



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Diseño e implementación de una aplicación web para la gestión integrada de redes sociales

Trabajo Fin de Grado

Grado en Ingeniería Informática

Autor: Guardiola Mulet, Vicente

Tutor: Sáez Barona, Sergio

2016 - 2017

Resumen

Este proyecto se basa en el desarrollo de una aplicación web encargada de publicar contenido programado de forma automática en redes sociales. Para el desarrollo del proyecto, se han utilizado principalmente las tecnologías PHP, MySQL y JavaScript entre otras. El desarrollo de este proyecto contempla la capacidad de programar contenido para Twitter, Facebook y Tumblr y que este sea publicado en la fecha y hora deseada, así como la gestión de las publicaciones pendientes y las ya publicadas.

Palabras clave: desarrollo web, PHP, MySQL, Bootstrap, redes sociales.

Abstract

This project is based on the development of a web application with the functionality of automatically publishing programmed content in social media platforms. For its development, the technologies mainly used are PHP, MySQL and JavaScript among others. The development of this project comprises the capabilities to program content for Twitter, Facebook and Tumblr, for this to be published at a given date and time; as well as the management of pending and published posts.

Keywords: web development, PHP, MySQL, Bootstrap, social networks.

Tabla de contenidos

| | | |
|-----|--|----|
| 1. | Introducción..... | 6 |
| 1.1 | Objetivos | 6 |
| 1.2 | Estructura | 7 |
| 2. | Especificación de requisitos | 8 |
| 2.1 | Introducción | 8 |
| 2.2 | Descripción general | 9 |
| 2.3 | Requisitos específicos | 10 |
| 2.4 | Requisitos no funcionales | 13 |
| 3. | Análisis | 14 |
| 3.1 | Diagrama entidad-relación..... | 14 |
| 3.2 | Casos de uso..... | 15 |
| 4. | Diseño..... | 20 |
| 4.1 | Capa de presentación..... | 20 |
| 4.2 | Capa de persistencia | 24 |
| 4.3 | Capa lógica | 25 |
| 5. | Detalles de implementación..... | 27 |
| 5.1 | Tecnologías utilizadas..... | 27 |
| 5.2 | Aplicaciones utilizadas..... | 31 |
| 5.3 | Estructura de los ficheros y directorios | 33 |
| 6. | Pruebas..... | 39 |
| 6.1 | Conformidad con el estándar W3C..... | 39 |
| 6.2 | Visualización en distintas resoluciones | 42 |
| 6.3 | Pruebas de uso | 47 |
| 7. | Conclusiones | 57 |
| 7.1 | Conclusiones técnicas | 57 |
| 7.2 | Conclusiones personales..... | 57 |
| 8. | Bibliografía..... | 58 |

1. Introducción

Hoy en día las redes sociales se han convertido en una de las herramientas más importantes a la hora de darse a conocer como empresa o negocio. Gracias a ellas se consigue un acercamiento más personal hacia el cliente que con un anuncio de televisión o un cartel en una marquesina del metro.

Realizando las prácticas en Bolsapubli S.L, una empresa de fabricación y venta de bolsas personalizadas de todo tipo, me he percatado de la importancia de tener todas las redes sociales al día con contenido de calidad. La gran problemática es que no siempre se puede estar publicando contenido, ya que hay muchas más tareas por hacer en la empresa. Esto se soluciona programando contenido para que se publique automáticamente en las redes sociales. Para ello, es necesario gestionar una gran cantidad de cuentas, cosa que hace la tarea muy tediosa.

Para resolver estos problemas, he decidido implementar una aplicación web en la que se integren tres redes sociales muy conocidas. Gracias a ella se podrá programar contenido de forma más sencilla, sin necesidad de estar cambiando de red social o de cuenta de usuario continuamente.

1.1 Objetivos

Los objetivos principales de este trabajo se centran en el diseño e implementación de una aplicación web. Implementaremos un sistema que nos permita programar contenido para que se publique en la fecha deseada.

Nos centraremos en tres de las redes sociales más conocidas: Twitter, Facebook y Tumblr. Desarrollaremos un sistema que nos permita, mediante las distintas APIs, publicar en las tres redes sociales sin necesidad de cambiar de cuenta y web continuamente.

Además de poder programar contenido, también se podrá revisar el contenido ya publicado o el que está pendiente de publicar, así como eliminar las publicaciones o editarlas a nuestro gusto.

Como objetivos personales están el aprender a programar de forma fluida con PHP así como mejorar el conocimiento aprendido en el grado de MySQL.

1.2 Estructura

En esta memoria empezaremos exponiendo la especificación de requisitos software siguiendo el estándar del IEEE 830/1998. A continuación, seguiremos con la etapa de análisis donde veremos los distintos casos de uso y el diagrama de clases entidad-relación.

Continuaremos explicando la fase de diseño, donde diferenciaremos en tres capas el desarrollo de la aplicación web. Estas serán la de presentación, la de persistencia y por último, la capa de lógica.

En la parte de la capa de presentación veremos los distintos bocetos de la web, con los que podremos intuir su futuro aspecto y algunas de sus funcionalidades. Para explicar la capa de persistencia, nos serviremos del esquema lógico, además de comentar las distintas tablas así como cada uno de sus campos. Por último, se explicará la capa lógica explorando el flujo de trabajo que sigue la aplicación web mediante un diagrama.

Explicadas las fases de análisis y diseño, nos dispondremos a comentar los detalles más importantes de la implementación. Para ello nos ayudaremos del árbol de ficheros resultante e iremos comentando los que se encarguen de las funcionalidades principales. Además, expondremos las distintas tecnologías utilizadas para implementar la aplicación.

En el siguiente punto, pasaremos un test de validación de HTML y CSS para comprobar que la aplicación web cumple con los estándares del W3C. Seguidamente probaremos la web en distintas resoluciones para comprobar su aspecto en ordenadores, dispositivos móviles y tabletas.

Siguiendo con la fase de pruebas, exploraremos las distintas funcionalidades de la aplicación, ayudándonos de capturas de pantalla y explicando paso a paso todas las acciones que debemos realizar.

Por último, dividiremos el apartado de conclusiones en dos partes, una para las conclusiones académicas del proyecto y otra para las personales. Además, al final de este documento, incluiremos la bibliografía que hemos utilizado para llevar a cabo el desarrollo del proyecto y un glosario con términos que han aparecido a lo largo de la memoria.

2. Especificación de requisitos

En este apartado del documento, expondremos la especificación de requisitos software de la aplicación web que vamos a implementar. Empezaremos con una pequeña introducción en la que hablaremos del propósito y el contexto del proyecto. Para continuar, hablaremos de la perspectiva del producto, las funciones principales y las características de los usuarios, así como las restricciones y suposiciones más importantes.

Seguiremos comentando los requisitos funcionales mediante unas tablas. Posteriormente, comentaremos los requisitos no funcionales como la seguridad, fiabilidad y mantenibilidad entre otros.

2.1 Introducción

Como hemos señalado anteriormente, en esta pequeña introducción hablaremos del propósito de este apartado del documento y del alcance del sistema.

Para la especificación de requisitos, utilizaremos el estándar IEEE 830/1998 en el que se describen todos los pasos a seguir para generar un buen documento de requisitos. Como se trata de un proyecto de pequeña envergadura y con fines académicos, algunos de los puntos que marca el estándar no aparecerán en el documento ya que no se pueden aplicar.

2.1.1 Propósito

El propósito principal de este apartado es definir de una forma clara todas las funcionalidades que el sistema va a cumplir, así como las principales restricciones de nuestra aplicación.

2.1.2 Ámbito del sistema

En este proyecto vamos a implementar una aplicación web a la que llamaremos Social Manager BP. Esta, hará la tarea de publicar contenido en redes sociales más sencilla, gracias a la posibilidad de programar contenido y que se publique posteriormente de forma automática.

Los usuarios del sistema, podrán programar publicaciones de Facebook, Twitter y Tumblr a la vez, sin la necesidad de cambiar de página. Además, podrán visualizar las publicaciones que están pendientes de publicar y las que ya han sido publicadas, pudiendo eliminar las publicadas y pendientes de publicar o modificar las que no han sido publicadas.

Los principales beneficios de la aplicación serán la posibilidad de tener todas las cuentas de las tres redes sociales en una misma web. Ya no será necesario acceder a cada una de las redes sociales para publicar contenido.

Además, gracias a la posibilidad de programar el contenido, no será necesario estar continuamente pendiente de las redes, ya que se publicará de forma automática después de haberlo programado.

Cabe señalar que la aplicación no tendrá la capacidad de generar contenido automáticamente, solo publicará lo que el usuario introduzca en los respectivos campos de los formularios, es decir, la aplicación publicará el contenido que el usuario genere.

2.1.3 Definiciones, Acrónimos y Abreviaturas

- API: Interfaz de Programación de Aplicaciones es un conjunto de métodos que ofrece una biblioteca para ser utilizados como capa de abstracción por otra aplicación.

2.1.4 Referencias

- IEEE Recommended Practice for Software Requirements Specification. ANSI/IEEE 830/1998

2.1.5 Visión general del documento

Este documento está formado por tres secciones. En la sección actual de introducción, proporcionamos una visión general del documento de especificación de requisitos. En la sección número dos, describimos de forma general el sistema que vamos a implementar, comentando las principales funciones, restricciones y dependencias. Por último, en la tercera sección, definimos detalladamente los requisitos que el sistema debe cumplir.

2.2 Descripción general

2.2.1 Perspectiva del producto

En este proyecto vamos a desarrollar una aplicación web que permita gestionar la publicación automática de contenido en distintas redes sociales. Esta aplicación va a depender de las APIs de las distintas redes sociales, en este caso de Twitter, Facebook y Tumblr.

2.2.2 Funciones del producto

A continuación vamos a describir las principales funciones que deberá realizar el futuro sistema que deseamos implementar:

- Programación de contenido para Facebook, Twitter y Tumblr: El usuario podrá programar contenido en las distintas redes sociales para que sea publicado posteriormente de forma automática por el sistema.
- Modificación del contenido programado: En caso de que así lo desee, el usuario podrá modificar las publicaciones que estén pendientes de ser publicadas.
- Borrado de contenidos: El usuario podrá borrar los contenidos que estén pendientes de publicar y los que ya hayan sido publicados en las redes sociales.
- Administración de cuentas: El usuario podrá ver las cuentas de las redes sociales que tiene agregadas al sistema. Podrá añadir nuevas o borrar cuentas existentes cuando lo desee.
- Búsqueda de publicaciones: El usuario podrá buscar entre las publicaciones publicadas o pendientes de publicar, así podrá ver los contenidos que van a ser publicados o los que ya han sido publicados sin necesidad de entrar en las distintas redes sociales.



2.2.3 Características de los usuarios

Ya que esta aplicación web está pensada para la gestión de redes sociales de una empresa, solo hay un tipo de usuario al que podríamos llamar usuario registrado, sus funciones serán las de programar contenido, gestionarlo una vez programado o publicarlo y añadir o eliminar cuentas de las distintas redes sociales.

Ningún otro usuario que no esté registrado podrá realizar las acciones descritas anteriormente. Los usuarios no registrados podrán registrarse en caso de que dispongan del código de la empresa necesario para ello.

2.2.4 Restricciones

A continuación vamos a enumerar las restricciones más importantes que nuestro sistema deberá respetar:

- El sistema deberá cumplir los requisitos sujetos a la utilización de las APIs de Facebook, Twitter y Tumblr.
- La aplicación web deberá funcionar de forma fluida en un alojamiento que tenga una capacidad de 10GB de disco duro, 1024MB de RAM y un solo núcleo de CPU.
- La aplicación deberá funcionar perfectamente en cualquier sistema operativo, siempre y cuando se acceda desde un navegador Google Chrome.
- La aplicación web deberá mostrarse perfectamente en monitores con una resolución de 1920x1080 y de 1366x768.

2.2.5 Suposiciones y dependencias

La aplicación está diseñada para funcionar con las distintas APIs de las redes sociales, en caso de que alguna API dejara de funcionar o se pusiera en mantenimiento durante un periodo de tiempo, la web podría experimentar problemas a la hora de programar y gestionar el contenido de la red social cuya API ha sido afectada.

2.3 Requisitos específicos

En este apartado, vamos a exponer los requisitos principales del sistema que se va a desarrollar. Vamos a comentar los requisitos funcionales, los requisitos de rendimiento y las interfaces externas, entre otros.

2.3.1 Requisitos funcionales

En este apartado vamos a describir de forma más detallada los requisitos funcionales principales que deberá cumplir nuestra aplicación web.

| | |
|---------------------------|---|
| Número del requisito | 1 |
| Nombre del requisito | Añadir cuenta de red social |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Esencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/Opcional |
| Descripción del requisito | Se podrán añadir cuentas de Twitter, Facebook y Tumblr para después poder utilizarlas para programar contenido. |

| | |
|---------------------------|---|
| Número del requisito | 2 |
| Nombre del requisito | Eliminar cuenta de red social |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrán eliminar cuentas de Twitter, Facebook y Tumblr que ya no queramos utilizar. |

| | |
|---------------------------|---|
| Número del requisito | 3 |
| Nombre del requisito | Programar publicación para Twitter |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá programar contenido exclusivamente para Twitter sin necesidad de programarlo en Facebook y Tumblr. |

| | |
|---------------------------|---|
| Número del requisito | 4 |
| Nombre del requisito | Programar publicación para Facebook |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá programar contenido exclusivamente para Facebook sin necesidad de programarlo en Twitter y Tumblr. |

| | |
|---------------------------|---|
| Número del requisito | 5 |
| Nombre del requisito | Programar publicación para Tumblr |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá programar contenido exclusivamente para Tumblr sin necesidad de programarlo en Twitter y Facebook. |

| | |
|---------------------------|---|
| Número del requisito | 6 |
| Nombre del requisito | Programar publicación para Twitter, Facebook y Tumblr |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá programar contenido para las tres redes sociales a la vez en la misma ventana. Para ello el usuario deberá rellenar los campos necesarios de cada publicación ya que cada red social requiere unos campos concretos. |

| | |
|---------------------------|---|
| Número del requisito | 7 |
| Nombre del requisito | Programar publicación con imagen |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá programar contenido que incluya una imagen para las distintas redes sociales, ya sea por separado o todas al mismo tiempo. |



| | |
|---------------------------|--|
| Número del requisito | 8 |
| Nombre del requisito | Programar publicación en varias cuentas de la misma red social |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá programar el mismo contenido en varias cuentas de una misma red social sin necesidad de crear una por una cada publicación. |

| | |
|---------------------------|---|
| Número del requisito | 9 |
| Nombre del requisito | Modificar publicación programada |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input type="checkbox"/> Alta/Eencial <input checked="" type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá modificar el contenido de las publicaciones programadas así como la fecha de publicación. |

| | |
|---------------------------|---|
| Número del requisito | 10 |
| Nombre del requisito | Eliminar publicación programada |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrá cancelar la publicación de contenido que esté pendiente de publicar. |

| | |
|---------------------------|---|
| Número del requisito | 11 |
| Nombre del requisito | Eliminar publicación realizada |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrán eliminar las publicaciones que ya hayan sido publicadas en la red social si lo estimamos oportuno. |

| | |
|---------------------------|---|
| Número del requisito | 12 |
| Nombre del requisito | Listar y buscar publicaciones pendientes |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrán listar las publicaciones pendientes de publicar, además se podrá buscar una publicación concreta escribiendo palabras que puedan estar en el contenido de la publicación. |

| | |
|---------------------------|---|
| Número del requisito | 13 |
| Nombre del requisito | Listar y buscar publicaciones realizadas |
| Tipo | <input checked="" type="checkbox"/> Requisito <input type="checkbox"/> Restricción |
| Fuente del requisito | Aplicación web de gestión integrada de redes sociales |
| Prioridad del requisito | <input checked="" type="checkbox"/> Alta/Eencial <input type="checkbox"/> Media/Deseado <input type="checkbox"/> Baja/ Opcional |
| Descripción del requisito | Se podrán listar las publicaciones que ya hayan sido publicadas, además se podrá buscar una publicación concreta escribiendo palabras que puedan estar en el contenido de la publicación. |

2.3.2 Requisitos de interfaces externas

- Interfaces de Usuario: La interfaz deberá ser fácil de utilizar para un usuario con unos conocimientos medios de informática. Todas las acciones se podrán realizar mediante el teclado y el ratón sin la necesidad de otros periféricos.
- Interfaces Hardware: En este proyecto intervendrán las máquinas de los servicios de Facebook, Twitter y Tumblr, además de la máquina encargada del alojamiento de nuestra aplicación y la del cliente.
- Interfaces Software: El usuario accederá a la aplicación web mediante el navegador de escritorio Google Chrome.

2.3.3 Requisitos de rendimiento

El sistema desarrollado deberá servir peticiones a un máximo de 3 usuarios a la vez en menos de 15 segundos. La aplicación no deberá soportar una gran cantidad de usuarios, ya que está diseñada con fines académicos.

2.4 Requisitos no funcionales

A continuación, vamos a exponer los requisitos no funcionales más importantes que deberá cumplir nuestra aplicación web.

2.4.1 Fiabilidad

No se permitirá ningún tipo de error a la hora de publicar el contenido de forma automática en las redes sociales.

2.4.2 Disponibilidad

La aplicación tendrá que ser accesible un mínimo de 23 horas al día, 7 días a la semana, hasta que se decida dar de baja el servicio.

2.4.3 Mantenibilidad

La aplicación web deberá ser fácil de mantener, el único mantenimiento necesario debería ser la creación de copias de seguridad de forma periódica.

2.4.4 Portabilidad

El proyecto tendrá que ser fácil de migrar de un servidor a otro, sin que sea necesario más de una persona y un par de días para completar el proceso.

2.4.5 Seguridad

Para acceder a todas las funcionalidades del sistema, el usuario deberá iniciar sesión con sus credenciales. Solo se podrán registrar nuevos usuarios que tengan un código proporcionado por la empresa.



3. Análisis

En este apartado vamos a exponer el diagrama entidad-relación de nuestra aplicación web. Explicaremos cada una de las entidades y posteriormente pasaremos a comentar cada uno de los casos de uso mediante unas tablas.

3.1 Diagrama entidad-relación

En este apartado de la fase de análisis vamos a comentar el diagrama entidad-relación. En el diagrama no se pueden apreciar los atributos, hemos decidido explicarlos de forma más detallada en el apartado reservado para la capa de persistencia que podremos encontrar en la fase de diseño.

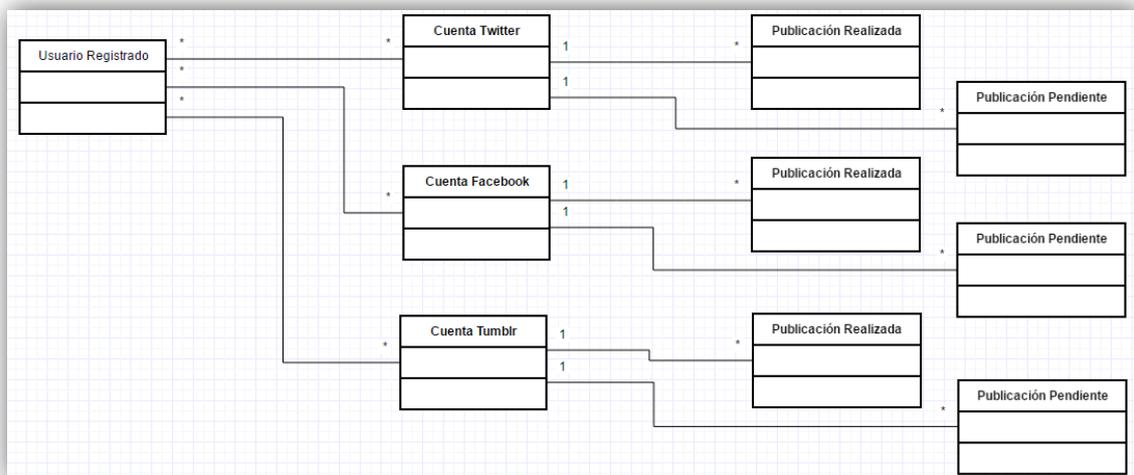


Figura 1 Diagrama Entidad-Relación

Como podemos apreciar en la Figura 1, un usuario registrado puede poseer cero, una o muchas cuentas de Twitter, de Facebook y de Tumblr, de igual forma las cuentas pueden pertenecer a uno o muchos usuarios registrados.

En cuanto a las publicaciones, podemos observar que tenemos publicaciones realizadas y publicaciones pendientes de cada red social. Cada cuenta de una red social podrá tener cero, una o muchas publicaciones pendientes y cero, una o muchas publicaciones realizadas. Las publicaciones realizadas o pendientes de una red social solo podrán estar relacionadas con una cuenta de su misma red social.

Como veremos más adelante, estas relaciones se llevarán a cabo asociando el campo “nombreCuenta” de la publicación con el campo “nombreCuenta” de la cuenta de la red social. Para relacionar esta con el usuario registrado, utilizaremos el campo “propietario” de la cuenta, que deberá coincidir con el campo “nombreUsuario” del usuario registrado.

3.2 Casos de uso

En este apartado vamos a describir cada uno de los casos de uso utilizando tablas, así podremos seguir paso a paso cómo se desarrolla cada acción que puede realizar el usuario. Para empezar, vamos a ver el diagrama de casos de uso para poder hacernos una idea rápidamente, a continuación pasaremos a describir los casos de uso detalladamente.

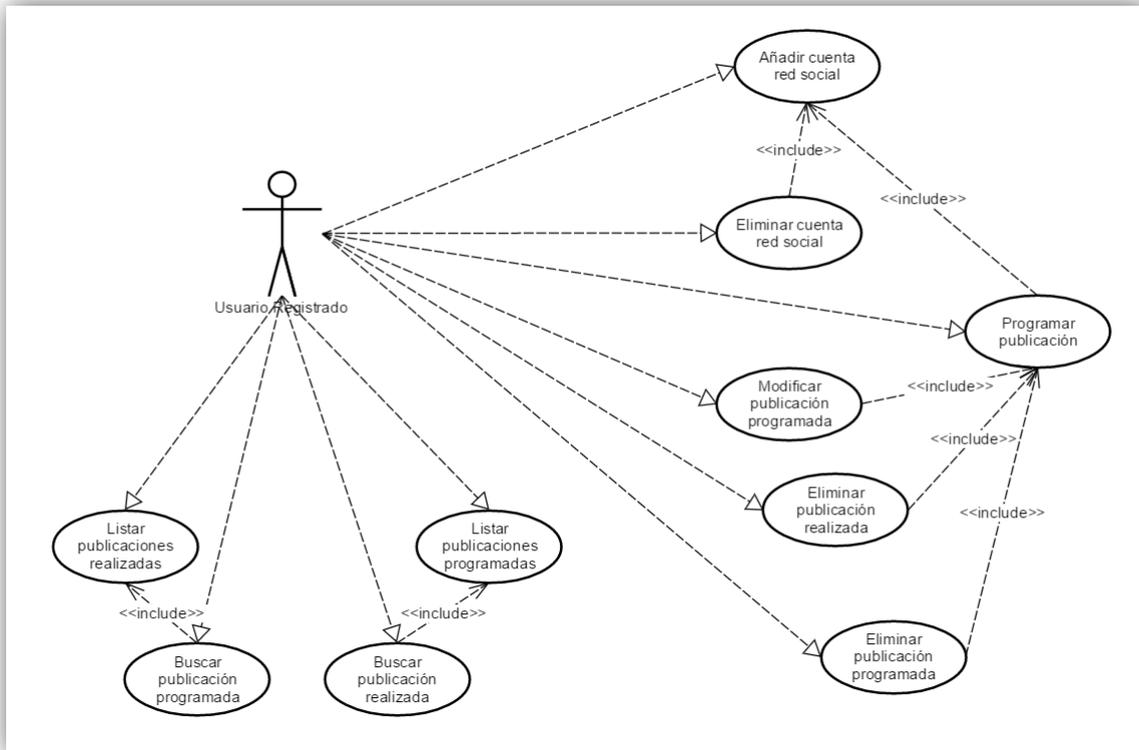


Figura 2 Diagrama de casos de uso

Vamos a pasar ahora a exponer cada uno de los casos de uso de forma detallada, como hemos comentado anteriormente. Cabe señalar que hemos obviado la necesidad de estar registrado y haber iniciado sesión para poder realizar las acciones que describiremos a continuación.

| | | |
|---|---|--|
| Nombre | Eliminar cuenta red social | |
| Descripción | El usuario podrá eliminar la cuenta que desee de las tres redes sociales disponibles, Twitter, Facebook y Tumblr. | |
| Precondición | El usuario debe tener la cuenta añadida en nuestra aplicación web. | |
| Curso Normal | Alternativas | |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de administrar cuentas. | | |
| 3) El usuario selecciona la opción de eliminar cuenta de la red social que desee. | | |
| 4) La aplicación web elimina las credenciales de la cuenta de la base de datos. | | |
| 5) La cuenta eliminada desaparece de las cuentas disponibles. | | |

| | | |
|---|--|--|
| Nombre | Añadir cuenta red social | |
| Descripción | El usuario podrá añadir una cuenta de las tres redes sociales disponibles, Twitter, Facebook y Tumblr. | |
| Precondición | El usuario debe disponer de una cuenta en la red social que desee añadir. | |
| Curso Normal | | Alternativas |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de administrar cuentas. | | |
| 3) El usuario selecciona la opción de añadir cuenta de la red social que desee. | | |
| 4) La aplicación web se comunica con la red social y redirige al usuario. | | |
| 5) El usuario inicia sesión en la red social y acepta todos los permisos necesarios. | | 5.1) El usuario no inicia sesión correctamente y lo tiene que volver a intentar. |
| 6) La red social redirige al usuario de vuelta a la web y la aplicación web almacena las credenciales de la red social. | | |
| 7) La nueva cuenta aparece en el sitio correspondiente de la sección de administrar cuentas. | | |

| | | |
|--|---|--------------|
| Nombre | Listar publicaciones programadas | |
| Descripción | El usuario podrá listar las publicaciones que estén pendientes de publicar. Aparecerán un máximo de cinco por red social. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad. | |
| Curso Normal | | Alternativas |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones pendientes. | | |
| 3) La aplicación web le muestra las cinco próximas publicaciones que se van a publicar de cada red social. | | |

| | | |
|--|---|--------------|
| Nombre | Listar publicaciones publicadas | |
| Descripción | El usuario podrá listar las publicaciones que ya hayan sido publicadas por la aplicación. Aparecerán un máximo de cinco por red social. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad y esta ha tenido que ser publicada. | |
| Curso Normal | | Alternativas |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones publicadas. | | |
| 3) La aplicación web le muestra las cinco publicaciones que se han publicado recientemente de cada red social. | | |

| | | |
|---|--|--|
| Nombre | Programar publicación | |
| Descripción | El usuario podrá programar una publicación ya sea para las tres redes sociales como para dos o una. | |
| Precondición | El usuario debe disponer de una cuenta en cada red social en la que desee programar una publicación. | |
| Curso Normal | Alternativas | |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de programar contenido. | | |
| 3) El usuario rellena los campos necesarios de cada red social en las que desea programar el contenido. | | |
| 4) El usuario selecciona la opción de programar. | | |
| 5) La aplicación web guarda las publicaciones en sus respectivas tablas de la base de datos. | | |

| | | |
|--|---|--|
| Nombre | Modificar publicación programada | |
| Descripción | El usuario podrá modificar el contenido, la fecha de publicación y la cuenta de una publicación que esté pendiente de publicar. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad. | |
| Curso Normal | Alternativas | |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones pendientes. | | |
| 3) La aplicación web le muestra las cinco próximas publicaciones que se van a publicar de cada red social. | | |
| 4) El usuario no encuentra la que quiere modificar y utiliza el buscador para encontrarla. | 4.1) El usuario encuentra la publicación que buscaba entre las listadas por la aplicación web. | |
| 5) El usuario selecciona la opción de modificar publicación. | | |
| 6) La aplicación web redirige al usuario a la página donde podrá modificar la publicación. | | |
| 7) El usuario modifica los campos que desea y selecciona la opción de modificar. | | |
| 8) La aplicación web actualiza la tabla pertinente de la base de datos. | | |

| | | |
|--|---|--|
| Nombre | Eliminar publicación programada | |
| Descripción | El usuario podrá eliminar una publicación que esté pendiente de publicar. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad. | |
| Curso Normal | | Alternativas |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones pendientes. | | |
| 3) La aplicación web le muestra las cinco próximas publicaciones que se van a publicar de cada red social. | | |
| 4) El usuario no encuentra la publicación que quiere eliminar y utiliza el buscador para encontrarla. | | 4.1) El usuario encuentra la publicación que buscaba entre las listadas por la aplicación web. |
| 5) El usuario selecciona la opción de eliminar publicación. | | |
| 6) La aplicación web actualiza la tabla pertinente de la base de datos. | | |

| | | |
|---|---|--|
| Nombre | Eliminar publicación publicada | |
| Descripción | El usuario podrá eliminar una publicación que esté publicada en la red social pertinente. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad y esta ha tenido que ser publicada. | |
| Curso Normal | | Alternativas |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones publicadas. | | |
| 3) La aplicación web le muestra las cinco publicaciones que se han publicado recientemente de cada red social. | | |
| 4) El usuario no encuentra la publicación que quiere eliminar y utiliza el buscador para encontrarla. | | 4.1) El usuario encuentra la publicación que buscaba entre las listadas por la aplicación web. |
| 5) El usuario selecciona la opción de eliminar publicación. | | |
| 6) La aplicación web se comunica con la red social para eliminar la publicación y posteriormente actualiza la tabla pertinente de la base de datos. | | |

| | | |
|--|---|--|
| Nombre | Buscar publicaciones programadas | |
| Descripción | El usuario podrá buscar entre las publicaciones que estén pendientes de publicar. Aparecerán un máximo de cinco por red social. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad. | |
| Curso Normal | Alternativas | |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones pendientes. | | |
| 3) La aplicación web le muestra las cinco próximas publicaciones que se van a publicar de cada red social. | | |
| 4) El usuario escribe palabras relacionadas con la publicación que desea buscar en el buscador. | | |
| 5) La aplicación web les muestras las publicaciones que coinciden con el término de búsqueda. | | |

| | | |
|--|---|--|
| Nombre | Buscar publicaciones programadas | |
| Descripción | El usuario podrá buscar entre las publicaciones que estén pendientes de publicar. Aparecerán un máximo de cinco por red social. | |
| Precondición | El usuario debe haber programado una publicación con anterioridad. | |
| Curso Normal | Alternativas | |
| 1) El usuario accede a la web. | | |
| 2) El usuario navega hasta la sección de publicaciones publicadas. | | |
| 3) La aplicación web le muestra las cinco publicaciones que se han publicado recientemente de cada red social. | | |
| 4) El usuario escribe palabras relacionadas con la publicación que desea buscar en el buscador. | | |
| 5) La aplicación web les muestras las publicaciones que coinciden con el término de búsqueda. | | |

4. Diseño

A continuación, vamos a dividir la fase de diseño en tres apartados. Empezaremos hablando de la capa de presentación, en la que mostraremos unos bocetos de cada una de las pantallas de la aplicación.

Seguiremos comentando la capa de persistencia, en la que mediante una figura mostraremos las tablas necesarias con sus campos y las relaciones que tendrán entre ellas. Por último, pasaremos a comentar la capa de lógica, en la que expondremos un diagrama de flujo que mostrará las acciones que el usuario podrá realizar.

4.1 Capa de presentación

Como ya hemos señalado, vamos a hacer una pequeña demostración con bocetos de lo que será la futura aplicación web.

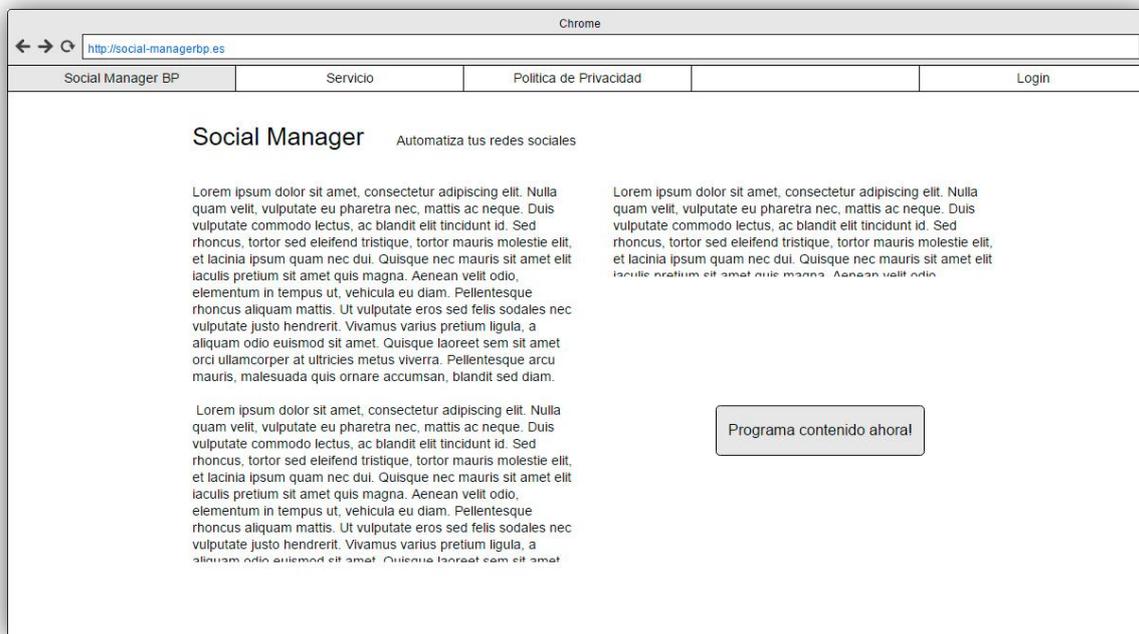


Figura 3 Boceto página principal

Como podemos observar en la Figura 3, la página de bienvenida de nuestra aplicación web dispondría de una pequeña descripción de la aplicación y un botón para acceder a las funcionalidades. También vemos el menú superior que estará presente en toda la web.



Figura 4 Boceto inicio de sesión

En el boceto de la Figura 4 mostramos como sería la página para iniciar sesión, sencilla y sin mucha decoración, ya que no nos vamos a centrar en vender la aplicación.



Figura 5 Boceto publicaciones pendientes

Esta página contendrá una de las funcionalidades principales, en ella podremos listar las publicaciones pendientes de ser publicadas o buscar otras que contengan la palabra deseada. Además una vez encontradas las publicaciones que deseemos, las podemos eliminar o modificar a nuestro gusto.

En caso de querer modificarlas o eliminarlas, lo tendríamos que hacer una a una pulsando el botón adecuado.



Figura 6 Boceto publicaciones publicadas

Al igual que en la Figura 5, el boceto de la Figura 6 nos permitirá listar y buscar publicaciones, esta vez las que ya hayan sido publicadas. En este caso las publicaciones solo podrán ser eliminadas de la respectiva red social.

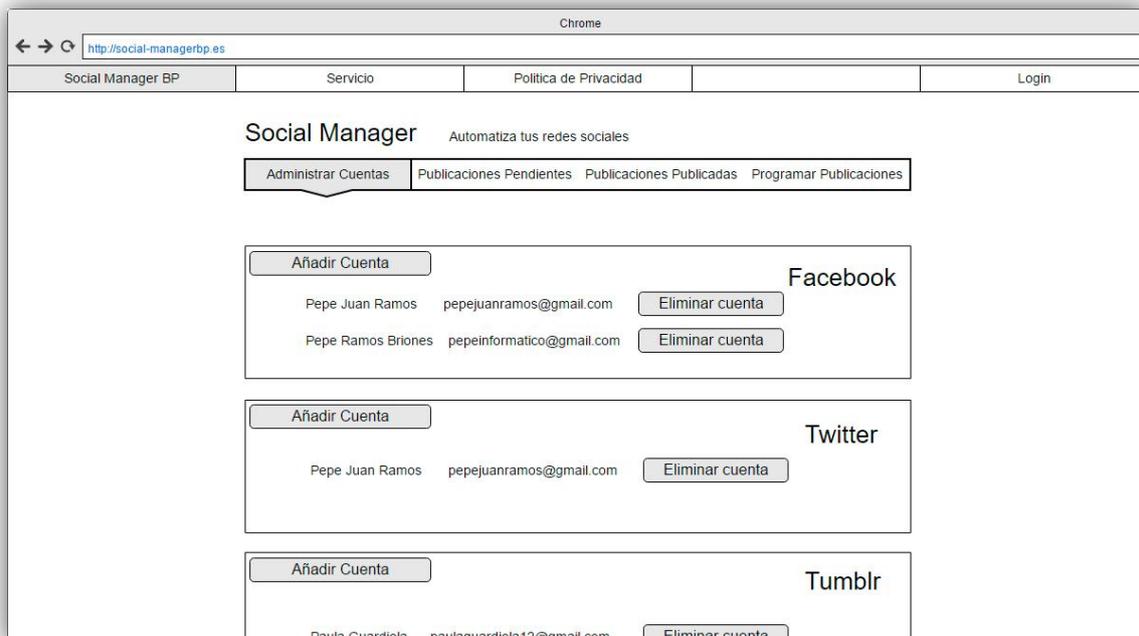


Figura 7 Boceto administrar cuentas

En la página de la Figura 7, podremos administrar nuestras cuentas de las redes sociales que tenemos conectadas a la aplicación web. Podremos ver las que ya tenemos añadidas, añadir cuentas nuevas o eliminar las cuentas que creamos oportuno.

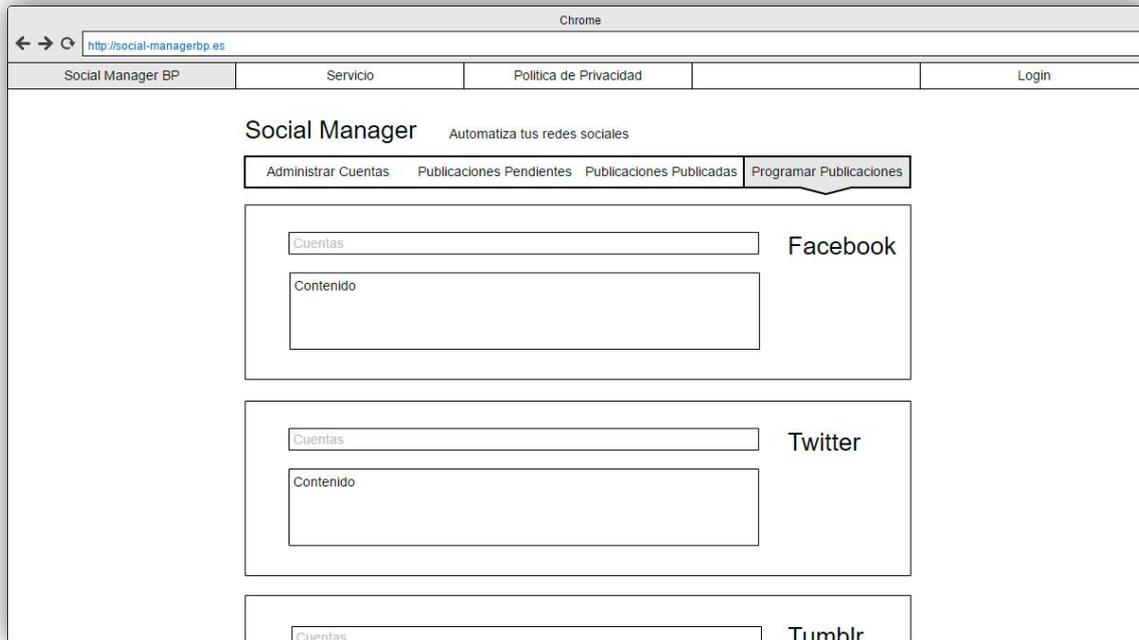


Figura 8 Boceto programar publicaciones

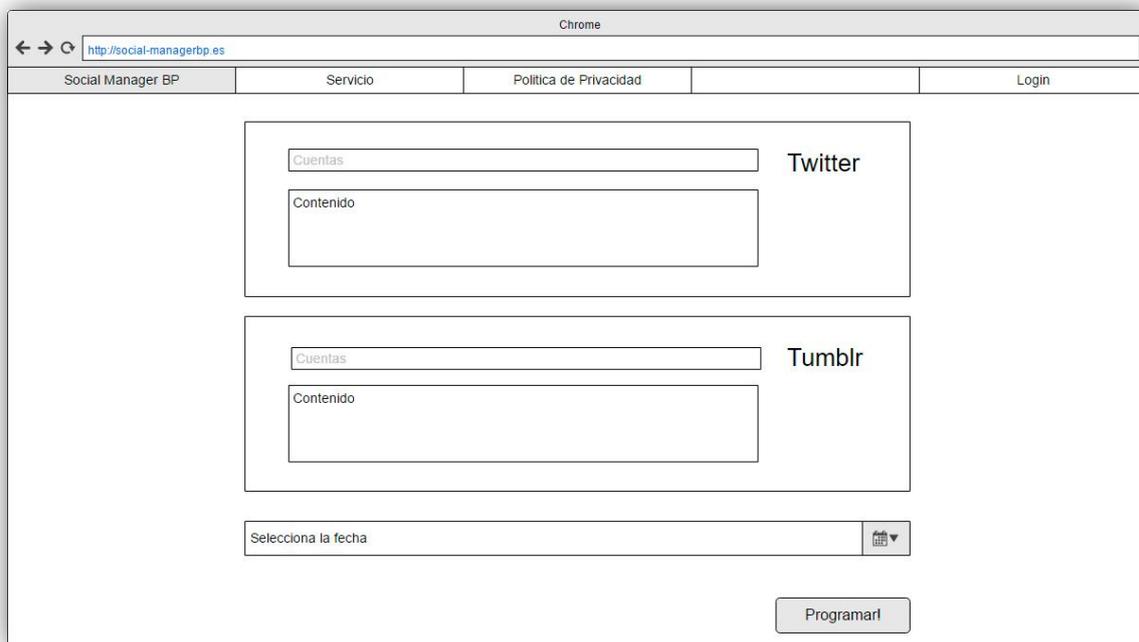


Figura 9 Boceto programar publicaciones

Para mostrar la página encargada de programar las publicaciones, hemos decidido insertar dos figuras. Como se puede apreciar en esa página, tendríamos un pequeño formulario para cada red social en los que rellenaríamos los datos necesarios.

Rellenados estos formularios, podríamos escoger la fecha con un pequeño calendario y seleccionar la opción programar para que la aplicación almacenara los datos.

4.2 Capa de persistencia

Después de analizar las tablas necesarias para almacenar todos los datos de la aplicación web, el esquema de la base de datos sería el siguiente:

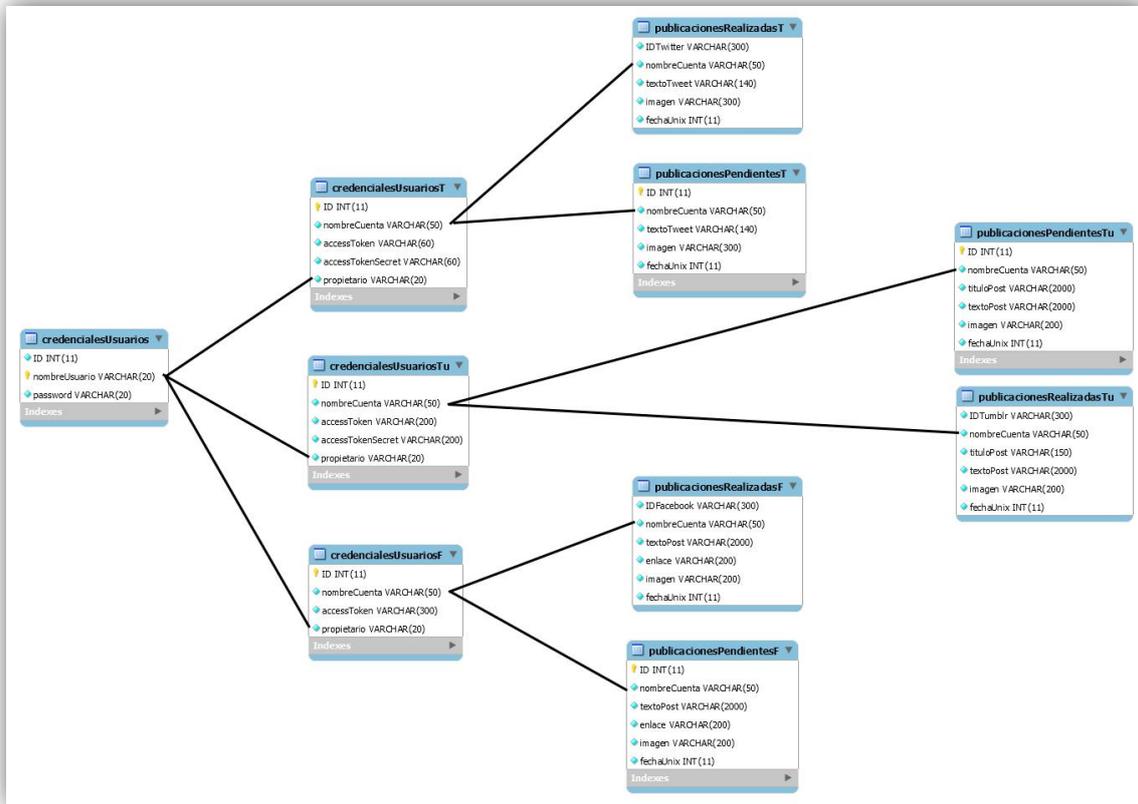


Figura 9 Capa de persistencia

Como podemos observar, en la Figura 9 se pueden distinguir dos tipos de tablas, las que almacenarán credenciales de usuario y las que se encargarán de almacenar los datos relacionados con las publicaciones.

Vamos a analizar primero el grupo de tablas de credenciales, en el que podemos encontrar la tabla credencialesUsuarios, donde se almacenarán los datos de los usuarios que estén autorizados a acceder a nuestra aplicación web, y también las tablas credencialesUsuariosT, credencialesUsuariosTu y credencialesUsuariosF. Estas tablas se van a encargar de almacenar las credenciales necesarias para publicar en las distintas redes sociales. Las terminaciones de los nombres, T, Tu y F indican que las tablas guardan la información referente a Twitter, Tumblr y Facebook respectivamente.

Vamos ahora a explicar los principales campos de estas tablas:

- nombreCuenta: Con este campo, podremos relacionar las credenciales de la cuenta con las publicaciones.
- accessToken y accessTokenSecret: Estos campos, son los proporcionados por las redes sociales, para qué junto a unos identificadores podamos autenticarnos y publicar contenido a través de las APIs.

- propietario: Con este campo, relacionamos las cuentas de las redes sociales con las cuentas de nuestra aplicación web.

Centrándonos ahora en las tablas que almacenan la información relacionada con las publicaciones, podemos observar que unas se encargan de almacenar las que están pendientes de ser publicadas y otras se encargan de las que ya han sido publicadas.

Los campos principales de estas tablas son los siguientes:

- IDFacebook, IDTwitter, IDTumbler: Estos campos almacenan el ID que nos devuelven las redes sociales al publicar contenido en ellas, nos servirá para poder borrar las publicaciones en caso de que nos hayamos equivocado o simplemente no queramos que el contenido esté publicado.
- nombreCuenta: Como en las demás tablas, la función principal de este campo es la de relacionar las publicaciones con las cuentas de las redes sociales.
- imagen: En este campo, almacenaremos la URL de la imagen que el usuario desee publicar.
- fechaUnix: Aquí almacenaremos la fecha en la que la publicación debe ser publicada, en caso de que ya lo esté representaría la fecha de publicación. La fecha se almacena en formato Unix.
- tituloPost, textoPost, textoTweet, enlace: Estos campos hacen referencia al contenido de las publicaciones.

Como podemos observar, cada red social tiene alguna pequeña diferencia en los campos que almacenamos respecto a las otras. Cada API requiere unos campos para autenticarse o unos campos que deben ser publicados, por ejemplo, en Twitter solo necesitamos el texto del tweet y la imagen, en cambio en Tumblr necesitamos, a parte del texto y la imagen, un campo para almacenar el título de la publicación.

4.3 Capa lógica

En este apartado, vamos a mostrar las acciones que se pueden realizar en la aplicación web y cómo podemos navegar por ella, para ello nos vamos a ayudar de un mapa web. Cabe señalar que las líneas de distintos colores solo sirven para diferenciar las líneas de cada uno de los orígenes de forma más sencilla.



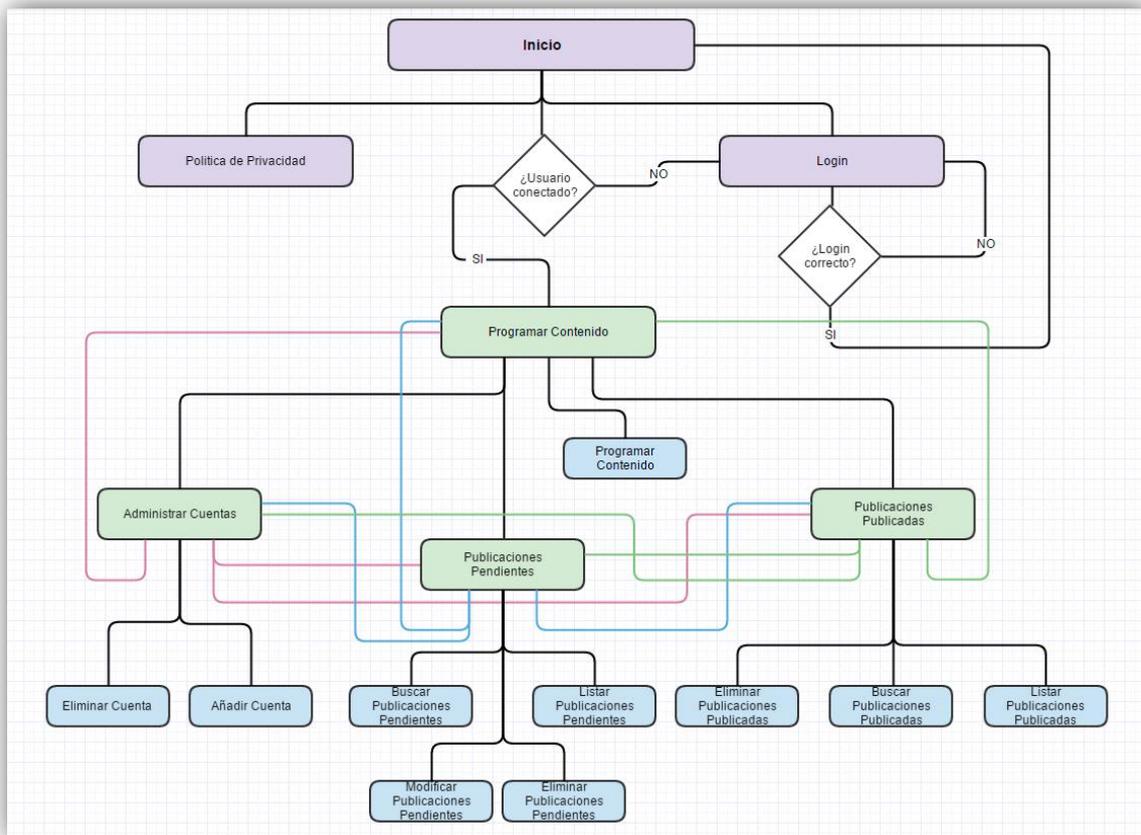


Figura 10 Mapa web

Para qué se comprenda mejor la Figura 10 vamos a comentar algunos aspectos. Las páginas más importantes, donde se pueden utilizar las principales funcionalidades, están marcadas en color verde.

Las páginas menos importantes están marcadas en color fucsia y las marcadas en color azul hacen referencia a las funcionalidades que se pueden realizar en cada página.

Como podemos ver, programar contenido, administrar cuentas, publicaciones pendientes y publicaciones publicadas, están interconectadas para qué podamos navegar entre las páginas principales de forma sencilla.

Además de lo ya comentado, se puede apreciar en la figura, la necesidad de haber iniciado sesión para poder acceder a las páginas donde podemos utilizar todas las funcionalidades.

En caso de no haber iniciado sesión anteriormente, deberíamos hacerlo para poder acceder a todas las funcionalidades de la aplicación.

5. Detalles de implementación

En este apartado de la memoria, vamos a ver la estructura de ficheros resultante tras haber implementado la aplicación. Comentaremos el contenido de los ficheros más importantes explicando que funcionalidades o métodos implementan. Además, veremos las tecnologías que hemos utilizado para la realización de este proyecto.

5.1 Tecnologías utilizadas

En este apartado, hablaremos de las tecnologías que hemos utilizado, tanto de aplicaciones utilizadas como de lenguajes de programación, bibliotecas, etc.

La base de este proyecto está formada por cuatro lenguajes. Con PHP hemos implementado la mayor parte de las funcionalidades, ayudándonos de MySQL para acceder a la base de datos. Utilizando HTML5 y CSS3 hemos programado la mayor parte de la interfaz de la aplicación.

Vamos ahora a comentar, con más detalle, cada una de las tecnologías que hemos utilizado.

5.1.1 HTML5

Las siglas HTML provienen del inglés y hacen referencia a *HyperText Markup Language*, que significa Lenguaje de Marcado de Hipertexto. Gracias a este lenguaje, el navegador renderiza las páginas web en pantalla cargando el documento HTML e interpretando las etiquetas.

Este lenguaje, es la base del desarrollo de cualquier aplicación web, ya que la mayor parte del contenido de las webs está creado en HTML. En mi caso he utilizado HTML5, que se diferencia mayoritariamente por la inclusión de nuevos atributos y nuevas etiquetas como pueden ser: *header*, *footer*, *section*, *canvas*, *audio*, entre muchas otras.

Sitio oficial: <https://www.w3.org/html/>

5.1.2 CSS3

El lenguaje CSS3 es la última versión de CSS, que ha ido evolucionando a lo largo del tiempo, al igual que otros muchos lenguajes como el HTML nombrado anteriormente. Proviene de las siglas en inglés de *Cascading Stylesheets* que traducido significa hojas de estilo en cascada.

Este famoso lenguaje sirve para crear y definir la presentación de un documento escrito en un lenguaje de marcado como HTML. El principal objetivo es marcar las separaciones entre el contenido de una web, así como definir colores y fuentes.

Sitio oficial: <https://www.w3.org/Style/CSS/>



5.1.3 JavaScript

JavaScript es considerado un lenguaje de programación interpretado que proviene del estándar *ECMAScript*. Su función principal es la creación de interfaces de usuario y páginas web dinámicas. Este lenguaje de programación se define como orientado a objetos, basado en prototipos, imperativo, dinámico y débilmente tipado.

En nuestra aplicación web, hemos utilizado este lenguaje para modificar dinámicamente elementos de la web como podrían ser los formularios de programación de contenido que se pueden mostrar u ocultar.

Sitio donde poder aprender los conceptos básicos: <https://www.javascript.com/>

5.1.4 AJAX y jQuery

En este subapartado hemos decidido juntar AJAX y jQuery, los dos son derivados del ya comentado JavaScript y al igual que él, sirven para desarrollar páginas webs dinámicas. En el caso de jQuery, estaríamos hablando de una biblioteca de JavaScript que nos ayuda a crear animaciones, manejar eventos e interactuar con los documentos HTML.

Por otro lado, AJAX es una técnica de desarrollo web basada en JavaScript que se ejecuta en la máquina del cliente, mientras se comunica de forma asíncrona con la máquina servidor.

En nuestro caso, el uso principal que le hemos dado a estas tecnologías lo podemos encontrar en las páginas que se encargan de administrar los contenidos pendientes de publicar o los que ya han sido publicados. Concretamente hemos utilizado estas tecnologías para implementar los buscadores que actualizan el contenido de la página en tiempo real.

Sitio oficial de jQuery: <https://jquery.com/>

Enlace al apartado de AJAX: <http://api.jquery.com/jquery.ajax/>

5.1.5 PHP 5.6.30

PHP es un acrónimo recursivo “*PHP Hypertext Preprocessor*”, su principal característica es que puede ser insertado en documentos HTML para generar lenguaje HTML de forma automática. A diferencia del lenguaje JavaScript, este lenguaje se ejecuta en la máquina servidor generando código HTML y mandándolo a la máquina del cliente.

En nuestro caso, este lenguaje es de los más importantes en el proyecto, con él hemos implementado todas las funcionalidades que requieren un acceso a la base de datos, como podría ser: la programación de contenido, buscar las publicaciones ya publicadas, eliminar y modificar publicaciones... todas las funcionalidades tienen en mayor o menor medida parte de código PHP.

Sitio oficial: <http://php.net/>

5.1.6 MySQL

MySQL es el sistema de gestión de bases de datos relacional más popular del mundo. Una curiosidad de este sistema, es que se ofrecen dos tipos de licencias, una pública general y una versión *Enterprise* para empresas que deseen integrar este famoso sistema a su infraestructura de datos.

El objetivo principal de MySQL es proporcionar velocidad y robustez, soporta una gran cantidad de tipos de datos y es capaz de utilizar la potencia de sistemas multihilo. En nuestro caso hemos utilizado este gestor de bases de datos, ya que en la documentación de PHP se pueden encontrar una gran variedad de métodos con los que interactuar con este sistema.

Sitio oficial: <https://www.mysql.com/>

5.1.7 Bootstrap 3

Bootstrap es uno de los *frameworks* de CSS más utilizados en la actualidad. Su desarrollo empezó de la mano de Twitter, hasta que en agosto de 2011 lo liberó. Ahora continua siendo desarrollado en un repositorio de GitHub por el equipo de getbootstrap. Se caracteriza por facilitar la creación de interfaces de usuario limpias y totalmente adaptables a todo tipo de dispositivos, ya sean móviles, tabletas u ordenadores.

En nuestro caso nos ha facilitado la organización del contenido de cada una de las páginas. Gracias a este *framework* hemos conseguido desarrollar una interfaz sencilla y fácil de utilizar de una forma rápida y cómoda.

Enlace al *framework*: <http://getbootstrap.com/>

5.1.8 SDK de Facebook para PHP versión 5

Este ha sido el SDK que hemos utilizado para implementar todas las funcionalidades relacionadas con Facebook. Existe otro SDK de JavaScript que también probamos a utilizar, pero al final nos pareció más sencillo el de PHP.

Este SDK recoge una serie de funciones que nos permiten tanto añadir cuentas de Facebook a nuestra aplicación web, como realizar solicitudes para publicar o eliminar contenido.

Enlace a la biblioteca: <https://developers.facebook.com/docs/reference/php/>

5.1.9 TwitterOAuth de Abraham Williams

Después de investigar varias APIs, llegamos a la conclusión de que esta era la más sencilla y mejor documentada para utilizar con Twitter. Concretamente es una biblioteca que facilita el uso de la API de Twitter.

Cuando estábamos a punto de finalizar la implementación, nos percatamos de que la publicación de imágenes no funcionaba correctamente, así que utilizamos otra biblioteca para realizar esa tarea.

Enlace a la biblioteca: <https://twitteroauth.com/>

Enlace a la API de Twitter: <https://dev.twitter.com/rest/public>



5.1.10 Codebird-PHP

Codebird-PHP es una biblioteca que al igual que la comentada anteriormente, facilita el uso de la Api de Twitter. Como hemos señalado en el punto anterior, al intentar implementar la publicación de fotos en Twitter nos vimos con serios problemas y decidimos utilizar esta biblioteca para publicar los tweets que incluyeran una imagen.

Enlace a la biblioteca: <https://www.jublo.net/projects/codebird/php>

5.1.11 Tumblr API

Esta es la API oficial de Tumblr, con la que hemos implementado todas las funcionalidades relacionadas con la red social Tumblr. Cabe señalar que a pesar de haber tenido algunos problemas ha sido la API más sencilla de utilizar. La documentación está bien organizada y es muy fácil encontrar los conceptos que buscas, a diferencia de las de Twitter y Facebook.

Enlace a la API: <https://www.tumblr.com/docs/en/api/v2>

5.1.12 OAuth 2.0

OAuth es un protocolo estándar de autorización que se centra en simplificar los flujos de autorización a los desarrolladores. En nuestro desarrollo web ha sido clave para obtener los permisos de cada cuenta de las redes sociales.

Ya que ha sido una parte importante y compleja de nuestro proyecto, vamos a comentar el funcionamiento por encima para que se comprenda en que se basa esa simplificación de los flujos de autorización.

El flujo de OAuth 2.0 se puede dividir en dos partes, la primera consiste en obtener el *access token* y la segunda en utilizar el *access token* para acceder al servicio del que hemos obtenido la autorización. Con eso se consigue eliminar la necesidad de proporcionar el nombre de usuario y la contraseña por parte del usuario, además los *tokens* pueden tener el acceso restringido a ciertas acciones o datos sensibles.

Vamos a poner un ejemplo un poco simplificado para que se pueda entender mejor. Un usuario de nuestra aplicación va a añadir una cuenta de Facebook:

1. El usuario selecciona la opción de añadir cuenta y la aplicación manda a Facebook dos *tokens* que hacen referencia a nuestra web. Con ello conseguimos identificar que Social Manager BP quiere tener acceso a ciertos datos o funcionalidades de Facebook de la cuenta que el usuario quiere agregar.
2. Como Social Manager BP está registrado en Facebook como aplicación en la cuenta personal del desarrollador del proyecto, pide al usuario que inicie sesión y acepte los permisos que la aplicación requiere.
3. Tras ello, Facebook responde con un *token* que será utilizado por nuestra aplicación web para publicar y eliminar contenido en nombre del usuario de forma automática.

Sitio oficial: <https://oauth.net/2/>

5.1.13 Mab Tag Input

Con esta biblioteca de jQuery, hemos implementado una parte de la funcionalidad de programar contenido, gracias a ella podemos introducir más de una cuenta de una misma red social. En una primera versión de prueba, solo se podía seleccionar una cuenta mediante unos desplegados y ahora se pueden escribir varias en el campo correspondiente de cada red social.

Enlace a la biblioteca: <http://www.jqueryscript.net/form/jQuery-Tags-Input-Plugin-with-Autocomplete-Support-Mab-Tag-Input.html>

5.1.14 Bootstrap 3 Datepicker v4

Se trata de una pequeña biblioteca derivada de Bootstrap 3, con la que hemos implementado la funcionalidad de escoger la fecha deseada para las publicaciones. Como veremos en la parte de pruebas, para escoger la fecha se nos despliega un pequeño calendario, para elegir el día y posteriormente las horas y los minutos.

Enlace a la biblioteca: <https://eonasdan.github.io/bootstrap-datetimepicker/>

5.2 Aplicaciones utilizadas

Enumeradas y explicadas las principales APIs, bibliotecas y lenguajes que hemos utilizado, vamos ahora a comentar el software que hemos necesitado para el desarrollo del proyecto:

5.2.1 Atom

Atom es un editor de texto que permite la inclusión de paquetes de utilidades para que el desarrollo en cualquier lenguaje sea más cómodo. Tras investigar un poco por internet, optamos por este IDE, el principal motivo para su elección ha sido la capacidad de personalización que tiene.

Hemos conseguido instalar varios paquetes que nos han ayudado a depurar código, subir los ficheros al servidor y en general facilitarnos el desarrollo.

Sitio oficial: <https://atom.io/>

5.2.2 Sublime Text 2

Se trata de otro editor de texto parecido al que ya hemos comentado. Por preferencias personales del pasado, al tener ya instalado este editor, lo hemos utilizado para tomar notas y guardar secciones de código que queríamos utilizar más adelante, o bien retocar sin miedo a desordenar las partes de código ya terminadas.

Podríamos haber utilizado el bloc de notas de Windows, pero a nuestro entender es demasiado simple y no nos hubiera facilitado la tarea de tratar el código.

Sitio oficial: <https://sublimetext.com/2>



5.2.3 SourceTree

SourceTree es un cliente de escritorio que proporciona una interfaz para interactuar con repositorios Git y Mercurial de forma más sencilla. Cuando realizas cambios en varios ficheros, resulta muy útil, ya que te muestra que has modificado en cada uno de ellos.

Hemos utilizado esta aplicación para gestionar el control de versiones. Además nos ha servido para mantener el proyecto actualizado en dos ordenadores distintos.

Sitio oficial: <https://www.sourcetreeapp.com/>

5.2.4 Adobe Photoshop CS6

Es un editor gráfico ampliamente conocido. Se utiliza principalmente para retocar fotos de forma profesional. En nuestro proyecto, nos ha servido para retocar las imágenes utilizadas en la web, así como las capturas de pantalla insertadas en esta memoria.

Sitio oficial: <http://www.adobe.com/es/products/photoshop.html>

5.2.5 FileZilla

FileZilla es un cliente FTP de código abierto y software libre, soporta los protocolos FTP, SFTP y FTPS. En nuestro desarrollo, lo hemos utilizado para subir archivos al servidor utilizando el protocolo FTP. Posteriormente, con la inclusión de un *plugin* en Atom capaz de comunicarse con el servidor por FTP, dejamos de utilizarlo de forma regular.

Sitio oficial: <https://filezilla-project.org/>

5.2.6 Toggl

Toggl es un servicio que nos ofrece la posibilidad de contabilizar el tiempo que invertimos en la realización de cualquier tarea. Nos permite clasificar las tareas en proyectos y posteriormente generar informes que nos muestren las horas invertidas.

Esta aplicación nos ha sido muy útil para contabilizar las horas que se dedicaban al desarrollo del proyecto. Gracias a esa contabilidad, conseguimos motivarnos para cumplir ciertos objetivos de horas semanales.

Sitio oficial: <https://toggl.com/>

5.2.7 Google Chrome y Mozilla Firefox

Son dos navegadores ampliamente conocidos que hemos utilizado en varias ocasiones durante el desarrollo. Chrome ha sido nuestro principal navegador para probar que todo funcionaba correctamente, ya que la aplicación web debe funcionar perfectamente en este navegador. Su inspector de código nos ha sido de gran utilidad.

Como teníamos que disponer de varias cuentas de redes sociales abiertas a la vez, decidimos utilizar Mozilla Firefox además del ya comentado Google Chrome para poder hacer pruebas con más facilidad.

Sitio oficial de Google Chrome: <https://www.google.es/chrome/browser/desktop/>

Sitio oficial de Mozilla Firefox: <https://www.mozilla.org/es-ES/firefox/new/>

5.2.8 Gliffy

Es una aplicación web con la que podemos crear distintos tipos de diagramas, diseños de interfaces y otros tipos de bocetos. En nuestro caso nos ha servido para crear el diagrama entidad relación y el diagrama de casos de uso.

Sitio oficial: <https://www.gliffy.com/>

5.2.9 Moqups

Al igual que la aplicación anterior, su principal utilidad es la creación de bocetos, diagramas y prototipos de interfaces. Hemos utilizado esta aplicación para la creación de los bocetos de nuestra web.

Sitio oficial: <https://moqups.com/>

5.3 Estructura de los ficheros y directorios

A continuación, vamos a explicar cómo están organizados los directorios y comentaremos los principales ficheros. Cabe señalar que algunos directorios contienen una gran cantidad de directorios y ficheros, concretamente las APIs y bibliotecas, para no extendernos demasiado solo señalaremos que API o biblioteca es.

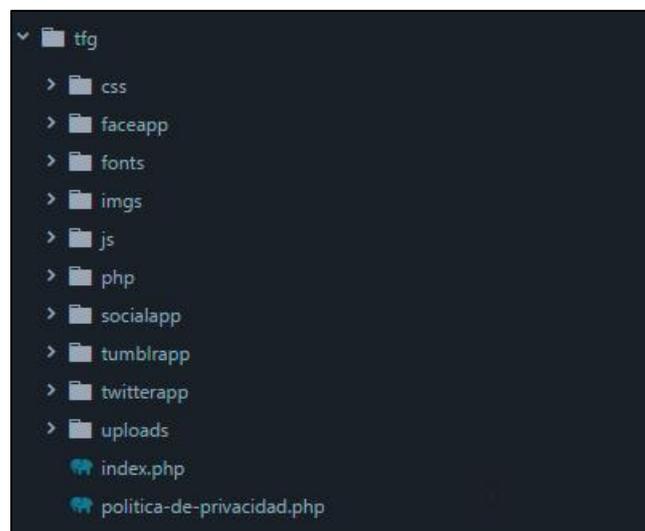


Figura 11 Estructura de ficheros general

Centrándonos en la Figura 11, vamos a comentar brevemente cada carpeta, y más adelante, nos centraremos de forma más detallada en las más importantes:

- **css:** En esta carpeta se encuentran los ficheros css que se utilizan en `index.php` y en `politica-de-privacidad.php`
- **faceapp:** Esta es una de las carpetas principales, en ella se encuentra la biblioteca de Facebook y los ficheros que nos permiten añadir una nueva cuenta.
- **fonts:** En esta carpeta se encuentran las fuentes que queremos utilizar.
- **imgs:** En este directorio encontramos las imágenes que utilizamos en `index.php`
- **js:** Aquí encontramos los ficheros .js de bootstrap y de jQuery.

- `php`: Esa carpeta contiene gran parte de las funcionalidades del proyecto, hace el papel de biblioteca generada por nosotros.
- `socialapp`: Aquí encontramos una gran cantidad de ficheros relacionados con la capa de presentación de la aplicación, así como algunos ficheros que se encargan de algunas funcionalidades.
- `tumblrapp`: En esta carpeta encontramos la biblioteca de Tumblr y los ficheros necesarios para añadir una nueva cuenta a nuestro sistema.
- `twitterapp`: En este directorio encontramos la biblioteca de Twitter y los ficheros necesarios para añadir nuevas cuentas de la red social.
- `uploads`: Aquí se guardan las imágenes que el usuario sube para que sean publicadas posteriormente en las redes sociales.
- `index.php`: Se trata de la página principal a la que cualquier usuario registrado o no puede acceder.
- `politica-de-privacidad.php`: Debido a que Facebook requería una página de política de privacidad para poder darnos permiso para utilizar su API, decidimos generar esta página.

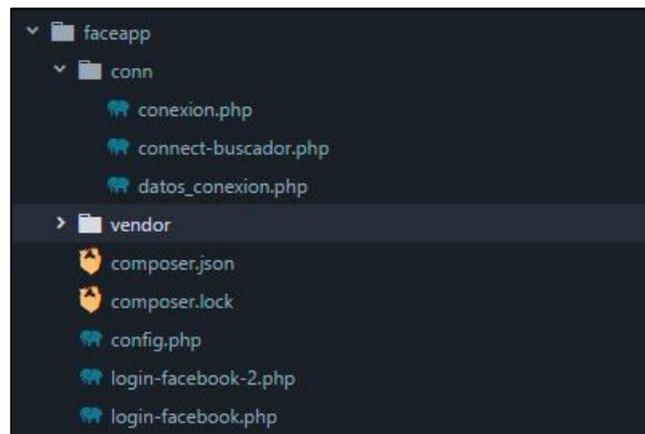


Figura 12 Estructura de ficheros faceapp

En la Figura 12, podemos apreciar los ficheros contenidos en la carpeta faceapp que hemos comentado anteriormente. Vamos a pasar a explicarlos a continuación:

- En el directorio `conn`, encontramos tres ficheros que se encargan de distintas conexiones con la base de datos.
- En el directorio `vendor`, encontramos los ficheros de la biblioteca de Facebook.
- Los ficheros `config.php`, `login-facebook-2.php` y `login-facebook.php` se encargan de añadir las nuevas cuentas de Facebook a nuestra aplicación.
- Los ficheros `composer.json` y `composer.lock` nos han permitido instalar la biblioteca de Facebook de forma sencilla.



Figura 13 Estructura de ficheros tumblrapp

Como podemos observar en la Figura 13, la estructura es muy similar a la de la figura anterior. Nos encontramos con la carpeta conn, que contiene ficheros de conexión con la base de datos, y con la carpeta tumblrroauth, que contiene la biblioteca de Tumblr.

También podemos ver los archivos connect.php y callback.php, que se encargan de conseguir las credenciales del usuario para almacenarlas en la base de datos mediante OAuth, gracias a esto conseguimos añadir cuentas de Tumblr a nuestra aplicación web.

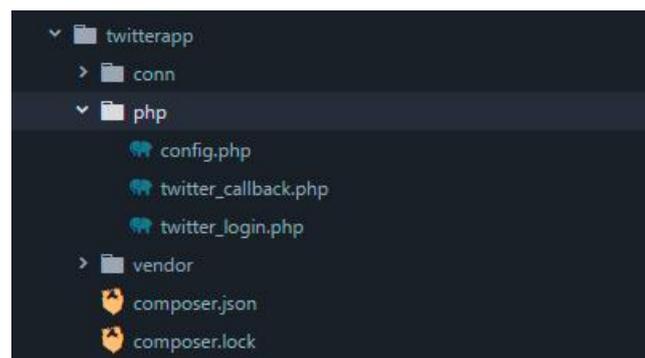


Figura 14 Estructura de ficheros twitterapp

En la Figura 14 podemos ver el contenido del directorio twitterapp. Aquí encontramos los ficheros necesarios para añadir cuentas de Twitter. Al igual que en la figura anterior, disponemos de una carpeta con ficheros de conexión, la librería de Twitter y los ficheros encargados de comunicarse con Twitter para conseguir las credenciales del usuario. Esta vez los ficheros se llaman twitter_login.php y twitter_callback.php.

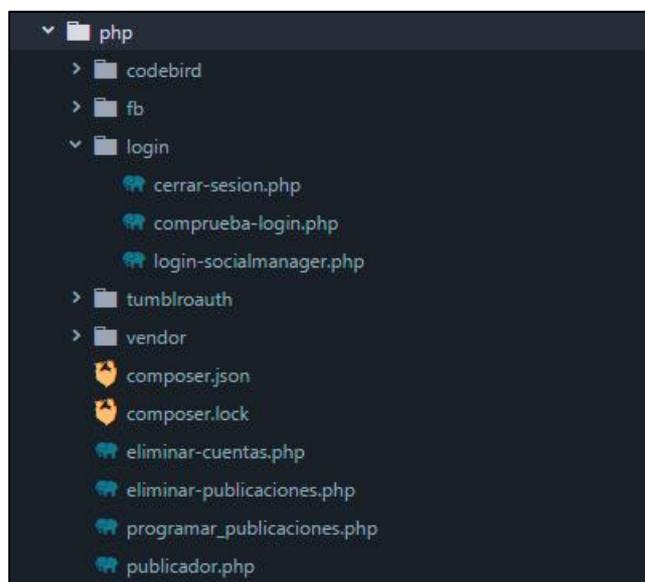


Figura 15 Estructura de ficheros php

En la Figura 15 encontramos el contenido de la carpeta `php`. Como hemos comentado anteriormente, es una de las carpetas principales de este proyecto. Vamos a explicar con más detalle las funcionalidades que implementa:

- `codebird`: En este directorio podemos encontrar la biblioteca que hemos utilizado para publicar fotos en twitter.
- `fb`: En este directorio encontramos la biblioteca de Facebook, necesaria para poder eliminar las publicaciones ya publicadas y poder publicar el contenido programado.
- `login`: En esta carpeta encontramos los ficheros encargados del inicio de sesión en nuestra web.
- `tumblrOAuth`: Esta es la biblioteca de Tumblr que nos permite realizar las acciones deseadas comunicándonos con la API.
- `vendor`, `composer.json` y `composer.lock`: Son todos los ficheros relacionados con la biblioteca de Twitter que utilizamos para publicar contenido sin imágenes y para eliminar los tweets que ya han sido publicados.
- `eliminar-cuentas.php`: En este fichero encontramos los métodos encargados de eliminar las cuentas de Twitter, Facebook y Tumblr de nuestra base de datos.
- `eliminar-publicaciones.php`: En este fichero encontramos los métodos encargados de eliminar las publicaciones. Disponemos de tres métodos encargados de eliminar las publicaciones que aún no han sido publicadas y tres métodos que eliminan las publicaciones ya publicadas llamando a la API de la respectiva red social.
- `programar_publicaciones.php`: Esta fue una primera versión del fichero que programa las publicaciones. Después de que el usuario pinche en programar, los datos son almacenados en las respectivas tablas de la base de datos, esa es la tarea que desempeña este fichero.
- `publicador.php`: Este fichero es llamado cada cierto tiempo, cuando se llama revisa la base de datos en busca de publicaciones que deban ser publicadas en ese momento. Si encuentra publicaciones que deben ser publicadas las publica en las respectivas redes sociales y actualiza las tablas de publicaciones pendientes y publicaciones publicadas de cada una de las redes sociales.

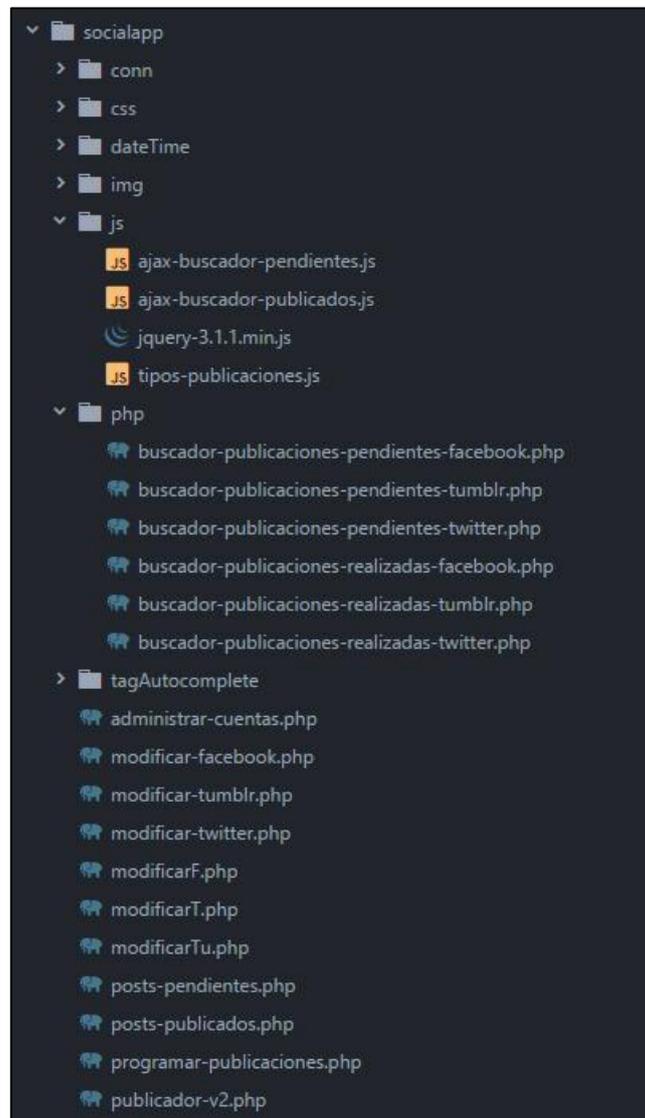


Figura 16 Estructura de ficheros socialapp

Como podemos apreciar en la Figura 16, este es el principal directorio de nuestro proyecto. Vamos a comentar los principales ficheros que encontramos en él, obviando carpetas como conn, css e img:

- `dateTime`: En este directorio encontramos la biblioteca encargada de generar un pequeño calendario que utilizamos para seleccionar la fecha de las publicaciones que queremos programar.
- `ajax-buscador-pendientes.js`: Este fichero es el encargado de actualizar de forma dinámica los resultados que se muestran en la página de publicaciones pendientes.
- `ajax-buscador-publicados.js`: Este fichero es el encargado de actualizar de forma dinámica los resultados que se muestran en la página de publicaciones que ya han sido publicadas.
- `jquery-3.1.1.min.js`: Es el fichero necesario para que todo el código contenido en ficheros js o en scripts pueda ser ejecutado correctamente.
- `tipos-publicaciones.js`: Este fichero se encarga de mostrar y esconder de forma dinámica los distintos formularios de cada red social. Además implemen-

ta la funcionalidad de la carga de imágenes de la página de programar publicaciones.

- `buscador-publicaciones-pendientes-*.php`: Estos tres ficheros se encargan de buscar en la base de datos las publicaciones que están pendientes de publicar. Con ellos implementamos el buscador que se encuentra en la página de posts pendientes.
- `buscador-publicaciones-realizadas-*.php`: Al igual de los otros tres ficheros comentados en el punto anterior, estos ficheros se encargan de buscar en la base de datos. En este caso, se buscan las publicaciones que ya han sido publicadas para mostrarlas de forma dinámica en la página de posts publicados.
- `tagAutocomplete`: En este directorio se encuentra la biblioteca que nos ha permitido implementar la selección de varias cuentas para una publicación de una misma red social.
- `administrar-cuentas`: En este fichero encontramos el código fuente de la página que se encarga de administrar las cuentas. Desde esta página podemos añadir cuentas, eliminar cuentas y ver las cuentas que tenemos añadidas a la aplicación web.
- `modificar*.php`: Estos seis ficheros se encargan de modificar las publicaciones pendientes. Gracias a ellos podemos modificar las publicaciones que aún no han sido publicadas, pudiendo cambiar el contenido de la publicación, la fecha, la imagen o lo que veamos oportuno.
- `posts-pendientes.php`: En este fichero implementamos la página que lista las publicaciones pendientes y en la que podemos buscar publicaciones para borrarlas o eliminarlas.
- `posts-publicados.php`: Al igual que el fichero comentado en el punto anterior, en este fichero encontramos la página que nos permite listar publicaciones, este caso las publicaciones publicadas. Aquí solo podremos borrar las publicaciones ya publicadas.
- `programar-publicaciones.php`: En este fichero encontramos el código de la página dedicada a programar el contenido en las distintas redes sociales.
- `publicador-v2.php`: Esta es la segunda versión del fichero que se encarga de publicar el contenido en las redes sociales. El que hemos comentado anteriormente ya no se utiliza.

6. Pruebas

En este apartado vamos a pasar las pruebas de conformidad de los estándares de W3C, comprobaremos los errores que nos muestre el validador y intentaremos arreglarlos. Posteriormente, mediante capturas de pantalla, veremos cómo se muestra nuestra aplicación web en distintas resoluciones.

Aunque la aplicación no está pensada para ser utilizada desde un móvil o tableta, también mostraremos algunas capturas por ver cómo se comporta en esas situaciones.

6.1 Conformidad con el estándar W3C

A continuación vamos a exponer los resultados obtenidos de comprobar la conformidad de nuestro sitio web con el estándar W3C. Para comprobarlo, hemos utilizado la aplicación oficial que se puede encontrar en: <https://validator.w3.org/>

Solo mostraremos algunos de los resultados de las partes más importantes de nuestra aplicación, así evitaremos extendernos demasiado.

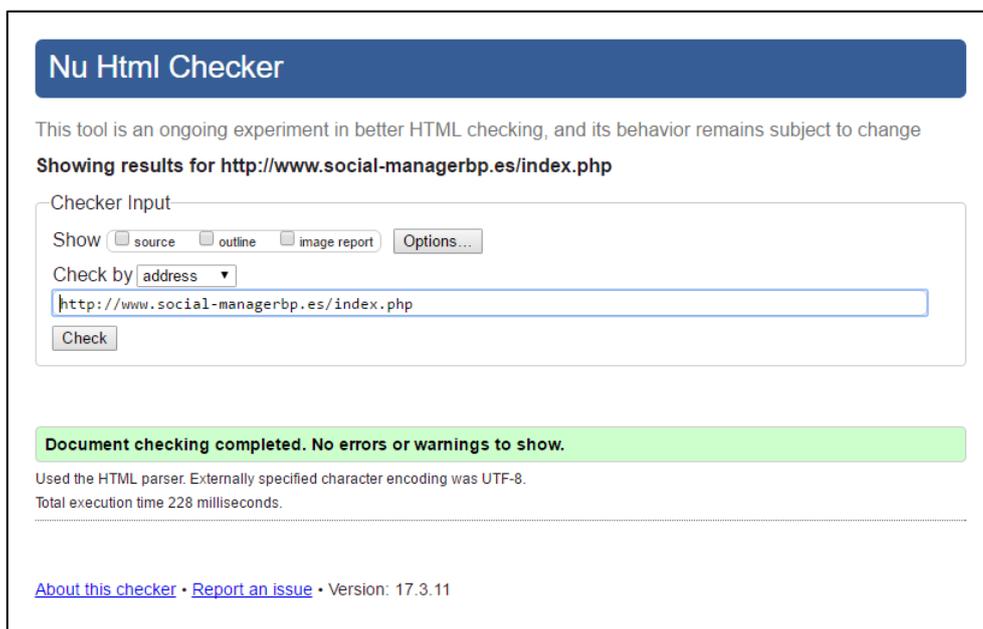


Figura 17 Conformidad con el W3C de index.php

Como podemos apreciar en la Figura 17, la página principal de nuestra aplicación web está correcta. Tras pasar el análisis por primera vez, se detectaron algunos pequeños errores que ya han sido corregidos.

The screenshot shows the Nu Html Checker interface. At the top, it says "Nu Html Checker" and "This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change". Below that, it displays "Showing results for http://www.social-managerbp.es/socialapp/administrar-cuentas.php". The "Checker Input" section includes a "Show" menu with options for "source", "outline", and "image report", and a "Check by" dropdown set to "address". The URL "http://www.social-managerbp.es/socialapp/administrar-cuentas.php" is entered in the input field, and a "Check" button is visible. Below the input section, there is a "Message Filtering" section with a "Message Filtering" button. The main area displays three error messages:

- Error** The element `button` must not appear as a descendant of the `a` element.
From line 76, column 83; to line 76, column 147
`book.php"><button style="width:70%;" type="button" class="btn btn-default">Añadir`
- Error** The element `button` must not appear as a descendant of the `a` element.
From line 87, column 78; to line 87, column 142
`nect.php"><button style="width:70%;" type="button" class="btn btn-default">Añadir`
- Error** The element `button` must not appear as a descendant of the `a` element.
From line 97, column 89; to line 97, column 153
`ogin.php"><button style="width:70%;" type="button" class="btn btn-default">Añadir`

At the bottom, a blue bar indicates "Document checking completed." Below this, it states "Used the HTML parser. Externally specified character encoding was UTF-8." and "Total execution time 215 milliseconds."

Figura 18 Conformidad con el W3C de administrar-cuentas.php

Como podemos observar en la Figura 18, se han detectado tres errores en el código que hacen referencia a los tres botones que tenemos en esa página para añadir cuentas nuevas. Estos errores no suponen ningún tipo de problema, son derivados de utilizar los `button` de Bootstrap 3 con los elementos `a` de HTML5.

Hemos optado por dejar el código como está, cambiarlo implicaría diseñar un botón con propiedades CSS que es bastante más complicado que utilizar los botones que vienen por defecto en Bootstrap 3.

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://www.social-managerbp.es/socialapp/posts-pendientes.php>

Checker Input

Show source outline image report

Check by

<http://www.social-managerbp.es/socialapp/posts-pendientes.php>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 205 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.3.11

Figura 19 Conformidad con el W3C de [posts-pendientes.php](http://www.social-managerbp.es/socialapp/posts-pendientes.php)

Nu Html Checker

This tool is an ongoing experiment in better HTML checking, and its behavior remains subject to change

Showing results for <http://www.social-managerbp.es/socialapp/posts-publicados.php>

Checker Input

Show source outline image report

Check by

<http://www.social-managerbp.es/socialapp/posts-publicados.php>

Document checking completed. No errors or warnings to show.

Used the HTML parser. Externally specified character encoding was UTF-8.
Total execution time 213 milliseconds.

[About this checker](#) • [Report an issue](#) • Version: 17.3.13

Figura 20 Conformidad con el W3C de [posts-publicados.php](http://www.social-managerbp.es/socialapp/posts-publicados.php)

Tras pasar el *checker* por las páginas de [posts-publicados.php](http://www.social-managerbp.es/socialapp/posts-publicados.php) y [posts-pendientes.php](http://www.social-managerbp.es/socialapp/posts-pendientes.php) y corregir los errores que nos mostraba, podemos ver el resultado en las figuras 19 y 20. En este caso, no había ningún error relacionado con botones o aspectos visuales de la web, así que decidimos corregirlos todos.

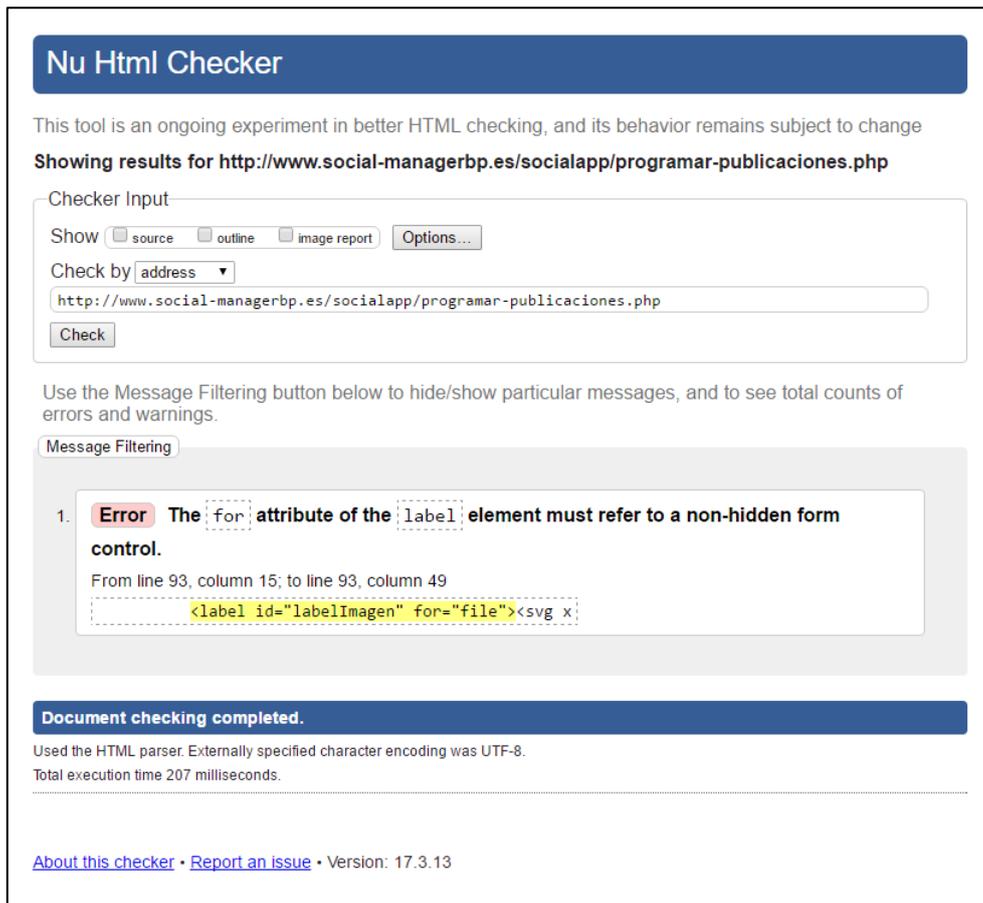


Figura 21 Conformidad con el W3C de programar-publicaciones.php

Al igual que en la página de administrar-cuentas.php, podemos ver que en la Figura 21 nos hemos dejado un error por corregir. El error hace referencia a la funcionalidad de subir imágenes a las publicaciones.

Por cuestiones visuales, hemos decidido esconder el texto que sale por defecto cuando añades un elemento para subir archivos. Así hemos podido mostrar un botón personalizado más bonito.

6.2 Visualización en distintas resoluciones

A continuación vamos a mostrar cómo se visualiza el contenido de algunas de nuestras páginas principales. Hemos decidido probar dos resoluciones bastante utilizadas de ordenador, una de tableta y otra de móvil.

Como podremos apreciar en las capturas, el contenido se muestra perfectamente en las resoluciones que hemos probado de PC. En la tableta el contenido se muestra adecuadamente y en el móvil queda un poco desordenado.

Hay que señalar que este proyecto está pensado para utilizar en un ordenador con un monitor actual, en esas resoluciones el contenido se muestra perfectamente. Vamos ahora a mostrar las capturas.

6.2.1 Resolución de PC 1920x1080

En las siguientes capturas podemos ver que el contenido se muestra a la perfección, ya que la aplicación está pensada para utilizarse con esta resolución o la que mostraremos en el siguiente punto.

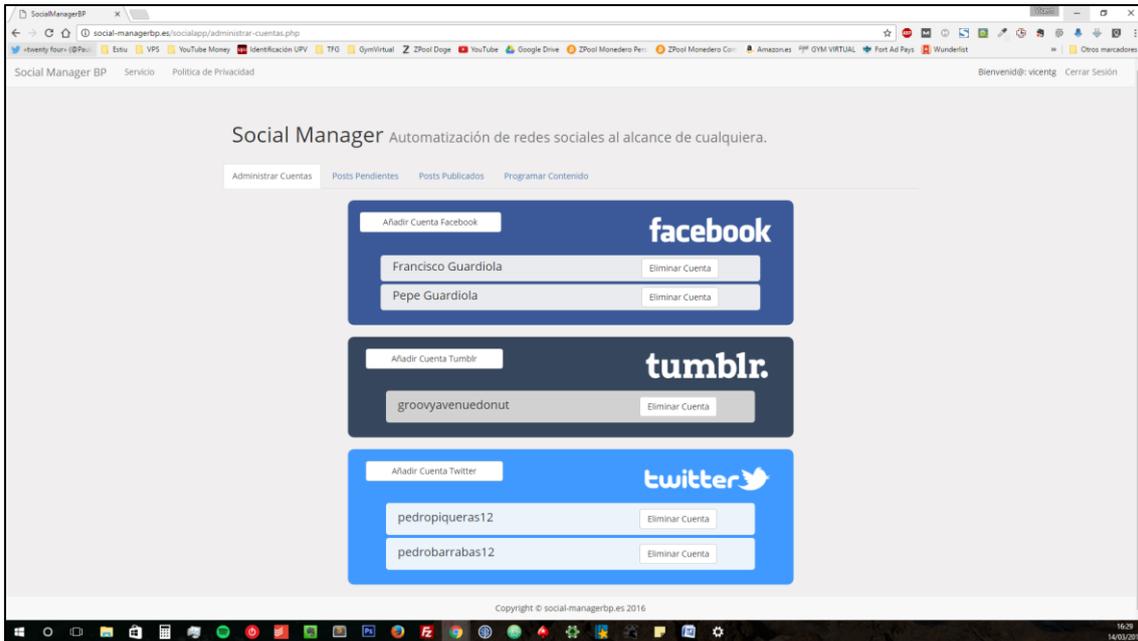


Figura 22 Prueba administrar cuentas con resolución 1920x1080

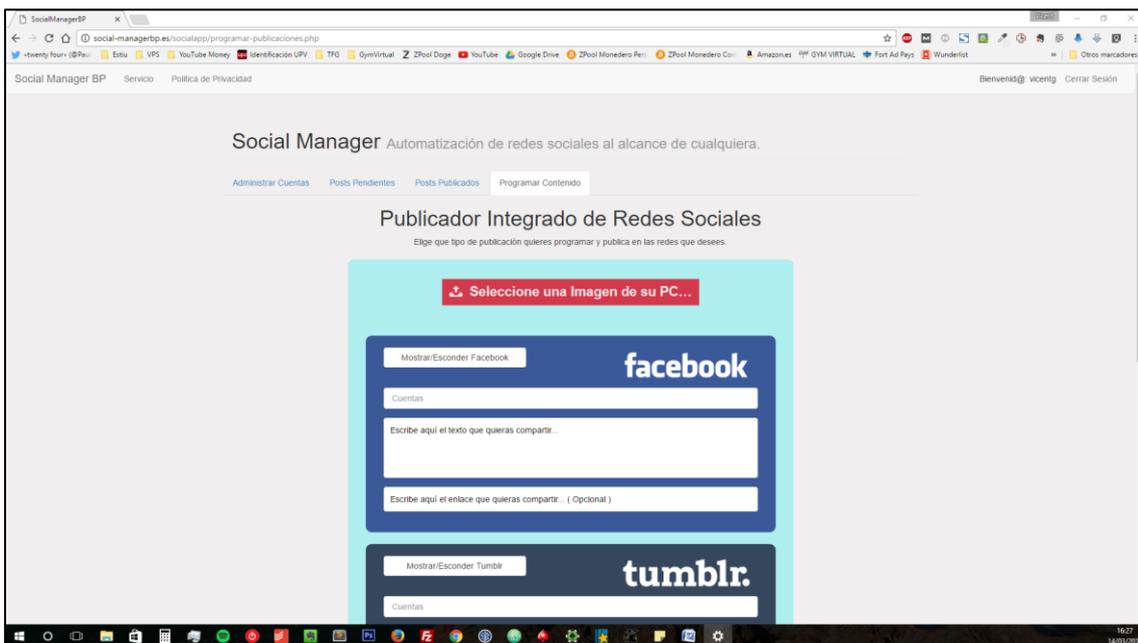


Figura 23 Prueba programar contenido con resolución 1920x1080



Figura 24 Prueba publicaciones pendientes con resolución 1920x1080

6.2.2 Resolución de PC 1366x768

Como ya hemos comentado anteriormente, en estas capturas vamos a ver como el contenido se muestra correctamente. En esta resolución, nuestra aplicación sigue siendo completamente usable sin ningún problema.

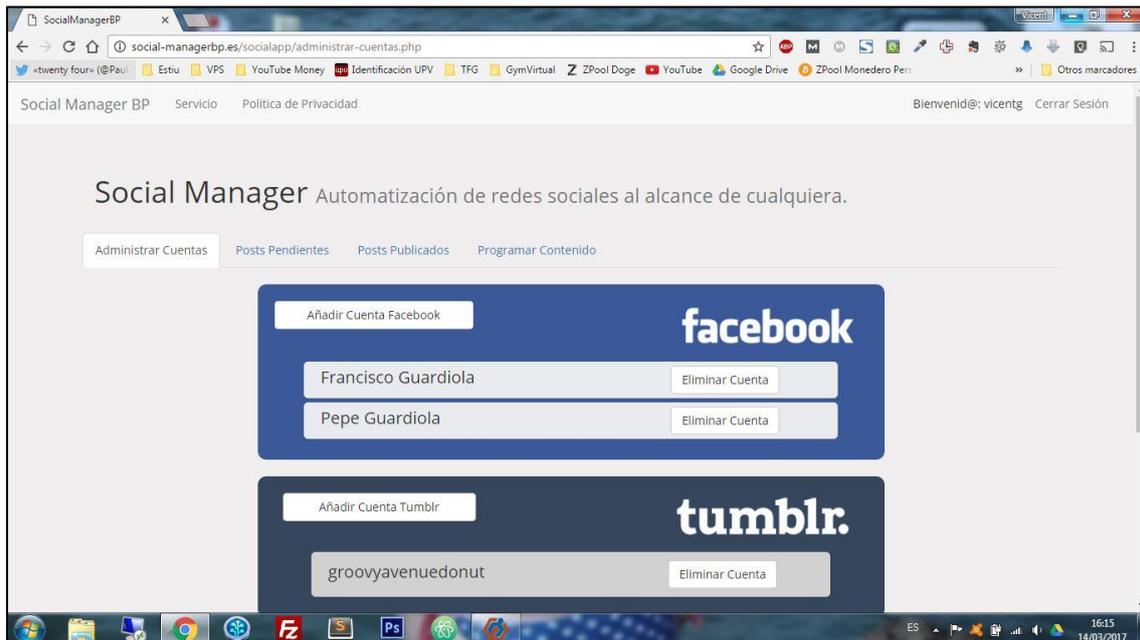


Figura 25 Prueba administrar cuentas con resolución 1366x768



Figura 26 Prueba programar contenido con resolución 1366x768

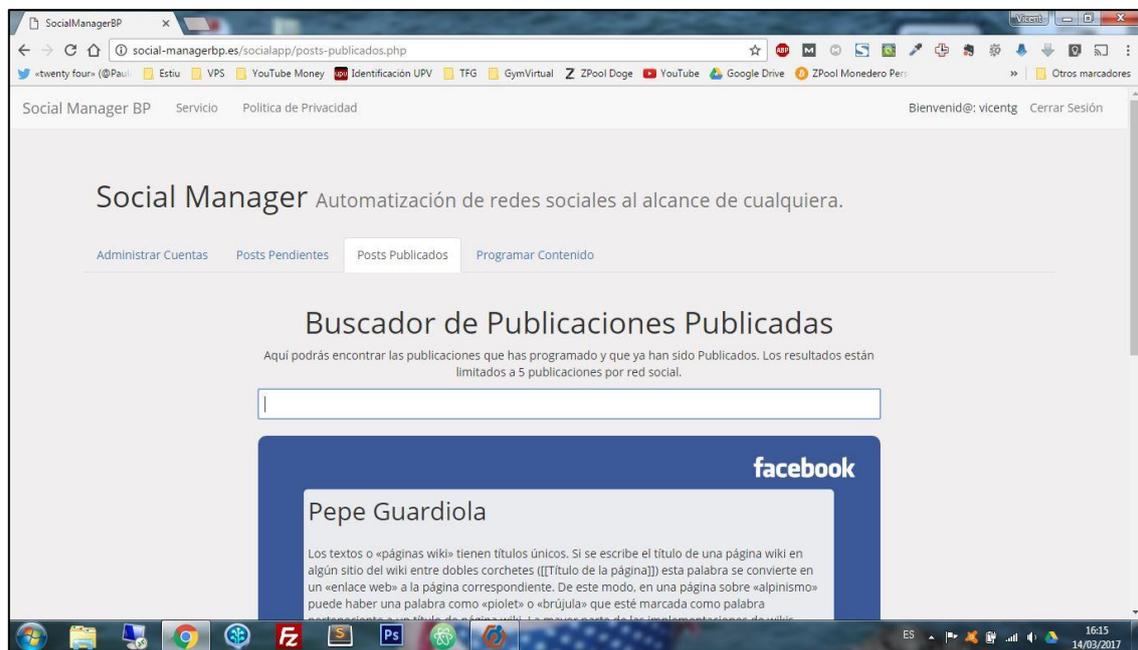


Figura 27 Prueba publicaciones publicadas con resolución 1366x768

6.2.3 Resolución de tableta 800x1280

Analizando ahora las capturas de la resolución 800x1280 de una tableta, podemos ver que el contenido se muestra correctamente y que sigue siendo usable. La principal diferencia es que ya no tenemos tanto margen por los lados.

En esta resolución, hemos optado por mostrar solo dos capturas, ya que quedan las dos juntas sin ocupar mucho espacio y así no nos extendemos demasiado.



Figura 28 Prueba programar contenido con resolución 800x1280



Figura 29 Prueba publicaciones pendientes con resolución 800x1280

6.2.4 Resolución de móvil 1080x1920

Observando las capturas hechas en un móvil con resolución 1080x1920, podemos ver que el contenido está un poco descolocado y que la aplicación web es un poco difícil de utilizar así. Esto se podría arreglar con unas líneas de código más, pero como ya hemos comentado, no es el objetivo de este proyecto.

Como podemos ver en la Figura 30, la programación de contenido se puede convertir en algo tedioso, ya que los botones están descolocados y los espacios para escribir son muy pequeños. En el caso de la Figura 31, que nos muestra la página de publicaciones ya publicadas, podemos ver que el contenido se muestra de forma adecuada.

Una vez más, al igual que en el punto anterior, hemos decidido insertar solo dos capturas para no ocupar demasiado espacio.



Figura 30 Prueba programar contenido con resolución 1080x1920



Figura 31 Prueba publicaciones publicadas con resolución 1080x1920

6.3 Pruebas de uso

A continuación vamos a mostrar, con capturas, el funcionamiento de las principales utilidades de nuestra aplicación web, que hemos desarrollado para este proyecto. Los casos de uso de iniciar sesión y cerrar sesión no los vamos a enseñar, ya que no son ninguna novedad en los proyectos web y nos ocuparía varias hojas.

Cabe señalar también, que no vamos a mostrar todas las funcionalidades con cada una de las tres redes sociales, ya que nos llevaría muchísimo tiempo y la memoria se alargaría demasiado.

6.3.1 Programación y publicación de contenido en Facebook

A continuación, vamos a mostrar como programamos una publicación para Facebook y después veremos cómo se publica a la hora adecuada.

Diseño e implementación de una aplicación web para la gestión integrada de redes sociales



Figura 32 Prueba programación y publicación de contenido en Facebook 1

Como podemos apreciar en la Figura 32, seleccionamos las opciones de esconder de Tumblr y Twitter, escribimos el contenido de la publicación y la cuenta, y finalmente, escogemos una fecha pulsando en el icono que nos abre el calendario. Posteriormente pulsamos el botón de programar.



Figura 33 Prueba programación y publicación de contenido en Facebook 2

Vamos a comprobar en la página de publicaciones pendientes que la publicación que acabamos de programar está pendiente de ser publicada. Tras esperar a que se haga la hora, vamos a Facebook y comprobamos que la publicación ha sido publicada adecuadamente a la hora que le hemos indicado.

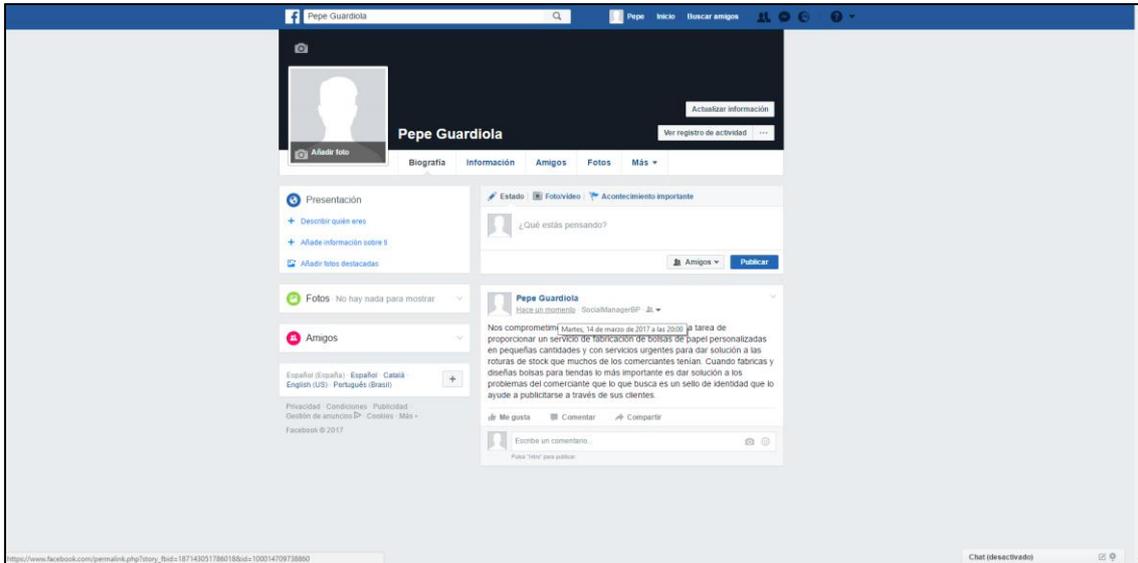


Figura 34 Prueba programación y publicación de contenido en Facebook 3

6.3.2 Buscar publicación de Twitter ya publicada y eliminar

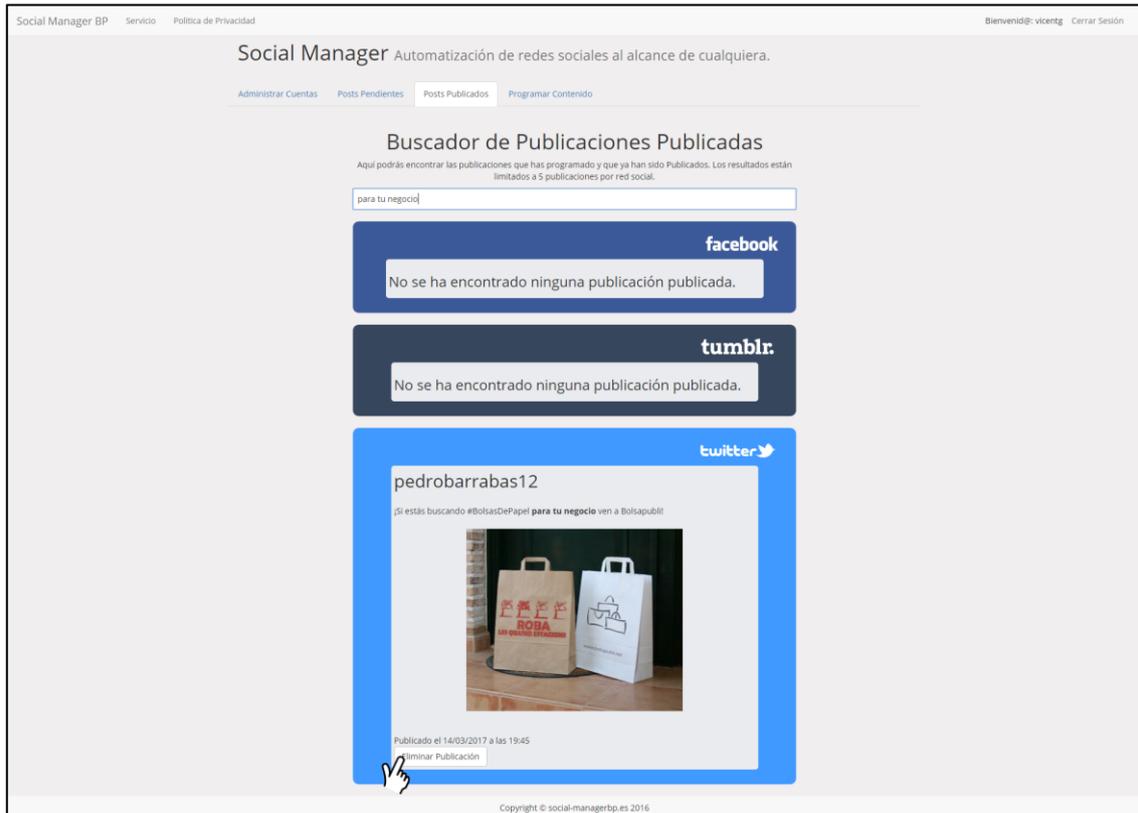


Figura 34 Prueba buscar publicación de Twitter ya publicada y eliminar 1



Diseño e implementación de una aplicación web para la gestión integrada de redes sociales

Como vemos en la Figura 34, escribimos las palabras a buscar en el buscador. Al encontrar la publicación que deseamos eliminar, seleccionamos la opción de eliminar publicación. En la siguiente figura podemos ver que la publicación ha sido eliminada.



Figura 35 Prueba buscar publicación de Twitter ya publicada y eliminar 2

En la Figura 36, ya no encontramos el Tweet que se había publicado anteriormente, la publicación ha sido eliminada con éxito de la red social.

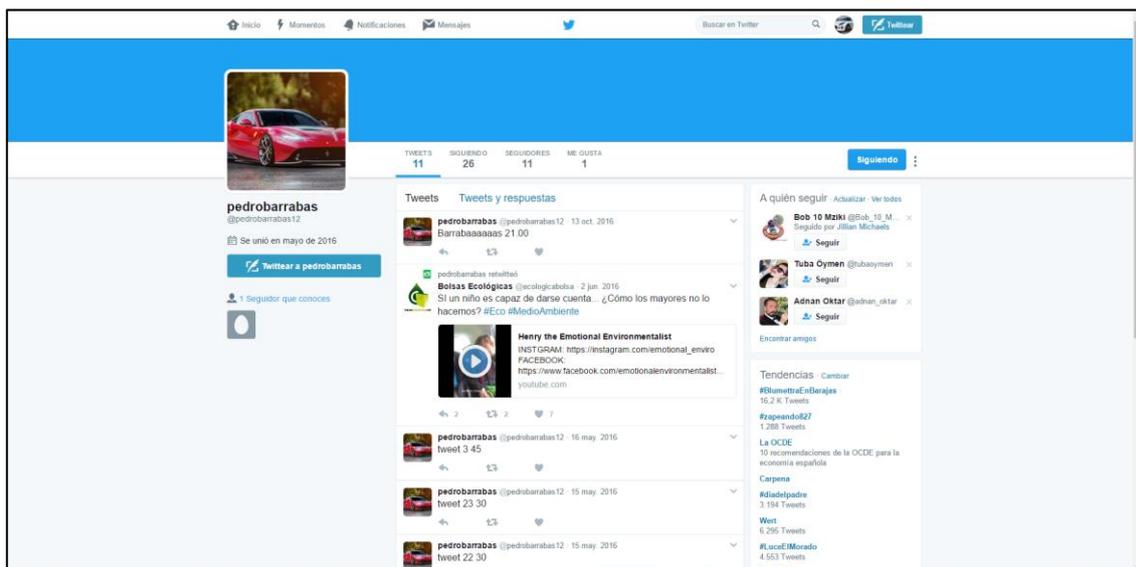


Figura 36 Prueba buscar publicación de Twitter ya publicada y eliminar 3

6.3.3 Añadir cuenta de Twitter

A continuación vamos a mostrar como añadimos una cuenta de Twitter a nuestra aplicación web. Como vemos en la Figura 37, seleccionamos la opción de añadir cuenta de Twitter.

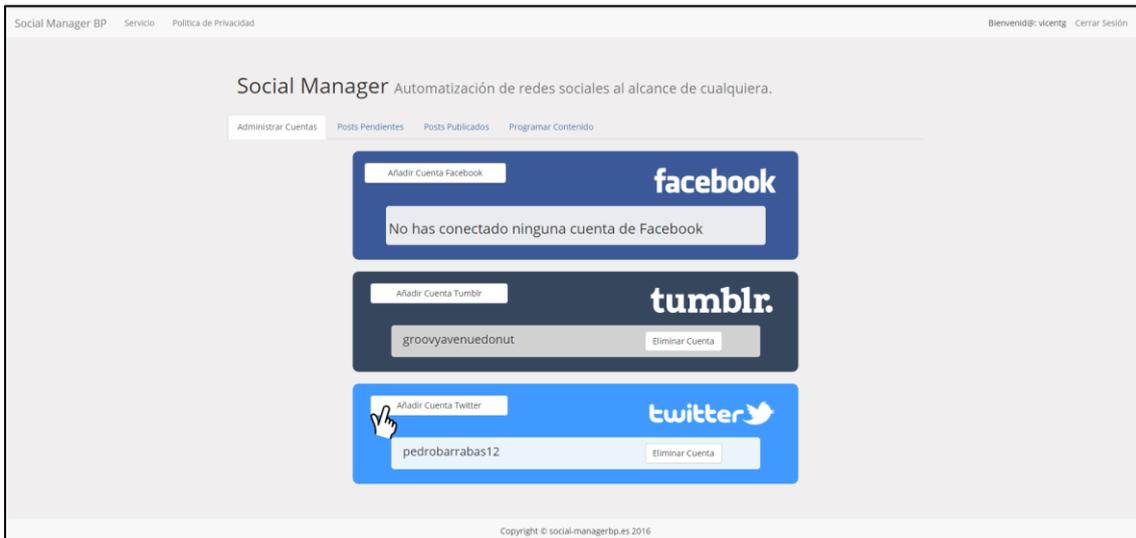


Figura 37 Prueba añadir cuenta de Twitter 1

Al pulsar el botón, la aplicación web nos redirige a Twitter para que autorizemos los permisos necesarios, así podrá publicar en nuestro nombre de forma automática.

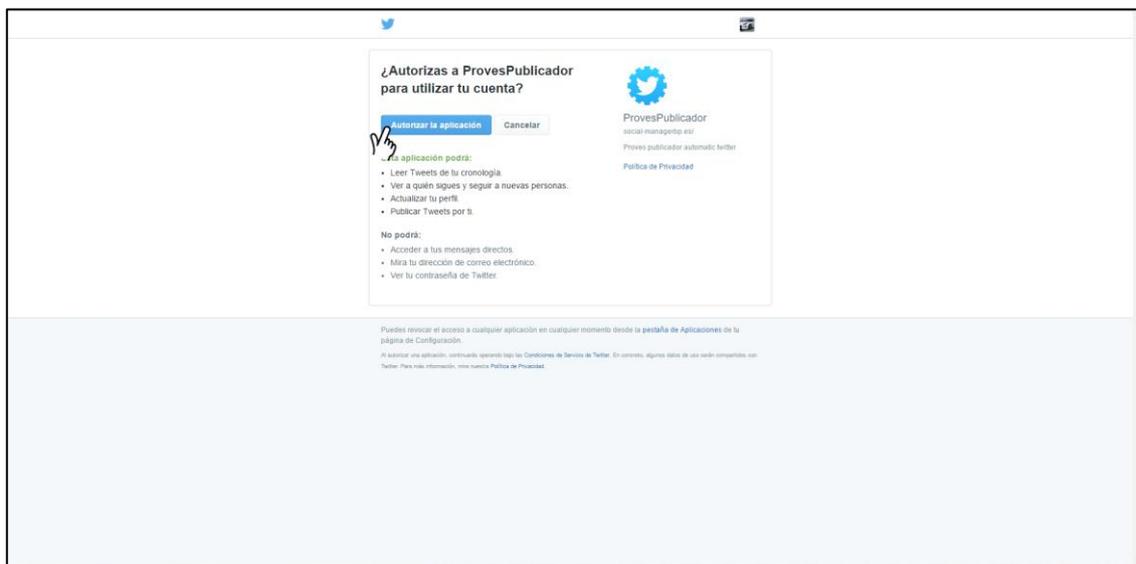


Figura 38 Prueba añadir cuenta de Twitter 2

En caso de que tengamos la sesión iniciada en Twitter, solo tendremos que seleccionar la opción de autorizar aplicación. Si no tuviéramos la cuenta iniciada, nos saldrían los campos necesarios para iniciar sesión y posteriormente podríamos pulsar el botón para autorizar las publicaciones automáticas.

Como veremos en la Figura 39, Twitter nos redirige a nuestra aplicación web y ya podemos ver que la cuenta de Twitter ha sido añadida.

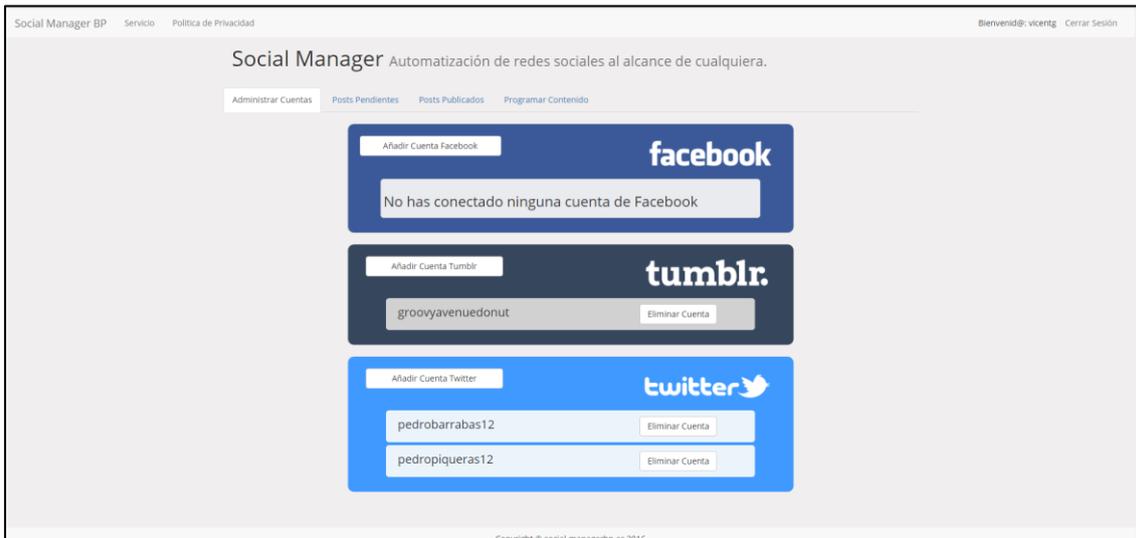


Figura 39 Prueba añadir cuenta de Twitter 3

6.3.4 Eliminar cuenta de Facebook

En este punto vamos a mostrar como eliminamos una cuenta de Facebook desde la página de administrar cuentas. Como veremos a continuación, la cuenta de Francisco Guardiola será eliminada, posteriormente se nos informará de que no disponemos de ninguna cuenta de Facebook conectada con la aplicación.

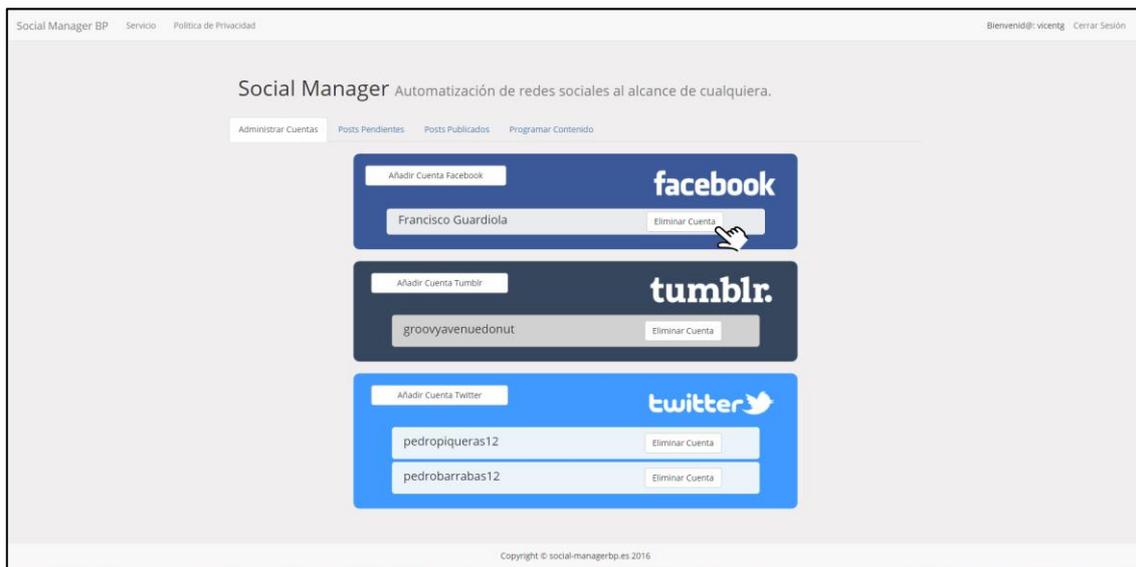


Figura 40 Prueba eliminar cuenta de Facebook 1

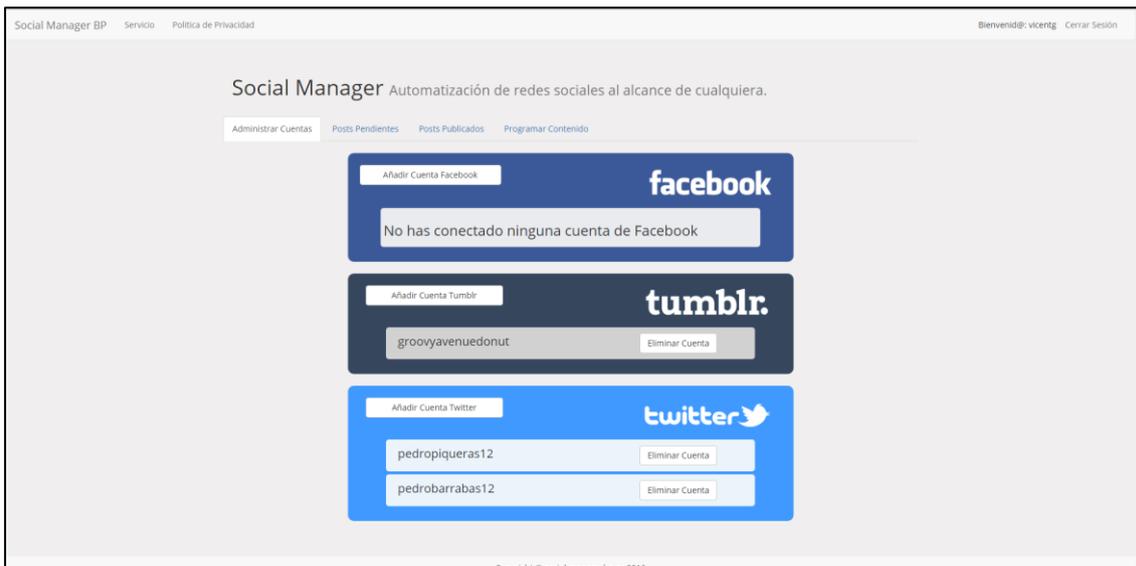


Figura 41 Prueba eliminar cuenta de Facebook 2

6.3.5 Modificar publicación pendiente de Tumblr

Vamos a mostrar cómo podemos modificar una publicación de Tumblr que aun no ha sido publicada en la página de publicaciones pendientes.

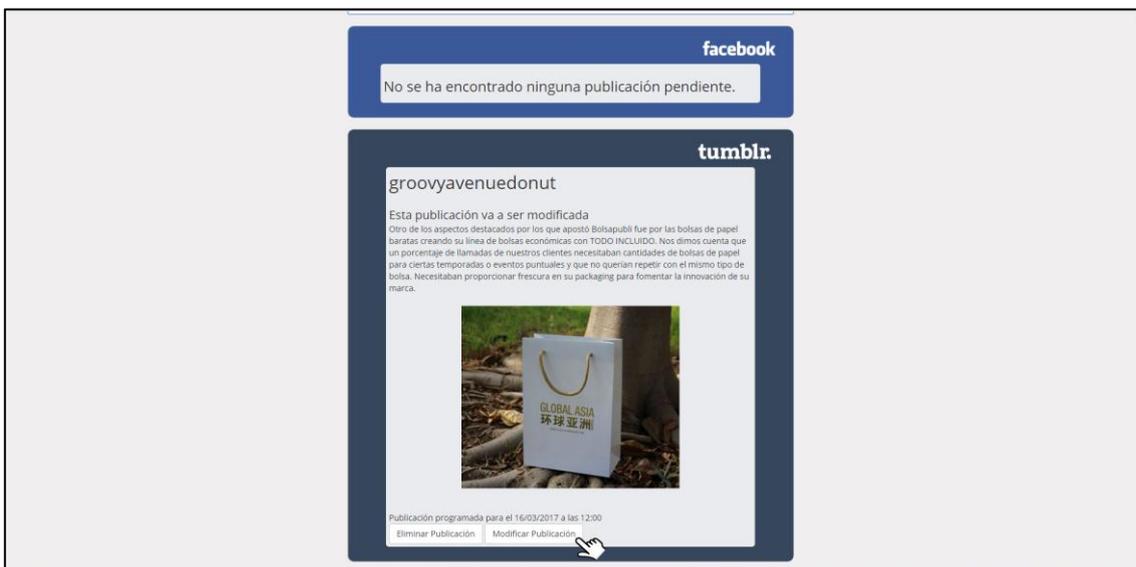


Figura 42 Prueba modificar publicación pendiente de Tumblr 1

Como podemos apreciar en la Figura 42, la publicación nos habla de las bolsas de papel y contiene una foto de una bolsa de papel de lujo. A continuación, al seleccionar la opción de modificar publicación, nos redirigirá a la página donde podremos modificar los campos que deseemos.

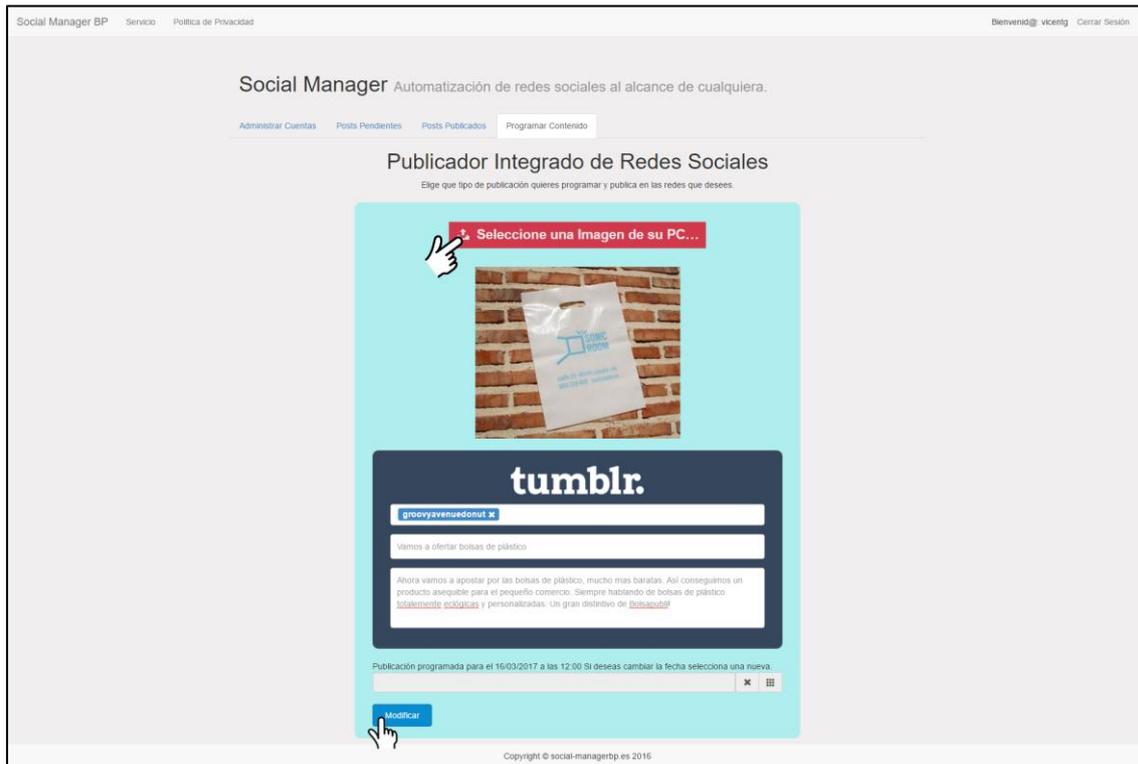


Figura 43 Prueba modificar publicación pendiente de Tumblr 2

En la Figura 43, vemos como cargamos una nueva imagen pulsando el botón correspondiente, posteriormente modificamos el contenido de la publicación y seleccionamos la opción modificar.



Figura 44 Prueba modificar publicación pendiente de Tumblr 3

Tras haber modificado la publicación, podemos ver en la página de publicaciones pendientes que el contenido de la publicación ha sido modificado. Ahora, la publicación se centra en las bolsas de plástico y no en las de papel como anteriormente.

6.3.6 Mostrar publicaciones pendientes y comprobar que se publican

En esta última prueba, vamos a ver las publicaciones que hay pendientes de publicar y posteriormente comprobaremos que se publican adecuadamente en las dos redes sociales.

The screenshot shows the 'Social Manager' web application interface. At the top, there are navigation links for 'Social Manager BP', 'Servicio', and 'Política de Privacidad'. The main header reads 'Social Manager Automatización de redes sociales al alcance de cualquiera.' Below this, there are tabs for 'Administrar Cuentas', 'Posts Pendientes', 'Posts Publicados', and 'Programar Contenido'. The 'Posts Pendientes' tab is active, displaying a search bar and a message: 'Buscador de Publicaciones Pendientes. Aquí podrás encontrar las publicaciones que están pendientes de publicar. Los resultados están limitados a 5 publicaciones por red social.' Below the search bar, there are three pending posts:

- facebook:** A message stating 'No se ha encontrado ninguna publicación pendiente.'
- tumblr:** A post from 'groovyavenuedonut' titled 'Bolsas de papel para todos!'. The text describes various paper bag models available in 'Bolsapubli', such as paper, cotton, and polypropylene. It includes an image of two paper bags, one brown and one white, both with the 'ROBA' logo. The post is scheduled for '14/03/2017 a las 19:45' and has 'Eliminar Publicación' and 'Modificar Publicación' buttons.
- twitter:** A post from 'pedrobarrabas12' with the text '¡si estás buscando #BolsasDePapel para tu negocio ven a Bolsapubli!'. It includes the same image of two paper bags as the Tumblr post. The post is also scheduled for '14/03/2017 a las 19:45' and has 'Eliminar Publicación' and 'Modificar Publicación' buttons.

At the bottom of the interface, there is a copyright notice: 'Copyright © social-managerbp.es 2016'.

Figura 45 Prueba mostrar publicaciones pendientes y comprobar que se publican 1

Como podemos ver en la Figura 45, tenemos dos publicaciones pendientes de ser publicadas en las respectivas redes sociales, una de Twitter y otra de Tumblr. En las dos siguientes figuras, veremos cómo las dos publicaciones han sido publicadas el día 14/03/2017 a las 19:45, como estaba programado.

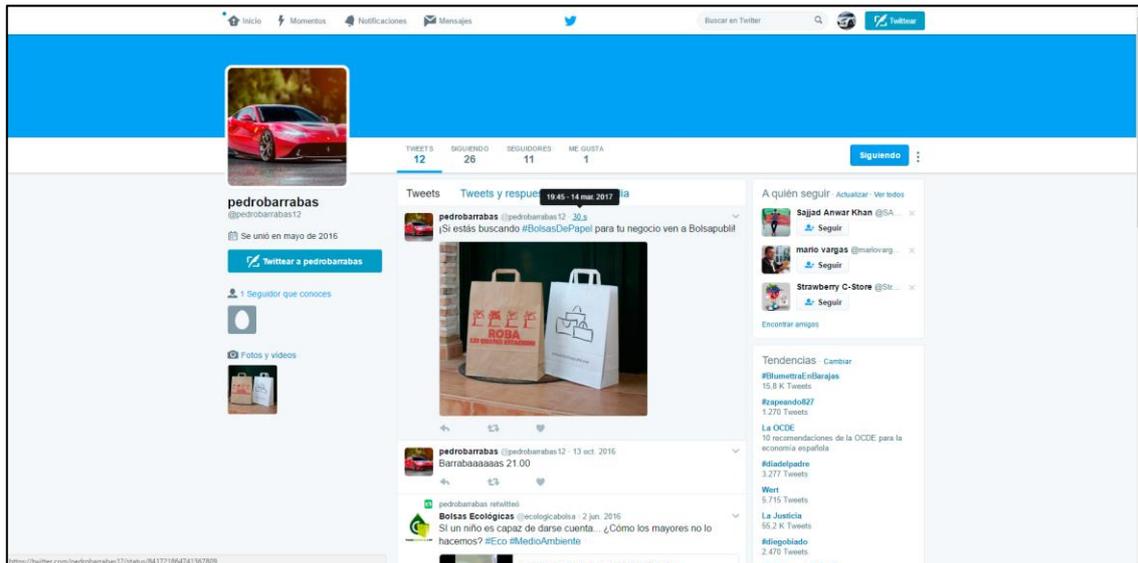


Figura 46 Prueba mostrar publicaciones pendientes y comprobar que se publican 2

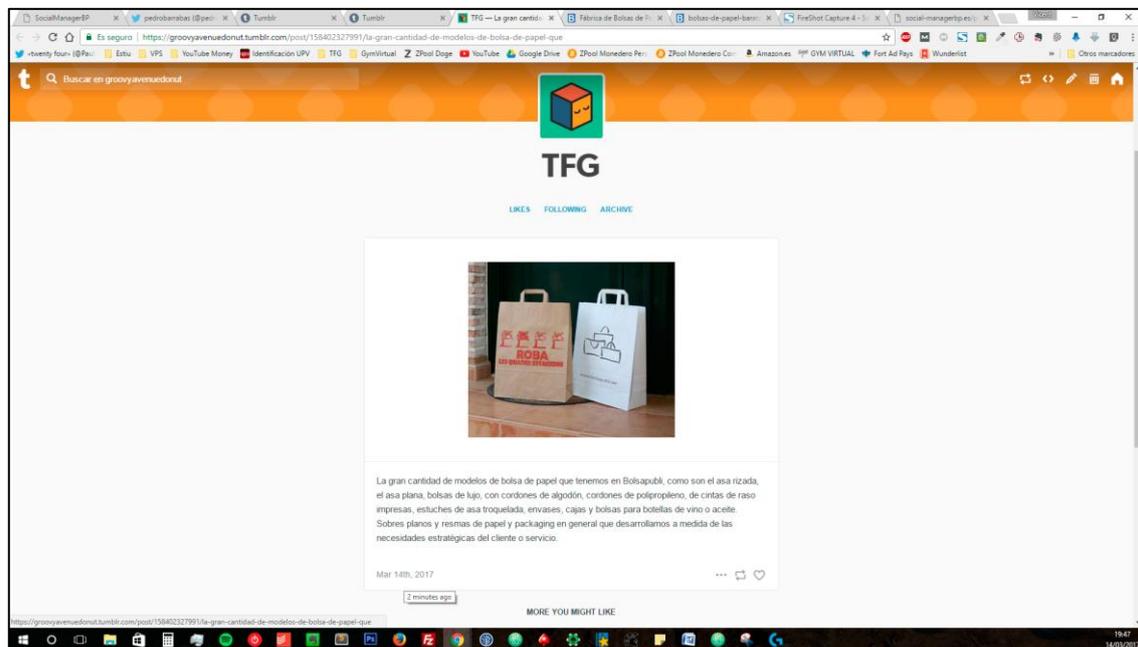


Figura 47 Prueba mostrar publicaciones pendientes y comprobar que se publican 3

En la Figura 47, hemos decidido dejar la barra de tareas para poder mostrar la hora. Tras intentar visualizar la hora de la publicación en la captura y no conseguirlo, hemos decidido utilizar como prueba la hora en la que hemos hecho la captura.

Teniendo en cuenta que el post de Tumblr se había publicado hacia dos minutos, la publicación se publicó a las 19:45 como estaba programado.

7. Conclusiones

Para finalizar la memoria de este Trabajo de Fin de Grado, vamos a exponer las conclusiones, que dividiremos en técnicas y personales.

7.1 Conclusiones técnicas

En este proyecto hemos aprendido a desarrollar una aplicación web con PHP, HTML5 y MySQL entre otros. Antes de empezar el proyecto, solo conocíamos únicamente las bases de esas tecnologías y nunca habíamos desarrollado nada con ellas.

Tras finalizar el proyecto, el nivel adquirido en las tecnologías utilizadas es enorme comparado con lo que sabíamos al principio, de hecho, algunas tecnologías solo las conocíamos de algunos compañeros que nos habían comentado para qué servían, como es el caso de AJAX o jQuery.

Al proyecto le han faltado algunos retoques para quedar perfecto, algún pequeño error aislado y la correcta visualización del contenido en resoluciones móviles. Estos pequeños retoques han quedado pendientes para futuras ampliaciones.

7.2 Conclusiones personales

Durante todo este tiempo que hemos estado desarrollando la aplicación web, hemos pasado por varias fases, hemos visto como poco a poco íbamos consiguiendo nuevas funcionalidades y tras horas de empeño conseguíamos que funcionara, esa es la mayor satisfacción que puede tener un desarrollador y es lo que nos animaba a seguir invirtiendo más horas.

Gracias al desarrollo de este proyecto, hemos sido conscientes de la capacidad que tenemos de aprender y de las oportunidades que la gente de hoy en día tiene para aprender cualquier cosa, gracias a internet.

No nos ha hecho falta invertir dinero, solo tiempo y muchas ganas de terminar el Grado de Ingeniería Informática, con un buen Trabajo de Fin de Grado, para abrirnos paso al mercado laboral con buen pie.

8. Bibliografía

- ANICAS, M (2014). *An Introduction to OAuth 2*.
<<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>>
[Consulta: 2 de marzo de 2017]
- CRUZ, A. *¿Cómo publicar un post en nuestro muro con la API Graph API de Facebook (PHP)?*. <<http://www.desarrollolibre.net/blog/tema/230/php/como-publicar-un-post-en-nuestro-muro-con-la-api-graph-api-de-facebook-php#.WMkWJDs1-Uk>>
[Consulta: 10 de enero de 2017]
- HAYERBEKE, M. (2014) *Eloquent JavaScript: a modern introduction to programming*. San Francisco: No Starch Press.
- LIBROSWEB. *Introducción a JavaScript*. <<http://librosweb.es/libro/javascript/>>
[Consulta: 10 de enero de 2017 – 2 de febrero de 2017]
- MARTÍNEZ, I. *Subir y visualizar imágenes con php y MySQL en 4 pasos*. <<http://www.formacionwebonline.com/guia-para-subir-y-visualizar-imagenes-con-php-y-mysql/>> [Consulta: 31 de enero de 2017]
- REFSNES DATA. *W3Schools Online Web Tutorials*.
<<https://www.w3schools.com/>> [Consulta: 12 de junio de 2016 – 3 de febrero de 2017]
- SUEHRING, S., VALADE, J. (2013) *PHP, MySQL, JavaScript & HTML5 All-in-one For Dummies*. United States: For Dummies.
- TECH STREAM. *File Upload With Ajax*. <<http://techstream.org/Web-Development/PHP/Simple-File-Upload-with-Ajax>> [Consulta: 3 de febrero de 2017]
- TECHSLIDES. *Tumblr API Example using OAuth and PHP*. <<http://techslides.com/tumblr-api-example-using-oauth-and-php>> [Consulta: 1 de febrero de 2017]
- THE PHP GROUP. *PHP: Manual de PHP – Manual*.
<<http://php.net/manual/es/>> [Consulta: 7 de junio de 2016 – 5 de febrero de 2017]
- THIS INTERESTS ME. *Simple Ajax Request Example With JQuery And PHP*. <<http://thisinterestsme.com/simple-ajax-request-example-jquery-php/>> [Consulta: 26 de enero de 2017]
- VALADE, J., BALLAD, T., BALLAD, B. (2008) *PHP & MySQL web development: All-in-one Desk Reference For Dummies*. United States: For Dummies.
- VARIOS AUTORES. *Bootstrap · The world's most popular mobile-first and responsive front-end framework*. <<http://getbootstrap.com/>> [Consulta: 7 de junio de 2016 – 24 de enero de 2017]

“Cap. 1 – Como hacer un buscador AJAX con jQuery, PHP, MySQL, CSS y HTML”. Youtube <<https://www.youtube.com/watch?v=jMp7ozbPgSs>> [Consulta: 14 de enero de 2017]

“Curso PHP MySql. Sistema de login I. Vídeo 59”. Youtube <<https://www.youtube.com/watch?v=BGY-SQtVzJU>> [Consulta: 25 de enero de 2017]

“Twitter API v1.1 Using PHP: Login and Get User Info - 1”. Youtube <<https://www.youtube.com/watch?v=t5hD96EYAtU&t>> [Consulta: 21 de junio de 2016 – 19 de octubre de 2016]

"Facebook PHP SDK v5: Getting Started - Part 1". Youtube <<https://www.youtube.com/watch?v=P9ktjGNeAzg&t>> [Consulta: 22 de diciembre de 2016 – 14 de febrero de 2016]