of the system, constituting the data model, the driver for the access to the model and data processing and view to interact by the end user.

Technologies used have been MySQL as relational DBMS for the data model, PHP on the controller access to the database and the generation of dynamic views and HTML, CSS, Javascript with its Framework for the presentation of views of the user side. For communication between the view and the controller the HTTP protocol has been used.

Users are able to interact with their own publications and publications from others, and perform all the actions detailed in the requirements. The social network structure is also reflected in the application, where users can follow and unfollow others and view your published content.

This work allowed to create a web application that is scalable and ready for deployment. In addition, it has been an opportunity to reinforce the learning acquired over the engineering degree and put them into practice, as well as discover the benefits provided by the web programming and especially with the PHP language. The design phase has been the most challenging and important, since it has required a detailed analysis of the different possible solutions. The most laborious tasks have been those related to the implementation phase, in particular to validate correct operation of functions as the experiences searcher in order to provide a complete and consistent filtering.

# Apéndice A.

## Competencia

Nombre Y Url	Objetivo/Temática	Descripción breve	Aspecto de Diferencia o Similitud
<b>CIAO</b> <a href="https://ciao.es">https://ciao.es</a>	Electrodomésticos y ocio	Un Portal Europeo del Consumidor y ofertas para ayudarte a elegir la mejor opción entre millones, Ciao paga por dar tu opinión sobre productos de todo tipo.	<b>Ciao</b> afirma que 1,3 millones de usuarios escribieron más de 7 millones opiniones sobre 1,4 millones productos, pero a diferencia de FeedBack es una plataforma especializada solo en objetos y productos comerciales.
<b>CRITIZEN</b> <a href="https://critize.com">https://critize.com</a>	Críticas	Critizen nace como una curiosa especie de red social donde puedes criticar y reclamar a empresas y marcas.	<b>Critizen</b> tiene una credibilidad bastante convincente por la transparencia de las opiniones de los usuarios y esto es parecido a lo que queremos conseguir, sin embargo, Critizen solo se centra en críticas a empresas o servicios y no todo tipo de opiniones.
<b>OPÍNAME</b> <a href="https://opiname.es">https://opiname.es</a>	Empresas y servicios	Opíname es un comparador “social” de opiniones sobre empresas y servicios en toda España.	<b>Opíname</b> es un portal para opinar empresas y negocios categorizados por ubicación y es bastante organizado además solo opera en España, Esta claro que es mucho más limitado de lo que pretendemos crear con FeedBack.
<b>ALATEST</b> <a href="https://alatest.es">https://alatest.es</a>	Electrodomésticos	Un buscador independiente de evaluaciones y pruebas de productos, con una calificación objetiva de la calidad de los mismos.	<b>Alatest</b> es un calculador de puntuaciones de los usuarios, así que el mismo portal compara y decide por ti cual es el mejor producto en su categoría.
<b>MENÉAME</b> <a href="https://meneame.net">https://meneame.net</a>	Noticias	Portal donde los usuarios envían y votan noticias, para promocionarlas en portada.	<b>Menéame</b> tiene comentarios reales de forma parecida a la que pretendemos desarrollar para FeedBack, aunque en este caso está especializado en noticias.
<b>TRIPADVISOR</b> <a href="https://tripadvisor.com">https://tripadvisor.com</a>	Turismo	Una web útil donde se puede consultar consejos de millones de usuarios, para quien busca viajes restaurante o hoteles cuando no conoces el lugar.	<b>TripAdvisor</b> se puede sacar de ella recomendaciones bastante sensatas sobre restaurantes, hoteles, así como advertencias sobre lugares a los que es mejor no acercarse, ambas basadas en comentarios de personas que han estado en ellos.

Tabla 27: Competencia

# Apéndice B.

## Protección de Datos Personales

Para la realización de este proyecto de forma legal es necesario durante el desarrollo y el diseño tener en cuenta las siguientes normativas.

### **B.1. Protección de los derechos al honor, propia imagen, intimidad y privacidad**

*Ley Orgánica 1/1982, de 5 de mayo, de protección civil del derecho al honor, a la intimidad personal y familiar y a la propia imagen.*

Conforme al artículo dieciocho, uno, de la Constitución, los derechos al honor, a la intimidad personal y familiar y a la propia imagen tienen el rango de fundamentales y hasta tal punto aparecen realzados en el texto constitucional que el artículo veinte, cuatro, dispone que el respeto de tales derechos constituya un límite al ejercicio de las libertades de expresión que el propio precepto reconoce y protege con el mismo carácter de fundamentales

### **B.2. Protección de Datos de Carácter Personal**

*LEY ORGÁNICA 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal.*

La presente Ley Orgánica tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas y especialmente de su honor e intimidad personal y familiar.

### **B.3. Protección de menores e incapaces**

*Real Decreto Legislativo 1/1996, de 12 de abril, de la Ley de Propiedad Intelectual.*

por el que se aprueba el texto refundido de la Ley de Propiedad Intelectual, regularizando, aclarando y armonizando las disposiciones legales vigentes sobre la materia.

### **B.4. Aspectos relativos a la seguridad de la información**

*Ley de Propiedad Intelectual N° 11.723*

Protege los derechos de autor de obras científicas, literarias y artísticas, incluyendo los programas de computación fuente y objeto, la compilación de datos y otros materiales.

Ley de Marcas N° 22.362: Protege la propiedad de una marca y la exclusividad de su uso.

Ley de Patentes e Invenciones y Modelos de Utilidad N° 24.481:

Protege el derecho del titular de la patente de invención para impedir que terceros utilicen su producto o procedimiento.

## **B.5. Ley de Servicios de la Sociedad de la información y el Comercio electrónico.**

*LEY 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico.*

Es objeto de la presente Ley la regulación del régimen jurídico de los servicios de la sociedad de la información y de la contratación por vía electrónica, en lo referente a las obligaciones de los prestadores de servicios incluidos los que actúan como intermediarios en la transmisión de contenidos por las redes de telecomunicaciones

## **B.6. Ley General de Publicidad**

*Ley 34/1988, de 11 de noviembre, General de Publicidad.*

La publicidad se regirá por esta Ley, por la Ley de Competencia Desleal y por las normas especiales que regulen determinadas actividades publicitarias.

A los efectos de esta Ley, se entenderá por publicidad toda forma de comunicación realizada por una persona física o jurídica, pública o privada, en el ejercicio de una actividad comercial, industrial, artesanal o profesional con el fin de promover de forma directa o indirecta la contratación de bienes muebles o inmuebles, servicios, derechos y obligaciones.

# Apéndice C.

## Glosario de términos

**Usuario:** Persona o entidad que utiliza y forma parte de una red social. El usuario puede acceder a ella con sus propias credenciales. Estos son únicos para cada usuario.

**Perfil:** Datos personales y rasgos propios que caracterizan a un usuario dentro de una red social, como su nombre, fotografía, lugar de residencia o preferencias. El perfil representa su identidad virtual.

**Post:** Entrada, mensaje o publicación en una red social que puede consistir en un texto, opinión, comentario, enlace o archivo compartido.

**Muro:** Espacio del usuario de una red social que comparte con el resto de sus contactos, donde estos pueden publicar sus comentarios u opiniones.

**Cuenta:** Conjunto de información que permite el acceso a una red social a través de la identificación de usuario. La cuenta se crea con un nombre de usuario y contraseña, en algunos casos, a través de una cuenta de correo electrónico.

**Seguidor:** Llamado favoriteer en la terminología de Twitter. Usuario de esta red social que se suscribe a los mensajes o publicaciones (tweets) de otros usuarios, bien por admiración, como en el caso de los seguidores de deportistas o cantantes; por simpatizar con sus ideas; por mantenerse informado de sus actividades en Twitter.

**Access:** La capacidad de ver lo que se está intentando ver, por ejemplo: “puede acceder a la foto de un amigo, pero no a su perfil”.

**Autenticación:** Es el acto de establecimiento o confirmación de algo o alguien como auténtico. A menudo consiste en verificar la identidad de una persona, confirmar que es quien dice ser.

**Autorización:** Es una parte del sistema que protege sus propios recursos, permitiendo que sólo sean usados por aquellos usuarios a los que se les ha concedido autorización para ello, o que puedan realizar ciertas acciones sobre los mismos recursos.

**Avatar:** Representación gráfica mediante un dibujo, fotografía o figura, generalmente humana, que se asocia a un usuario para su identificación en la Red.

**Feedback (retroalimentación):** Desde el punto de vista social, es el proceso de compartir observaciones, preocupaciones y sugerencias, con la intención de recabar información, a nivel individual o colectivo, para intentar mejorar el funcionamiento de una organización o de cualquier grupo formado por seres humanos.

# Apéndice D.

## Presupuesto

<b>PRESUPUESTO DETALLADO DEL PROYECTO</b>				
	Unidad de Medida	Coste /Unidad	Nº unidades	Coste Estimado
<b>1. PERSONAL PROPIO EN EL PROYECTO</b>				
Salario	Coste hora	30€	200 Hs	6000
TOTAL 1				6000
<b>2. HARDWARE</b>				
Equipo IMAC (Intel Core i7, 3,3 GHz)	Amortización	600	1 Pcs	600
TOTAL 2				600
<b>3. SOFTWARE</b>				
Herramientas de desarrollo (Apache, php)		0	2	0
Herramientas de Diseño (visual paradigm licencia academica)		0	1	0
Bases de Datos (MariaDB)		0	1	0
TOTAL 3				0
<b>4. MATERIALES Y SUMINISTROS</b>				
Consumibles informáticos	Estimado			40
Material de oficina	Estimado			10
Suministros generales (luz, teléfonos,...)	Estimado			20
TOTAL 4				70
<b>PRESUPUESTO TOTAL</b>				<b>6670</b>

Tabla 28: Presupuesto

## Bibliografía

- [1] [1] Gómez Marcos, Mayte, "queaprendemoshoy," 3 Abril 2014. [Online]. Available: <http://queaprendemoshoy.com/la-nueva-economia-compartida-como-funciona-la-p2p/>. [Accessed 15 Agosto 2016].
- [2] Michael Abernethy, «ibm,» 14 Junio 2011. [En línea]. Available: <https://www.ibm.com/developerworks/ssa/opensource/library/os-nodejs/>. [Último acceso: 6 Junio 2016].
- [3] Y. m. o. Mehdi Achour, «php.net,» 04 Septiembre 2016. [En línea]. Available: <http://php.net/manual/es/index.php>. [Último acceso: 28 Agosto 2016].
- [4] A. Friends, «www.apachefriends.org,» 23 Diciembre 2016. [En línea]. Available: [https://www.apachefriends.org/blog/new\\_xampp\\_20160831.html](https://www.apachefriends.org/blog/new_xampp_20160831.html). [Último acceso: 05 08 2016].
- [5] Bartolomé Sintés, Marco, «mclibre,» 2015 septiembre 16. [En línea]. Available: [http://www.mclibre.org/consultar/php/otros/in\\_php\\_xampp.html](http://www.mclibre.org/consultar/php/otros/in_php_xampp.html). [Último acceso: 01 03 2016].
- [6] M. Medina, «interdominios,» 25 octubre 2013. [En línea]. Available: <http://blog.interdominios.com/7-grandes-diferencias-entre-adaptive-web-design-y-responsive-web-design/>. [Último acceso: 01 06 2016].
- [7] SEDIC, «sedic,» 5 Mayo 1999. [En línea]. Available: <http://www.sedic.es/autoformacion/accesibilidad/programa.html>. [Último acceso: 7 Julio 2016].
- [8] L. WEB, «librosweb,» 2016. [En línea]. Available: [http://librosweb.es/libro/jobeeet\\_1\\_4/capitulo\\_4/la\\_arquitectura\\_mvc.html](http://librosweb.es/libro/jobeeet_1_4/capitulo_4/la_arquitectura_mvc.html). [Último acceso: 20 07 2016].
- [9] Oracle Corporation and/or its affiliates, «mysql,» 2016. [En línea]. Available: <http://dev.mysql.com/doc/refman/5.7/en/fulltext-search.html>. [Último acceso: 06 07 2016].
- [10] T. P. Group., «http://php.net,» 2016. [En línea]. Available: <http://php.net/manual/es/session.security.php>. [Último acceso: 10 08 2016].
- [11] Mayte Gómez, Marcos, «http://queaprendemoshoy.com,» 3 4 2014. [En línea]. Available: <http://queaprendemoshoy.com/la-nueva-economia-compartida-como-funciona-la-p2p/>. [Último acceso: 6 6 2016].



- [13] wikipedia, «wikipedia,» 4 Abril 2016. [En línea]. Available: [https://es.wikipedia.org/wiki/V8\\_\(motor\\_JavaScript\)](https://es.wikipedia.org/wiki/V8_(motor_JavaScript)). [Último acceso: 4 Mayo 2016].
- [14] Dr. Maria dle Carmen Gómez Fuentes, «cua.uam.mx,» Mayo 2011. [En línea]. Available: [http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas\\_Analisis\\_Requerimiento.pdf](http://www.cua.uam.mx/pdfs/conoce/libroselec/Notas_Analisis_Requerimiento.pdf). [Último acceso: 02 07 2016].