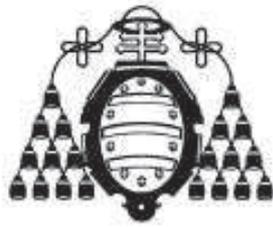


UNIVERSIDAD DE OVIEDO

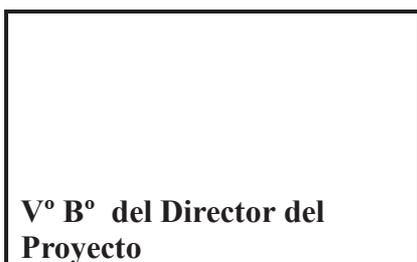


ESCUELA DE INGENIERÍA INFORMÁTICA

PROYECTO FIN DE MÁSTER

“APLICACIÓN WEB PARA LA GESTIÓN AGREGADA DE
DIFERENTES REDES SOCIALES Y APLICACIONES”

DIRECTOR: Darío Álvarez Gutiérrez



Vº Bº del Director del
Proyecto

AUTOR: David Cordon Blanco

Agradecimientos

A mi director Darío Álvarez por la ayuda constante recibida para llevar el proyecto adelante.

A la empresa Procesoiva por permitirme desarrollar el proyecto y hacerme sentir uno más.

A mis compañeros por los buenos momentos que hemos vivido.

A mi familia por el apoyo y los ánimos constantes.

A mis amigos por estar siempre cuando se les necesita.

A toda la gente que he conocido en estos 2 años.

GRACIAS.

Resumen

En este proyecto fin de Máster se desarrolla una aplicación web, desde ahora flypper.es, con la cual se puede gestionar tanto redes sociales como otras aplicaciones.

Actualmente están integrados los servicios de Twitter, Facebook, Youtube y Spotify.

A través de Flypper los usuarios podrán realizar funciones habituales de los anteriores servicios mencionados, tales como:

- Ver tu time line de Facebook.
- Compartir publicaciones en Facebook.
- Hacer me gusta en las publicaciones de Facebook de otros amigos.
- Ver el time line de Twitter.
- Twittear.
- Hacer retweets.
- Buscar videos en Youtube.
- Reproducir vídeos de Youtube.
- Buscar canciones en Spotify.
- Reproducir canciones de Spotify.
- Etc.

Flypper tiene la gran ventaja de realizar todas las funciones descritas anteriormente desde una única pestaña del navegador.

Además, la aplicación cuenta con una interfaz diseñada por módulos, al estilo iGoogle, que permite añadirlos, eliminarlos o arrastrarlos al lugar que mejor se adapte a los gustos de cada persona.

Actualmente, flypper.es es una prueba beta para usuarios, con el fin de poder presentarlo ante posibles inversores para conseguir capital de riesgo y llevar la idea a cabo.

Los lenguajes de programación de los módulos utilizados han sido Html, Css y Php, enriquecido con AJAX y Jquery. Estos módulos han sido montados sobre el CMS Joomla.

Palabras Clave

Redes sociales, aplicaciones, panel de control, jquery, ajax

Índice General

CAPÍTULO 1. MEMORIA DEL PROYECTO.....	17
1.1 RESUMEN DE LA MOTIVACIÓN, OBJETIVOS Y ALCANCE DEL PROYECTO	17
1.2 ESTRUCTURA DEL DOCUMENTO	18
CAPÍTULO 2. INTRODUCCIÓN.....	21
2.1 JUSTIFICACIÓN DEL PROYECTO.....	21
2.2 OBJETIVOS DEL PROYECTO	23
2.3 ESTUDIO DE LA SITUACIÓN ACTUAL	24
2.3.1 <i>Evaluación de Alternativas</i>	24
CAPÍTULO 3. ASPECTOS TEÓRICOS.....	27
3.1 SOCIAL MEDIA.....	27
3.2 OAUTH.....	28
3.3 HTML5.....	29
3.4 CSS3.....	30
3.5 PHP.....	31
3.6 MYSQL.....	32
3.7 AJAX.....	33
3.8 JQUERY.....	35
3.9 NETBEANS.....	36
3.10 JOOMLA.....	37
CAPÍTULO 4. PLANIFICACIÓN DEL PROYECTO Y RESUMEN DE PRESUPUESTOS.....	39
4.1 PLANIFICACIÓN	39
4.2 RESUMEN DEL PRESUPUESTO.....	42
CAPÍTULO 5. ANÁLISIS	43
5.1 DEFINICIÓN DEL SISTEMA.....	43
5.1.1 <i>Determinación del Alcance del Sistema</i>	43
5.2 REQUISITOS DEL SISTEMA.....	44
5.2.1 <i>Obtención de los Requisitos del Sistema</i>	44
5.2.2 <i>Identificación de Actores del Sistema</i>	46
5.2.3 <i>Especificación de Casos de Uso</i>	47
5.3 IDENTIFICACIÓN DE LOS SUBSISTEMAS EN LA FASE DE ANÁLISIS	55
5.3.1 <i>Descripción de los Subsistemas</i>	55
5.3.2 <i>Descripción de los Interfaces entre Subsistemas</i>	56
5.4 DIAGRAMA DE CLASES PRELIMINAR DEL ANÁLISIS	57
5.4.1 <i>Diagrama de Clases</i>	57
5.4.2 <i>Descripción de las Clases</i>	58
5.5 ANÁLISIS DE CASOS DE USO Y ESCENARIOS	64
5.5.1 <i>Obtener permisos Facebook</i>	64
5.5.2 <i>Ver inicio de Facebook</i>	65
5.5.3 <i>Ver perfil de Facebook</i>	65
5.5.4 <i>Hacer “me gusta”</i>	66
5.5.5 <i>Ver publicación</i>	66

5.5.6	Publicar.....	66
5.5.7	Comentar publicación.....	67
5.5.8	Borrar publicación.....	67
5.5.9	Cerrar sesión Facebook.....	68
5.5.10	Obtener permisos Twitter.....	68
5.5.11	Ver inicio de Twitter.....	69
5.5.12	Ver perfil de Twitter.....	69
5.5.13	Ver menciones.....	69
5.5.14	Twitear.....	70
5.5.15	Retwitear.....	70
5.5.16	Borrar tweet.....	70
5.5.17	Cerrar sesión Twitter.....	70
5.5.18	Ver videos recomendados.....	71
5.5.19	Ver videos más vistos.....	71
5.5.20	Ver videos más comentados.....	71
5.5.21	Buscar videos.....	72
5.5.22	Reproducir videos.....	72
5.5.23	Buscar canción.....	73
5.5.24	Ver portada del álbum de la canción.....	73
5.5.25	Reproducir canción.....	73
5.5.26	Compartir en Facebook.....	74
5.5.27	Compartir en Twitter.....	74
5.6	ANÁLISIS DE INTERFACES DE USUARIO.....	75
5.6.1	Descripción de la Interfaz.....	75
5.6.2	Descripción del Comportamiento de la Interfaz.....	76
5.6.3	Diagrama de Navegabilidad.....	77
5.7	ESPECIFICACIÓN DEL PLAN DE PRUEBAS.....	78
CAPÍTULO 6. DISEÑO DEL SISTEMA.....		83
6.1	ARQUITECTURA DEL SISTEMA.....	83
6.1.1	Diagramas de Paquetes.....	83
6.1.2	Diagramas de Componentes.....	86
6.1.3	Diagramas de Despliegue.....	87
6.2	DISEÑO DE CLASES.....	88
6.2.1	Diagrama de Clases.....	88
6.3	DIAGRAMAS DE INTERACCIÓN Y ESTADOS.....	89
6.3.1	Obtener permisos Facebook.....	89
6.3.2	Ver inicio de Facebook.....	90
6.3.3	Ver perfil de Facebook.....	90
6.3.4	Hacer “me gusta”.....	91
6.3.5	Ver publicación.....	91
6.3.6	Publicar.....	91
6.3.7	Comentar publicación.....	92
6.3.8	Borrar publicación.....	92
6.3.9	Cerrar sesión Facebook.....	92
6.3.10	Obtener permisos Twitter.....	92
6.3.11	Ver inicio de Twitter.....	93
6.3.12	Ver perfil de Twitter.....	93
6.3.13	Ver menciones.....	93
6.3.14	Twitear.....	93

6.3.15	<i>Retwitear</i>	93
6.3.16	<i>Borrar tweet</i>	93
6.3.17	<i>Cerrar sesión Twitter</i>	93
6.3.18	<i>Ver videos recomendados</i>	93
6.3.19	<i>Ver videos más vistos</i>	94
6.3.20	<i>Ver videos más comentados</i>	94
6.3.21	<i>Buscar videos</i>	94
6.3.22	<i>Buscar canción</i>	94
6.4	DISEÑO DE LA BASE DE DATOS	95
6.4.1	<i>Descripción del SGBD Usado</i>	95
6.4.2	<i>Integración del SGBD en Nuestro Sistema</i>	95
6.4.3	<i>Diagrama E-R</i>	96
6.5	DISEÑO DE LA INTERFAZ.....	98
6.5.1	<i>Pantalla principal</i>	98
6.5.2	<i>Pantalla de login</i>	99
6.5.3	<i>Pantalla Flypper</i>	100
6.6	ESPECIFICACIÓN TÉCNICA DEL PLAN DE PRUEBAS	101
6.6.1	<i>Pruebas Unitarias</i>	101
6.6.2	<i>Pruebas de Usabilidad y Accesibilidad</i>	101
6.6.3	<i>Pruebas de Rendimiento</i>	104
CAPÍTULO 7. IMPLEMENTACIÓN DEL SISTEMA		105
7.1	ESTÁNDARES Y NORMAS SEGUIDOS	105
7.1.1	<i>CSS 3</i>	105
7.1.2	<i>XHTML 1.0 Transitional</i>	105
7.2	LENGUAJES DE PROGRAMACIÓN	107
7.2.1	<i>HTML</i>	107
7.2.2	<i>CSS</i>	107
7.2.3	<i>PHP</i>	107
7.2.4	<i>JavaScript</i>	107
7.2.5	<i>jQuery</i>	107
7.2.6	<i>AJAX</i>	107
7.2.7	<i>cURL</i>	108
7.2.8	<i>SDK Facebook para PHP</i>	108
7.2.9	<i>Librería OAuth Twitter para PHP</i>	108
7.3	HERRAMIENTAS Y PROGRAMAS USADOS PARA EL DESARROLLO	109
7.3.1	<i>NetBeans</i>	109
7.3.2	<i>Enterprise Architect 7</i>	109
7.3.3	<i>Microsoft Office 2007</i>	109
7.3.4	<i>HeidiSQL</i>	109
7.3.5	<i>Sublime Text 2</i>	109
7.3.6	<i>Google Chrome</i>	109
7.3.7	<i>FileZilla</i>	110
7.4	CREACIÓN DEL SISTEMA.....	111
7.4.1	<i>Descripción Detallada de las Clases</i>	111
7.4.2	<i>Facebook</i>	111
7.4.3	<i>Twitter</i>	112
7.4.4	<i>Youtube</i>	114
7.4.5	<i>Spotify</i>	115
CAPÍTULO 8. DESARROLLO DE LAS PRUEBAS.....		117

8.1	PRUEBAS UNITARIAS	117
8.2	PRUEBAS DE USABILIDAD Y ACCESIBILIDAD.....	125
8.2.1	<i>Pruebas de Usabilidad</i>	125
8.2.2	<i>Pruebas de Accesibilidad</i>	133
8.3	PRUEBAS DE RENDIMIENTO.....	140
CAPÍTULO 9. MANUALES DEL SISTEMA.....		141
9.1	MANUAL DE INSTALACIÓN Y EJECUCIÓN	141
9.2	MANUAL DE USUARIO	149
9.2.1	<i>Usuario anónimo</i>	149
9.2.2	<i>Usuario registrado</i>	150
9.3	MANUAL DEL PROGRAMADOR.....	167
9.3.1	<i>Facebook</i>	167
9.3.2	<i>Twitter</i>	167
9.3.3	<i>Youtube</i>	168
9.3.4	<i>Spotify</i>	169
9.3.5	<i>Nuevos módulos</i>	169
CAPÍTULO 10. CONCLUSIONES Y AMPLIACIONES		171
10.1	CONCLUSIONES.....	171
10.2	AMPLIACIONES	171
10.2.1	<i>Creación de nuevos módulos</i>	171
10.2.2	<i>Crear un login para registrarse con redes sociales</i>	172
10.2.3	<i>Crear un acortador de URL</i>	172
10.2.4	<i>Crear un sistema para publicar el mismo contenido en varias redes sociales</i>	172
10.2.5	<i>Programar la aplicación para prescindir de Joomla</i>	172
10.2.6	<i>Cumplir al menos accesibilidad AA</i>	172
10.2.7	<i>Aplicación móvil</i>	172
CAPÍTULO 11. PRESUPUESTO		173
CAPÍTULO 12. REFERENCIAS BIBLIOGRÁFICAS.....		175
12.1	REFERENCIAS EN INTERNET	175
CAPÍTULO 13. APÉNDICES		177
13.1	GLOSARIO Y DICCIONARIO DE DATOS	177
13.2	CONTENIDO ENTREGADO EN EL CD-ROM.....	178
13.2.1	<i>Contenidos</i>	178
13.2.2	<i>Código Ejecutable e Instalación</i>	179
13.3	ÍNDICE ALFABÉTICO.....	180
13.4	CÓDIGO FUENTE	181
13.4.1	<i>Paquete Facebook</i>	181
13.4.2	<i>Paquete Twitter</i>	191
13.4.3	<i>Paquete Youtube</i>	196
13.4.4	<i>Paquete Spotify</i>	201

Índice de Figuras

Figura 2.1. Situación actual de la web	21
Figura 2.2. Navegador web un día cualquiera.....	22
Figura 2.3. Flypper.....	22
Figura 2.4. Hootsuite	24
Figura 2.5. Tweetdeck	25
Figura 2.6. iGoogle	26
Figura 3.1. Social Media	27
Figura 3.2. OAuth	28
Figura 3.3. AJAX	34
Figura 4.1. Diagrama de Gantt de la fase de análisis	39
Figura 4.2 Diagrama de Gantt de la fase de diseño.....	40
Figura 4.3 Diagrama de Gantt de la fase de implementación y pruebas	40
Figura 4.4. Diagrama de Gantt de la fase de documentación y presentación del proyecto	41
Figura 5.1. Casos de uso relacionados por la autenticación llevada a cabo por Joomla.	47
Figura 5.2. Casos de uso sobre el módulo de Facebook.....	48
Figura 5.3. Casos de uso sobre el módulo de Twitter.....	50
Figura 5.4. Casos de uso sobre el módulo de Youtube.....	52
Figura 5.5. Casos de uso sobre el módulo de Spotify	53
Figura 5.6. Diagrama de clases	57
Figura 5.7. Módulo Facebook	58
Figura 5.8. Módulo Twitter.....	60
Figura 5.9. Módulo Youtube.....	62
Figura 5.10. Módulo Spotify	63
Figura 5.11. Diagrama de robustez: Obtener permisos Facebook	64
Figura 5.12. Boceto de la interfaz de inicio	75
Figura 5.13. Boceto de la interfaz de la pantalla del usuario registrado.....	76
Figura 5.14. Diagrama de navegabilidad.....	77
Figura 6.1. Diagrama de paquetes.....	83
Figura 6.2. Paquete Facebook	84
Figura 6.3. Paquete Twitter	84
Figura 6.4. Paquete Youtube	85
Figura 6.5. Paquete Spotify	85
Figura 6.6. Diagrama de componentes	86
Figura 6.7. Diagrama de despliegue.....	87
Figura 6.8. Diagrama de clases	88
Figura 6.9. Diagrama de secuencia para obtener los permisos de Facebook.....	89
Figura 6.10. Diagrama de secuencia para ver el inicio de Facebook.....	90
Figura 6.11. Diagrama de secuencia para hacer “me gusta”	91
Figura 6.12. Diagrama de secuencia para publicar en Facebook.....	91
Figura 6.13. Diagrama de secuencia para borrar una publicación de Facebook	92
Figura 6.14. Diagrama de secuencia para cerrar sesión de Facebook.....	92
Figura 6.15. Diagrama ER Joomla	96
Figura 6.16. Tablas del componente Drag&Drop	97
Figura 6.17. Pantalla principal Flypper	98
Figura 6.18. Pantalla de login	99

Figura 6.19. Pantalla de Flypper.....	100
Figura 7.1. CSS válido.....	105
Figura 7.2. HTML válido.....	106
Figura 8.1. Página principal sin imágenes.....	134
Figura 8.2. Panel de módulos de Flypper sin imágenes.....	135
Figura 8.3. Página principal cambiando tamaño del texto.....	135
Figura 8.4. Panel de módulos de Flypper cambiando el texto.....	136
Figura 8.5. XHTML Transitional válido.....	137
Figura 8.6. CSS3 válido.....	137
Figura 8.7. Prueba de rendimiento.....	140
Figura 9.1. Instalación Joomla paso 1.....	141
Figura 9.2. Instalación Joomla paso 2.....	142
Figura 9.3. Instalación Joomla paso 3.....	142
Figura 9.4. Instalación Joomla paso 4.....	143
Figura 9.5. Instalación Joomla paso 5.....	143
Figura 9.6. Instalación Joomla paso 6.....	144
Figura 9.7. Instalación Joomla paso 7.....	144
Figura 9.8. Instalación Joomla completada.....	145
Figura 9.9. Instalación Drag&Drop.....	145
Figura 9.10. Archivos FTP.....	146
Figura 9.11. Configuración Drag&Drop.....	146
Figura 9.12. Creación módulo Facebook.....	147
Figura 9.13. Configuración del módulo Facebook.....	147
Figura 9.14. Añadir módulos al componente Drag&Drop.....	148
Figura 9.15. Configuración módulo Facebook en Componente Drag&Drop.....	148
Figura 9.16. Pantalla inicial de Flypper.....	149
Figura 9.17. Pantalla de registro en Flypper.....	150
Figura 9.18. Pantalla de login.....	150
Figura 9.19. Pantalla principal de Flypper.....	151
Figura 9.20. Acción de arrastrar módulos.....	152
Figura 9.21. Acción de minimizar un módulo.....	152
Figura 9.22. Acción de cerrar un módulo.....	152
Figura 9.23. Acción de añadir un módulo.....	153
Figura 9.24. Pantalla iniciar sesión en Facebook.....	154
Figura 9.25. Pantalla login Facebook.....	154
Figura 9.26. Ir a la aplicación de Flypper en Facebook.....	155
Figura 9.27. Concesión de permisos a Flypper en Facebook.....	155
Figura 9.28. Pantalla de inicio de Facebook.....	156
Figura 9.29. Pantalla de perfil de Facebook.....	156
Figura 9.30. Pantalla comentar publicación.....	157
Figura 9.31. Pantalla iniciar sesión en Twitter.....	158
Figura 9.32. Pantalla login Twitter.....	158
Figura 9.33. Concesión de permisos a Flypper en Twitter.....	159
Figura 9.34. Pantalla de inicio de Twitter.....	160
Figura 9.35. Pantalla de perfil de Twitter.....	160
Figura 9.36. Pantalla de menciones en Twitter.....	161
Figura 9.37. Videos recomendados por Youtube.....	162
Figura 9.38. Videos más vistos en Youtube España.....	162
Figura 9.39. Videos trending en Youtube.....	163
Figura 9.40. Resultados de la búsqueda en Youtube.....	164

Figura 9.41. Reproducción de un video en Youtube.....	164
Figura 9.42. Resultado de una búsqueda en Spotify	165
Figura 9.43. Reproductor de Spotify	166

Capítulo 1. Memoria del Proyecto

1.1 Resumen de la Motivación, Objetivos y Alcance del Proyecto

Actualmente existen muchas aplicaciones en la web. En ellas se nos presentan grandes cantidades de información que no somos capaces de asimilar. Además estas aplicaciones están en constante crecimiento.

Diariamente, cada vez que accedemos a internet, tenemos que abrir un sinnúmero de páginas para acceder a los contenidos que deseamos.

Este proyecto nace con la necesidad personal de tener una aplicación donde concentre toda esa información a la que normalmente se accede, para así leer únicamente la información que me interesa además de no tener que abrir infinidad de páginas web.

El proyecto será llamado Flypper y en su fase inicial contendrá, entre sus módulos disponibles, 4 de las aplicaciones más utilizadas de internet como son Facebook, Twitter, Youtube y Spotify.

1.2 Estructura del documento

En el siguiente apartado describiremos cada uno de los capítulos en los que el documento está dividido.

- **Introducción**

En este primer apartado se describe brevemente el motivo por el que se va a proceder al desarrollo de la aplicación y sus objetivos.

- **Aspectos teóricos**

En este apartado realizaremos una descripción de los aspectos teóricos del proyecto, así como la situación actual de la competencia en el mercado del mismo.

- **Planificación del proyecto y resumen del presupuesto**

Esta sección describe la planificación seguida para la realización del proyecto.

- **Análisis**

En esta sección se construirá un modelo teórico del sistema, llamado modelo de análisis, en donde se enumeran y describen cada uno de los requerimientos del sistema de software a desarrollar.

- **Diseño del sistema**

En esta sección, en base al modelo de análisis, se construye un modelo de diseño que contiene la arquitectura del sistema del software y su diseño más detallado.

- **Implementación del sistema**

En esta sección, tomando como punto de partida el modelo de la fase anterior, se procede a describir los lenguajes y herramientas utilizadas para programar o implementar los diseños especificados en el modelo de diseño.

- **Desarrollo de las pruebas**

En esta sección se hace un estudio de los resultados de las pruebas realizadas en el proyecto.

- **Manuales del sistema**

Esta sección incluye los manuales de instalación, ejecución, de usuario y del programador.

- **Conclusiones y ampliaciones**

En este apartado se describen las conclusiones obtenidas una vez concluido el proyecto y las posibles ampliaciones del mismo.

- **Presupuesto**

En este apartado se realiza una tabla donde se muestra el presupuesto del proyecto.

- **Referencias bibliográficas**

En este apartado se incluyen todas las referencias utilizadas para el desarrollo del proyecto.

- **Apéndice**

Este apartado contiene el contenido de la aplicación entregada y el código fuente de la misma.

Capítulo 2. Introducción

2.1 Justificación del Proyecto

En la actualidad las opciones en internet son muy numerosas. Cada día aparecen nuevas aplicaciones con las cuales podemos realizar todo tipo de acciones, desde disfrutar de nuestro ocio oyendo música o viendo videos, hasta compartir nuestras vidas en internet mediante las redes sociales. El día a día en internet es un rompecabezas que hay que resolver.



Figura 2.1. Situación actual de la web

Cada vez que decido navegar por internet la tarea se convierte en algo tediosa. En primer lugar debo elegir, entre las infinitas aplicaciones, las que deseo consultar en ese momento. Una vez tomada la decisión tengo que abrir una página por cada web que deseo visitar. Después se presenta el problema de que, para cada una de esas aplicaciones, es necesario introducir un usuario y una contraseña. Esto conlleva a que hay que recordar las credenciales para poder acceder a la información o a utilizar los servicios, con el consiguiente problema de que se te olvide alguna que no usas muy asiduamente. Una vez que he accedido a los servicios, surge otro problema. En cada página tienes una aplicación diferente y no puedes ver un video musical y leer tu Facebook, o realizar cualquier otra actividad al mismo tiempo. Definitivamente nuestro navegador se convierte en un caos de pestañas.



Figura 2.2. Navegador web un día cualquiera

Resumiendo, la actualidad diaria a la hora de consultar la web es algo complicada. Se nos presentan problemas tales como:

- Muchas cuentas de usuario.
- Diferentes usuarios y contraseñas para acceder a las webs.
- Una página abierta por cada web que visitamos.
- Demasiada información para asimilar.

Flypper nace con la finalidad de resolver todos estos problemas. Se trata de desarrollar una aplicación donde puedas elegir entre tus aplicaciones favoritas. Estas aplicaciones serán visualizadas en una misma pestaña del navegador. Además para acceder a Flypper solamente será necesario recordar un único usuario y contraseña.

Una vez desarrollada la aplicación los usuarios podrán acceder a su información preferida rápidamente, desde una única ventana y sin tener que recordar un montón de contraseñas.

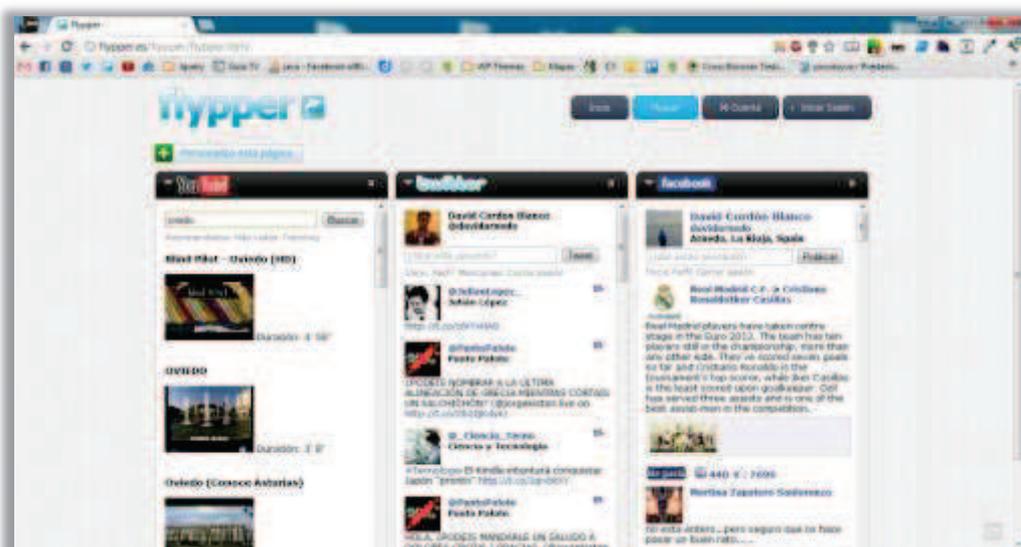


Figura 2.3. Flypper

2.2 Objetivos del Proyecto

El objetivo del proyecto consiste en desarrollar Flypper, una aplicación donde los usuarios puedan visualizar toda su información y aplicaciones favoritas, desde una única pestaña del navegador, a través de sus módulos.

Para este proyecto se han seleccionado 4 de las aplicaciones que los usuarios más usan en internet para realizar el desarrollo de los módulos que van a funcionar inicialmente dentro de Flypper. Estas son:

- Facebook
- Twitter
- Youtube
- Spotify

A continuación se muestran enumeradas algunas de las características principales de cada módulo:

- Facebook
 - Ver toda la información del usuario en Facebook.
 - Compartir publicaciones.
 - Hacer “me gusta” en otras publicaciones.
 - Comentar las publicaciones de tus amigos en Facebook.
- Twitter
 - Ver toda la información del usuario en Twitter.
 - Compartir tweets.
 - Retwitear otros tweets.
- Youtube
 - Buscar videos en Youtube.
 - Reproducir los videos de Youtube.
- Spotify
 - Buscar canciones en Spotify.
 - Reproducir las canciones de Spotify.

2.3 Estudio de la Situación Actual

2.3.1 Evaluación de Alternativas

2.3.1.1 Hootsuite

Hootsuite [**hootsuite**] nace en diciembre de 2008. Se trata tanto de una aplicación para web como para móvil (iPad, iPhone, iPod Touch, BlackBerry y Android). Fue diseñada para gestionar redes sociales por parte de personas u organizaciones.

HootSuite permite utilizar, las siguientes redes sociales: Facebook, Twitter, LinkedIn, Foursquare, MySpace y WordPress.

Frente a otros clientes similares, sus ventajas más destacadas son:

- No requiere la instalación de ningún programa, sino que se utiliza directamente a través del navegador de Internet
- Gestión colaborativa (distintos miembros de equipo en una misma cuenta, asignación de mensajes...).
- Envío de mensajes a varias redes simultáneamente.
- Acortador de URLs.
- Actualizaciones en tiempo real.
- Organización de contenidos a través de pestañas y columnas.
- Posibilidad de añadir feeds RSS.
- Uso de informes avanzados con integración de Google Analytics y Facebook Insights.

Uno de los principales inconvenientes es que no permite añadir otro tipo de aplicaciones que no sean redes sociales. Además, esta aplicación solamente puede usarse gratuitamente hasta 5 redes sociales, a partir de ahí tiene un coste mensual.

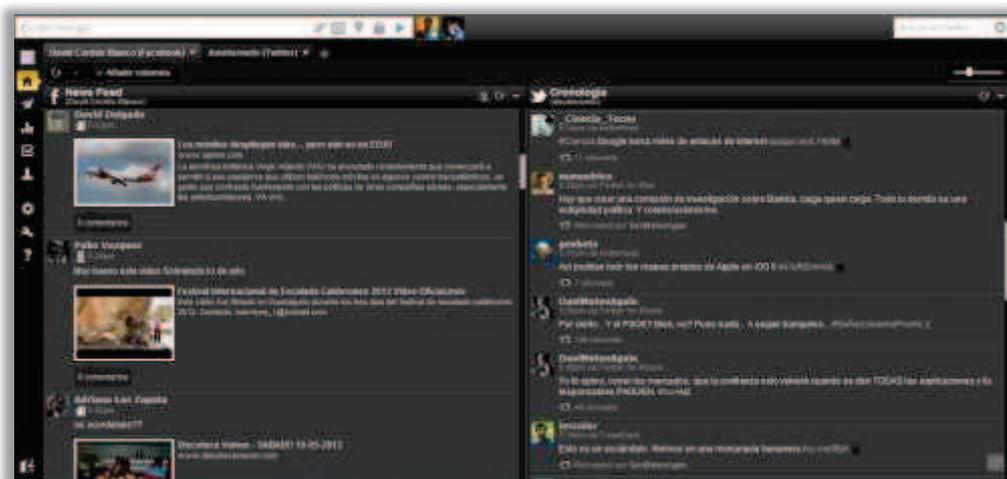


Figura 2.4. Hootsuite

2.3.1.2 Tweetdeck

Tweetdeck [**tweetdeck**] hace su lanzamiento en Julio de 2008. En sus inicios era una aplicación de escritorio, pero actualmente también se puede utilizar tanto en la web como en dispositivos móviles. Esta aplicación ha sido desarrollada en html5 por Twitter, e interactúa con la API de Twitter para permitir a los usuarios ver y enviar tweets y ver sus perfiles. Además se pueden añadir las siguientes redes: Facebook, LinkedIn, Google Buzz, Foursquare, y MySpace.

Las principales ventajas de esta aplicación son:

- Organización de contenidos en columnas.
- Posibilidad de enviar mensajes de forma simultánea en varias redes sociales.
- Dispone de un acortador de URLs.
- La configuración puede exportarse a otros dispositivos para no tener que volver a configurarla.
- Permite marcar como favoritos los tweets que deseemos, y guardarlos para posterior lectura.
- Función llamada TweetSearch para buscar cualquier tipo de tweet.
- Avisos de actualizaciones en tiempo real.
- Todas las opciones de la aplicación son gratuitas.

La principal desventaja de Tweetdeck reside en que, al igual que en Hootsuite, sólo se pueden añadir redes sociales, limitando así la incorporación de otras aplicaciones de la web.

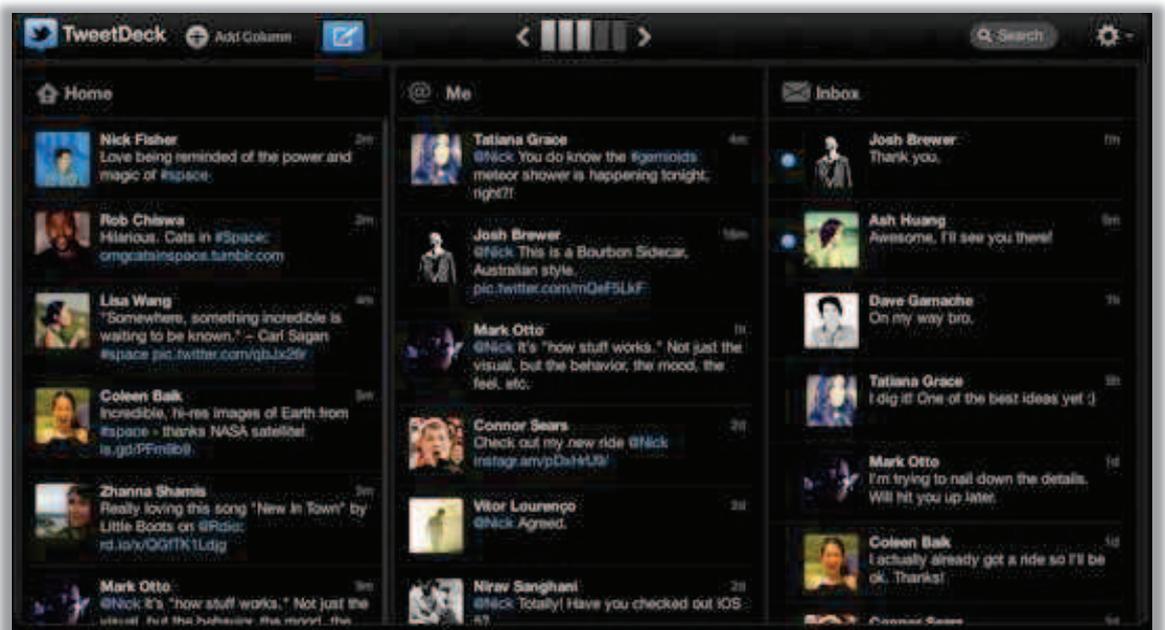


Figura 2.5. Tweetdeck

2.3.1.3 iGoogle

iGoogle [igoogle] surge en el año 2005. Se trata de una página de inicio personalizable basada en AJAX.

Te permite crear una página principal personalizada que incluya un cuadro de búsqueda de Google en la parte superior y todos los gadgets que quieras en la parte inferior. Los gadgets pueden adoptar una gran variedad de formas y permiten acceder a actividades y a información de toda la Web.

Entre las principales características de iGoogle se pueden incluir:

- Ver tus mensajes más recientes de gmail.
- Leer los titulares de Google Noticias y otras fuentes de noticias.
- Consultar las previsiones meteorológicas.
- La información bursátil.
- Cartelera de cine.
- Suscribirte a tus feeds RSS.

Una gran ventaja de esta aplicación es que es muy configurable ya que posee todo tipo de gadgets, tanto de Google como de terceros.

La contra de esta aplicación, reside en que el diseño de los gadgets no son muy atractivos lo que influye con un efecto negativo en el usuario.



Figura 2.6. iGoogle

Capítulo 3. Aspectos Teóricos

3.1 Social media

El “Social Media” [social] o medios de comunicación sociales, son plataformas de comunicación online donde el contenido es creado por los propios usuarios mediante el uso de las tecnologías de la Web 2.0, que facilitan la edición, la publicación y el intercambio de información.

Los tipos de medios sociales más utilizados son las redes sociales, los blogs, los microblogs y los servicios de compartición multimedia.

El social media está en continua evolución. Actualmente Facebook es la red social más popular y probablemente antes de que acabe este año llegará a la cifra redonda de 1.000 millones de usuarios. La segunda red en cuanto a número de usuarios es Twitter, con casi 600 millones, pero seguida muy de cerca por Youtube, que también es considerada una red social por permitir la interacción entre usuarios.

Además de estas existen otras muchas como LinkedIn, red social de profesionales con 150 millones de usuarios; Google+, competencia directa de Facebook o Twitter con 100 millones de usuarios; o Spotify, que cuenta con 15 millones de usuarios de los cuales 3 millones son clientes de pago.



Figura 3.1. Social Media

3.2 OAuth

OAuth [**oauth**] (Open Authorization) es un protocolo abierto, propuesto por Blaine Cook y Chris Messina en noviembre de 2006, que permite autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles, y web. OAuth es utilizado por grandes empresas como Google Apps, Twitter, Facebook, LinkedIn, etc.

Sus principales ventajas son:

- El proceso de integración con las APIs de otras aplicaciones es sencillo.
- Ahorro de tiempo en el desarrollo de la autenticación del usuario.
- No es necesario implementar el apoyo para la renovación de contraseña, contraseña olvidada, etc.
- El usuario introduce sus datos de usuario y contraseña directamente en su proveedor, evitando así que queden guardados en otras aplicaciones.
- Riesgo nulo de robo de identidad, ya que OAuth cuenta con un sistema de tokens encriptados.
- Muchos menos datos para guardar en nuestra aplicación, debido a que se delega parte del trabajo al proveedor.

El funcionamiento es el siguiente:

- En primer lugar el consumidor (aplicación de terceros) ha de haber obtenido una clave privada que identificará a este frente al proveedor del servicio y un “secreto” para establecer la propiedad de la clave y evitar que esta sea explotada por otros.
- Después el usuario accede a su cuenta dentro del proveedor y autorizará al consumidor a acceder a sus recursos.
- A partir de este momento, toda la comunicación se realizará entre el consumidor y el proveedor del servicio.

En el proyecto se utiliza para delegar el sistema de autenticación a la aplicación que lo requiera. Así Flypper no guardará contraseñas de los diferentes servicios de los usuarios.

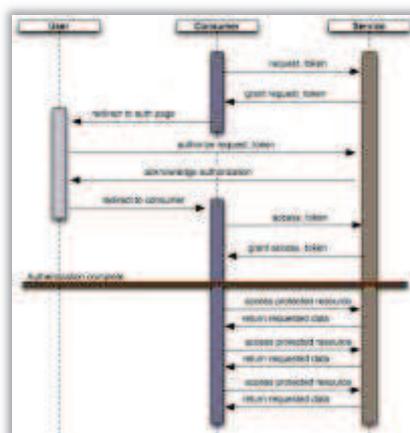


Figura 3.2. OAuth

3.3 HTML5

HTML5 [**html5**] (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un clásico HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).

El desarrollo de este código es regulado por el Consorcio W3C.

Las principales ventajas de html5 son:

- Mejores tiempos de carga ya que separa los elementos del sitio en dos planos, presentando en un primer plano las partes más significativas y que menos tiempo tardan en cargar de manera completa mientras que el resto se va cargando conforme el usuario va leyendo el sitio.
- Permite la renderización de las imágenes lo que incrementa significativamente las posibilidades de crear elementos gráficos de gran interacción para el consumidor.
- Posibilidad de insertar contenido de video y audio sin requerir el apoyo de otras aplicaciones mejorando tiempo y estandarizando calidad de reproducción.
- Mejora la búsqueda semántica, ya que posee mayor claridad de cada sección, lo que facilita el trabajo a los buscadores dentro de los sitios web mejorando el posicionamiento SEO.
- Guardado de la aplicación para poder trabajar offline con ella.

Se utiliza para determinar la estructura y el contenido de la aplicación.

3.4 CSS3

El nombre hojas de estilo en cascada viene del inglés Cascading Style Sheets, del que toma sus siglas. CSS **[css3]** es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML). El W3C (World Wide Web Consortium) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

En la aplicación se utiliza para darle estilo a la estructura y los contenidos creados anteriormente con html.

3.5 PHP

PHP [**php**] es un acrónimo recursivo que significa PHP Hypertext Pre-processor (inicialmente PHP Tools, o, Personal Home Page Tools). Fue creado originalmente por Rasmus Lerdorf en 1994; sin embargo la implementación principal de PHP es producida ahora por The PHP Group y sirve como el estándar de facto para PHP al no haber una especificación formal. Publicado bajo la PHP License, la Free Software Foundation considera esta licencia como software libre.

Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. El lenguaje PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores, el número de sitios en PHP ha compartido algo de su preponderante dominio con otros nuevos lenguajes no tan poderosos desde agosto de 2005. El sitio web de Wikipedia está desarrollado en PHP.

Cuando el cliente hace una petición al servidor para que le envíe una página web, el servidor ejecuta el intérprete de PHP. Éste procesa el script solicitado que generará el contenido de manera dinámica (por ejemplo obteniendo información de una base de datos). El resultado es enviado por el intérprete al servidor, quien a su vez se lo envía al cliente.

Permite la conexión a diferentes tipos de servidores de bases de datos tales como MySQL, PostgreSQL, Oracle, ODBC, DB2, Microsoft SQL Server, Firebird y SQLite.

Ventajas:

- Orientado al desarrollo de aplicaciones web dinámicas con acceso a información almacenada en una base de datos.
- El código fuente escrito en PHP es invisible al navegador web y al cliente ya que es el servidor el que se encarga de ejecutar el código y enviar su resultado HTML al navegador. Esto hace que la programación en PHP sea segura y confiable.
- Capacidad de conexión con la mayoría de los motores de base de datos que se utilizan en la actualidad, destaca su conectividad con MySQL y PostgreSQL.
- Capacidad de expandir su potencial utilizando módulos.
- Posee una amplia documentación en su sitio web oficial, entre la cual se destaca que todas las funciones del sistema están explicadas y ejemplificadas en un único archivo de ayuda.
- Es libre, por lo que se presenta como una alternativa de fácil acceso para todos.
- Permite aplicar técnicas de programación orientada a objetos.
- Biblioteca nativa de funciones sumamente amplia e incluida.

Se utiliza para realizar la lógica de negocio de la aplicación. Se utiliza PHP ya que es el lenguaje con el que está programado Joomla. Joomla se utiliza como base de la aplicación.

3.6 MySQL

MySQL [**mysql**] es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Las principales ventajas de usar MySQL son:

- Amplio subconjunto del lenguaje SQL. Algunas extensiones son incluidas igualmente.
- Disponibilidad en gran cantidad de plataformas y sistemas.
- Posibilidad de selección de mecanismos de almacenamiento que ofrecen diferente velocidad de operación, soporte físico, capacidad, distribución geográfica, transacciones, etc.
- Transacciones y claves foráneas.
- Conectividad segura.
- Replicación.
- Búsqueda e indexación de campos de texto.

Se utiliza para realizar la persistencia de los datos de la aplicación. Se ha elegido este sistema por las ventajas mencionadas anteriormente.

3.7 AJAX

AJAX [**ajax**], acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

AJAX es una tecnología asíncrona, en el sentido de que los datos adicionales se solicitan al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de AJAX mientras que el acceso a los datos se realiza mediante XMLHttpRequest, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

AJAX no constituye una tecnología en sí, sino que es un término que engloba a las siguientes tecnologías que trabajan conjuntamente:

- XHTML (o HTML) y hojas de estilos en cascada (CSS) para el diseño que acompaña a la información.
- Document Object Model (DOM) accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
- El objeto XMLHttpRequest para intercambiar datos de forma asíncrona con el servidor web. En algunos frameworks y en algunas situaciones concretas, se usa un objeto iframe en lugar del XMLHttpRequest para realizar dichos intercambios.
- XML es el formato usado generalmente para la transferencia de datos solicitados al servidor, aunque cualquier formato puede funcionar, incluyendo HTML pre formateado, texto plano, JSON y hasta EBML.

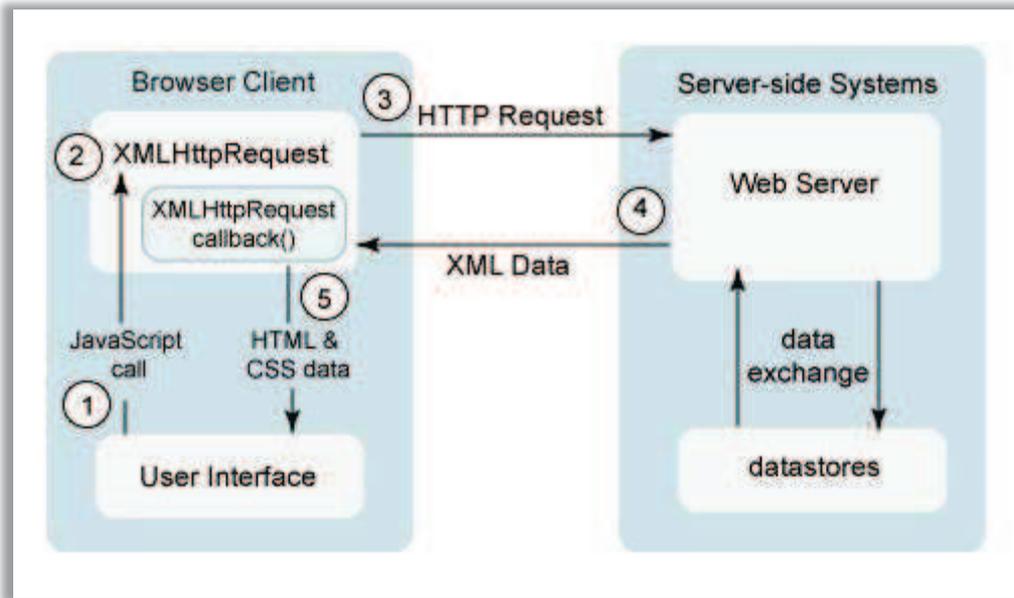


Figura 3.3. AJAX

Se utiliza esta tecnología para realizar las llamadas asíncronas al sistema y realizar cambios en la web sin tener que recargar la página.

3.8 jQuery

jQuery [**jquery**] es una biblioteca de JavaScript, creada inicialmente por John Resig, que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web. Fue presentada el 14 de enero de 2006 en el BarCamp NYC.

jQuery es software libre y de código abierto, posee un doble licenciamiento bajo la Licencia MIT y la Licencia Pública General de GNU v2, permitiendo su uso en proyectos libres y privativos. jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

Se utiliza en la aplicación para mostrar los datos de una forma estéticamente atractiva para los usuarios.

3.9 NetBeans

El IDE NetBeans [**netbeans**] es un entorno de desarrollo integrado - una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Existe además un número importante de módulos para extender el IDE NetBeans. El IDE NetBeans es un producto libre y gratuito sin restricciones de uso.

El NetBeans IDE es un IDE de código abierto escrito completamente en Java usando la plataforma NetBeans. El NetBeans IDE soporta el desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles). Entre sus características se encuentra un sistema de proyectos basado en Ant, control de versiones y refactoring.

NetBeans IDE 6.5, la cual fue publicada el 19 de noviembre de 2008, extiende las características existentes del Java EE (incluyendo Soporte a Persistencia, EJB 3 y JAX-WS). Adicionalmente, el NetBeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales con Java EE 5, incluyendo herramientas de desarrollo visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML. El NetBeans C/C++ Pack soporta proyectos de C/C++, mientras el PHP Pack, soporta PHP 5.

Se utiliza este entorno para llevar a cabo el desarrollo de la aplicación.

3.10 Joomla

Joomla **[joomla]** es un sistema de gestión de contenidos que puede ser utilizado independientemente. Entre sus principales virtudes está la de permitir integrar, añadir o editar el contenido de un sitio web de manera sencilla. Es un código abierto programado mayoritariamente en PHP bajo una licencia GPL. Este administrador de contenidos puede trabajar en redes locales dígase intranet o internet y requiere de una base de datos creada con un gestor MySQL, así como de un servidor HTTP Apache.

En Joomla se incluyen características como: mejoras en el rendimiento web, versiones imprimibles de páginas, flash con noticias, blogs, foros, polls (encuestas), calendarios, búsqueda en el sitio web e internacionalización del lenguaje.

Una de las mayores potencialidades que tiene este CMS es la gran cantidad de extensiones existentes programadas por su comunidad de usuarios que aumentan las posibilidades de Joomla con nuevas características y que se integran fácilmente en él.

Existen cientos de extensiones disponibles y con diversas funcionalidades como por ejemplo:

- Generadores de formularios dinámicos
- Directorios de empresas u organizaciones
- Gestores de documentos
- Galerías de imágenes multimedia
- Motores de comercio y venta electrónica
- Software de foros y chats
- Calendarios
- Software para blogs
- Servicios de directorio
- Boletines de noticias
- Herramientas de registro de datos
- Sistemas de publicación de anuncios

Servicios de suscripción

A su vez estas extensiones se agrupan en:

- Componentes
- Módulos
- Plantillas

Se utiliza en la aplicación como base sobre la que montar nuestros módulos. Se ha elegido Joomla porque simplifica tareas como el login o la gestión del contenido.

Algunos sitios importantes como Linux.com, iTWire.com e incluso eBay.com están basados o usan de algún modo el sistema de gestión de contenidos Joomla.

Capítulo 4. Planificación del Proyecto y Resumen de Presupuestos

4.1 Planificación

A continuación se muestra la planificación del desarrollo del proyecto mediante un diagrama de Gantt. El proyecto tiene como fecha inicial el 28 de Marzo de 2012 y como fecha final el 27 de Julio de 2012, con una duración total de unos 4 meses. Por otro lado, se ha estimado una jornada laboral de 8 horas diarias (40 semanales) trabajando una sola persona.

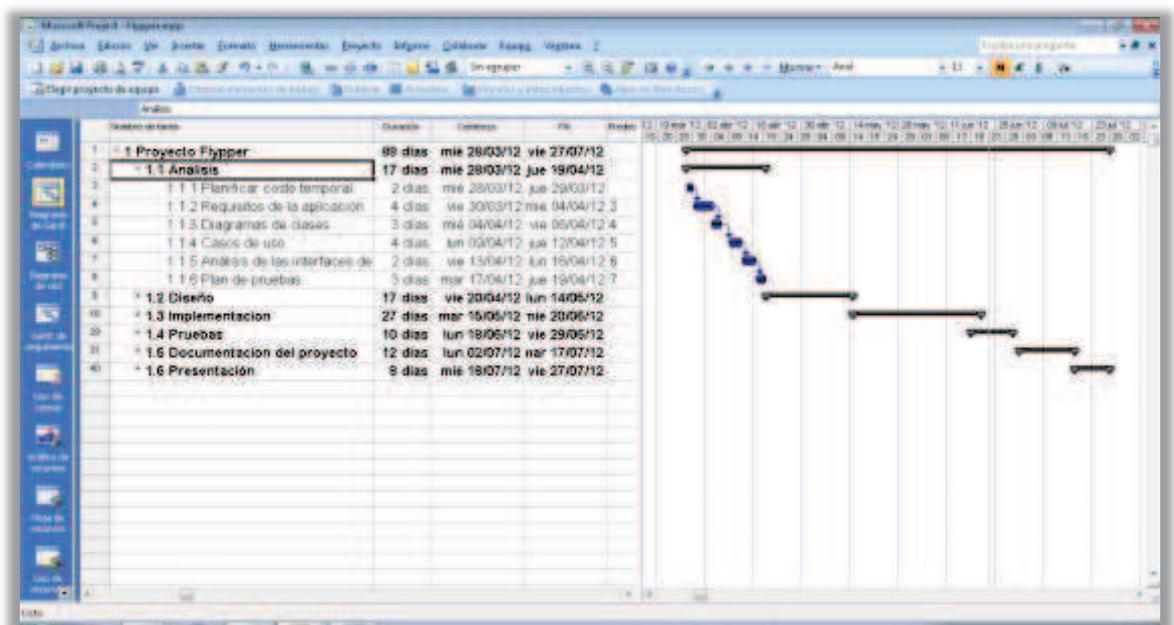


Figura 4.1. Diagrama de Gantt de la fase de análisis

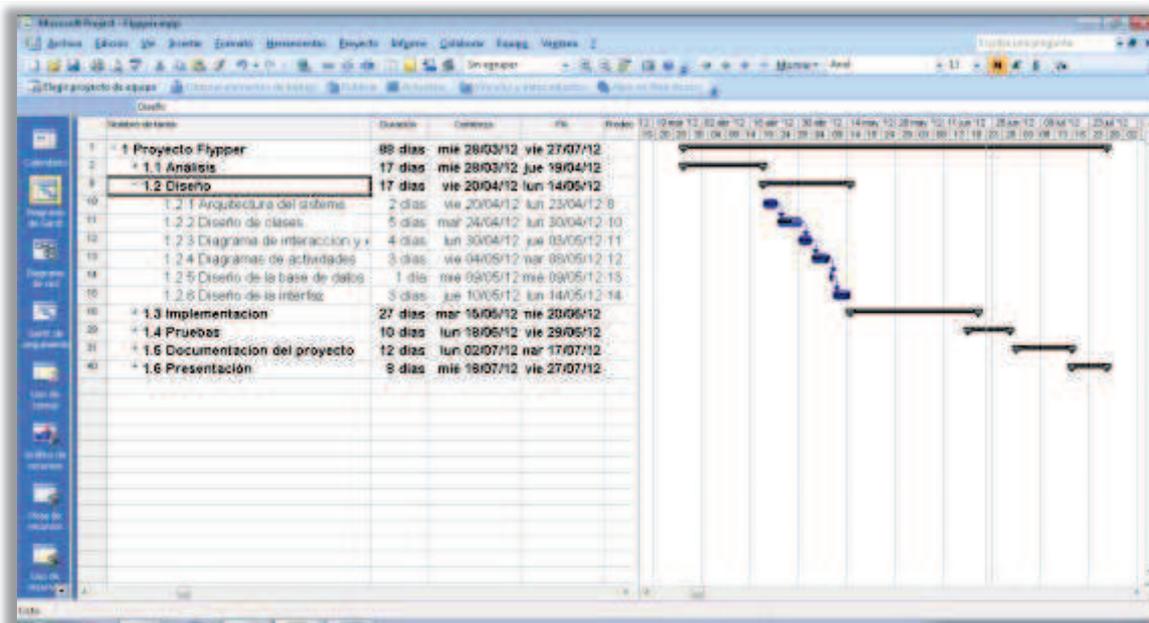


Figura 4.2 Diagrama de Gantt de la fase de diseño

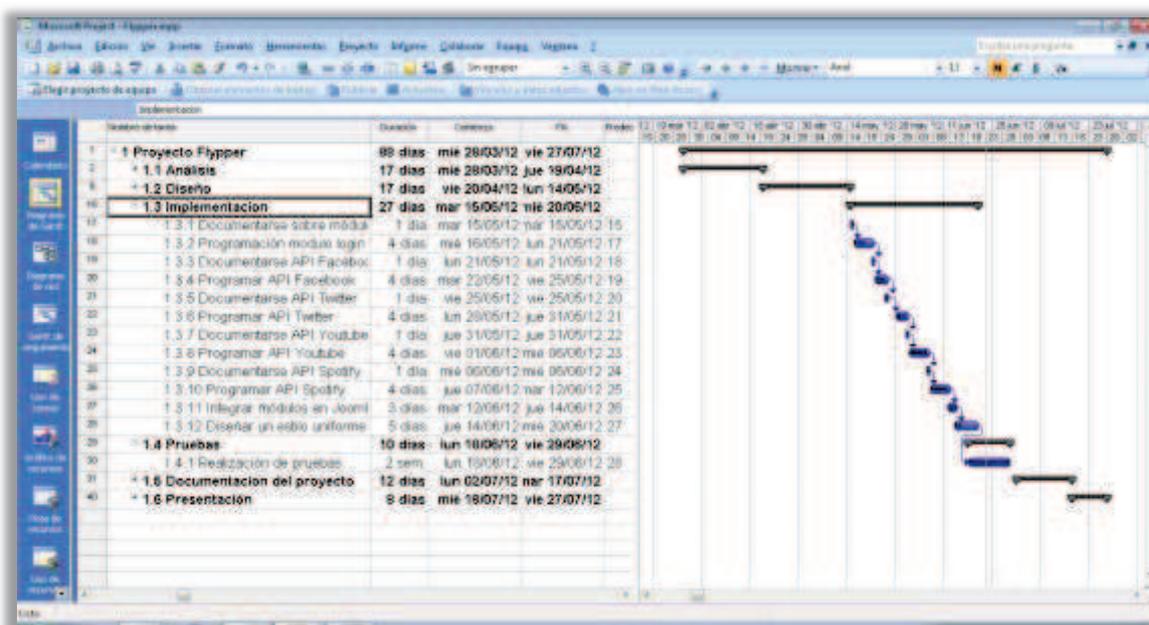


Figura 4.3 Diagrama de Gantt de la fase de implementación y pruebas

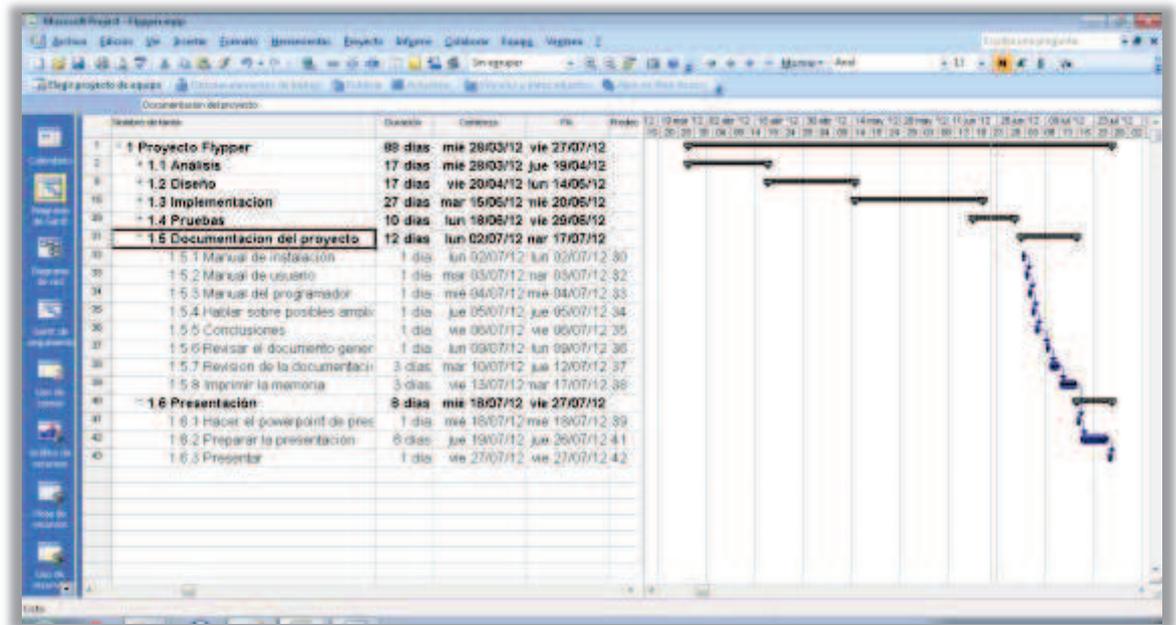


Figura 4.4. Diagrama de Gantt de la fase de documentación y presentación del proyecto

4.2 Resumen del Presupuesto

Item	Concepto	Total
1	Desarrollo de la aplicación	14.880,00 €
2	Software	465,00 €
3	Hardware	600,00 €
4	Servidor	185,00 €
5	Oficina	150,00 €

Subtotal	16.280,00 €
Beneficio (20%)	3.256,00 €
IVA (18%)	2.930,40 €
TOTAL	22.466,40 €

Capítulo 5. Análisis

5.1 Definición del Sistema

5.1.1 Determinación del Alcance del Sistema

Se pretende crear una aplicación web que reúna las aplicaciones de Facebook, Twitter, Youtube y Spotify. Las funcionalidades de cada una son las siguientes:

- Facebook
 - Ver toda la información del usuario en Facebook.
 - Compartir publicaciones.
 - Hacer “me gusta” en otras publicaciones.
 - Comentar las publicaciones de tus amigos en Facebook.
- Twitter
 - Ver toda la información del usuario en Twitter.
 - Compartir tweets.
 - Retwitear otros tweets.
- Youtube
 - Buscar videos en Youtube.
 - Reproducir los videos de Youtube.
- Spotify
 - Buscar canciones en Spotify.
 - Reproducir las canciones de Spotify.

Para acceder a todas estas funcionalidades el usuario deberá estar registrado en nuestra aplicación.

Además, se podrá seleccionar que módulos mostrar y en qué posición, según las preferencias de cada persona.

5.2 Requisitos del Sistema

5.2.1 Obtención de los Requisitos del Sistema

5.2.1.1 Requisitos funcionales

Código	Nombre Requisito	Descripción del Requisito
R1.1	Usuarios	
R1.1.1	Crear usuario	La aplicación deberá tener un sistema de registro de usuarios.
R1.1.2	Hacer login	Los usuarios podrán realizar login en la aplicación.
R1.1.3	Editar usuario	Los usuarios podrán modificar su información en cualquier momento.
R1.1.4	Borrar usuario	El usuario podrá darse de baja de la aplicación.
R1.1.5	Hacer logout	El usuario podrá desconectarse de la aplicación.
R1.2	Twitter	
R1.2.1	Ver inicio Twitter	El usuario podrá ver el inicio de su Twitter.
R1.2.2	Ver perfil Twitter	El usuario podrá acceder a la información de su perfil de Twitter.
R1.2.3	Ver menciones	El usuario podrá ver las menciones de su nombre en Twitter.
R1.2.4	Twittear	El usuario podrá Twittear con una restricción de 140 caracteres.
R1.2.5	Borrar tweet	El usuario podrá eliminar sus tweets.
R1.2.6	Retweet	El usuario podrá retweetear un tweet de otro usuario.
R1.2.7	Cerrar sesión Twitter	El usuario podrá cerrar sesión en Twitter y borrar sus permisos de acceso a la aplicación.
R1.3	Facebook	
R1.3.1	Ver inicio Facebook	El usuario podrá ver el inicio de su Facebook.
R1.3.2	Ver perfil Facebook	El usuario podrá acceder a la información de su perfil de Facebook.
R1.3.3	Publicar en Facebook	El usuario podrá realizar publicaciones en Facebook.
R1.3.4	Ver publicación	El usuario podrá ver la publicación con sus comentarios y "me gusta".
R1.3.5	Borrar publicación	El usuario podrá borrar cualquier publicación que haya realizado.
R1.3.6	Comentar publicación	El usuario podrá comentar tanto sus publicaciones como las de los demás usuarios.
R1.3.7	Hacer "me gusta"	El usuario podrá hacer "me gusta" tanto en sus publicaciones como en las de los demás usuarios.
R1.3.8	Cerrar sesión Facebook	El usuario podrá cerrar sesión en Facebook y borrar sus permisos de acceso a la aplicación.
R1.4	Youtube	
R1.4.1	Ver videos recomendados	El usuario podrá acceder a una lista de videos recomendados.
R1.4.2	Ver videos más vistos	El usuario podrá acceder a una lista de los videos

		más vistos de España.
R1.4.3	Ver videos trending	El usuario podrá ver una lista de los videos más comentados del momento.
R1.4.4	Buscar video	El usuario podrá realizar búsquedas de videos en Youtube.
R1.4.5	Reproducir video	El usuario podrá ver cualquier video de Youtube.
R1.5	Spotify	
R1.5.1	Buscar canción	El usuario podrá buscar una canción en Spotify.
R1.5.2	Ver portada	El usuario podrá ver la portada del álbum de la canción que desee.
R1.5.3	Reproducir canción	El usuario podrá reproducir cualquier canción que desee.
R1.5.4	Compartir en Facebook	El usuario podrá compartir cualquier canción en su Facebook.
R1.5.5	Compartir en Twitter	El usuario podrá compartir cualquier canción en su Twitter.

5.2.1.2 Requisitos no funcionales

Código	Nombre Requisito	Descripción del Requisito
R2.1	De usuario	
R2.1.2	Conocimientos básicos de internet.	El usuario deberá tener unos conocimientos básicos sobre el uso de internet y las redes sociales.
R2.2	Tecnológicos	
R2.2.1	Navegador	El usuario deberá tener un navegador que soporte Javascript, debido al uso de AJAX, y html5.
R2.2.2	Spotify	Para poder reproducir las canciones de Spotify, el usuario deberá tener instalada la aplicación de escritorio de Spotify.
R2.2.3	Joomla	Se implantará el sistema sobre el gestor de contenidos Joomla
R2.2.4	Template para Joomla	El template para Joomla será impuesto por el cliente.
R2.3	Usabilidad	
R2.3.1	Interfaz amigable	La aplicación debe tener una interfaz amigable para que el usuario no tenga dificultades en su uso.
R2.3.2	Presentación sencilla de la información	La aplicación debe presentar la información de forma que el usuario la asimile fácilmente.
R2.4	Seguridad	
R2.4.1	Registro de usuarios	Para acceder a la aplicación y poder usarla, el usuario deberá registrarse, y tener permisos de usuario registrado.
R2.5	Tiempo de respuesta	
R2.5.1	Rapidez en la presentación de la información	El tiempo de respuesta de la aplicación deberá ser el menor posible para evitar que el usuario tenga que esperar.

5.2.2 Identificación de Actores del Sistema

En la aplicación existen los siguientes tipos de actores:

- Usuario anónimo: este usuario solamente podrá acceder a la página principal de la aplicación donde se pueden ver sus características y acceder a la zona de registro.
- Usuario registrado: este usuario podrá acceder a toda la aplicación. Podrá utilizar todos los servicios como son Facebook, Twitter, Youtube y Spotify.
- API de Facebook: sistema con el que interactúa la aplicación para realizar consultas de información sobre un usuario, compartir o borrar una publicación en Facebook.
- API de Twitter: sistema con el que interactúa la aplicación para realizar consultas de información sobre un usuario, compartir o borrar una publicación en Twitter.
- API de Youtube: sistema con el que interactúa la aplicación para realizar búsquedas de vídeos.
- API de Spotify: sistema con el que interactúa la aplicación para realizar búsquedas de canciones.

5.2.3 Especificación de Casos de Uso

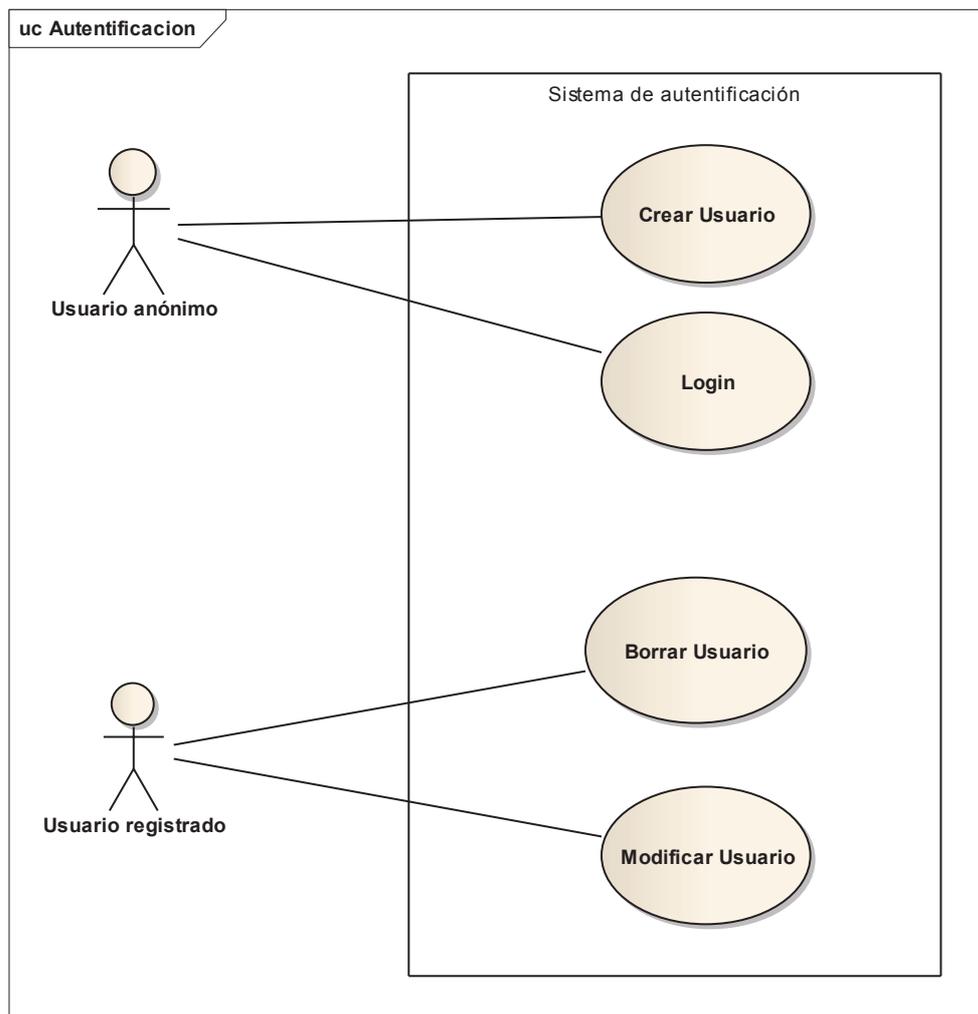


Figura 5.1. Casos de uso relacionados por la autenticación llevada a cabo por Joomla.

Nombre del Caso de Uso
Crear usuario
Descripción
Un usuario podrá entrar en la web y registrarse en la aplicación para empezar a utilizarla. Si no se ha registrado sólo podrá acceder a la página principal, donde se encuentra información sobre la aplicación. De este caso de uso se encargará Joomla.

Nombre del Caso de Uso
Login
Descripción
Una vez que el usuario se ha registrado en la aplicación, podrá realizar login para entrar en la aplicación y acceder a su información. De este caso de uso se encargará Joomla.

Nombre del Caso de Uso
Borrar usuario

Descripción
Previo registro del usuario, éste podrá eliminar su perfil de usuario en la aplicación. De este caso de uso se encargará Joomla.

Nombre del Caso de Uso
Modificar usuario
Descripción
Previo registro del usuario, se podrá modificar los datos del mismo mediante un formulario. De este caso de uso se encargará Joomla.

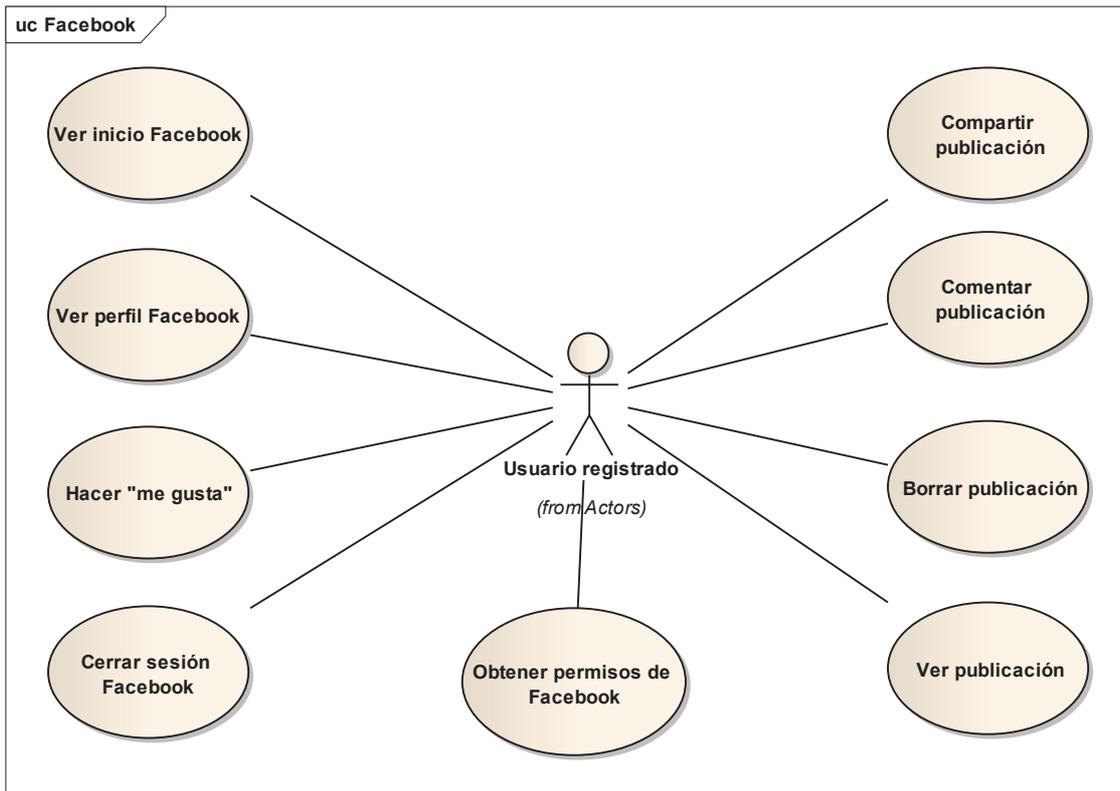


Figura 5.2. Casos de uso sobre el módulo de Facebook

Nombre del Caso de Uso
Obtener permisos de Facebook
Descripción
Una vez logueado, el usuario deberá autorizar a Facebook para que la aplicación obtenga los permisos necesarios para acceder a la información y realizar modificaciones en la cuenta del usuario.

Nombre del Caso de Uso
Ver inicio Facebook
Descripción
Una vez logueado y obtenida la autorización de Facebook, el usuario podrá acceder a ver su

inicio en Facebook.

Nombre del Caso de Uso

Ver perfil Facebook

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá acceder a ver su perfil en Facebook.

Nombre del Caso de Uso

Hacer me gusta

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá hacer “me gusta” en las publicaciones suyas o de otros usuarios.

Nombre del Caso de Uso

Compartir publicación

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá compartir cualquier publicación en Facebook y que los demás usuarios la vean.

Nombre del Caso de Uso

Comentar publicación

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá comentar una publicación suya o de otro usuario.

Nombre del Caso de Uso

Ver publicación

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá ver los comentarios de cada publicación además de cuantos “me gusta” tiene.

Nombre del Caso de Uso

Borrar publicación

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá borrar cualquier publicación que haya compartido él mismo.

Nombre del Caso de Uso

Cerrar sesión Facebook

Descripción

Una vez logueado y obtenida la autorización de Facebook, el usuario podrá borrar la autorización de Facebook cerrando así la sesión.

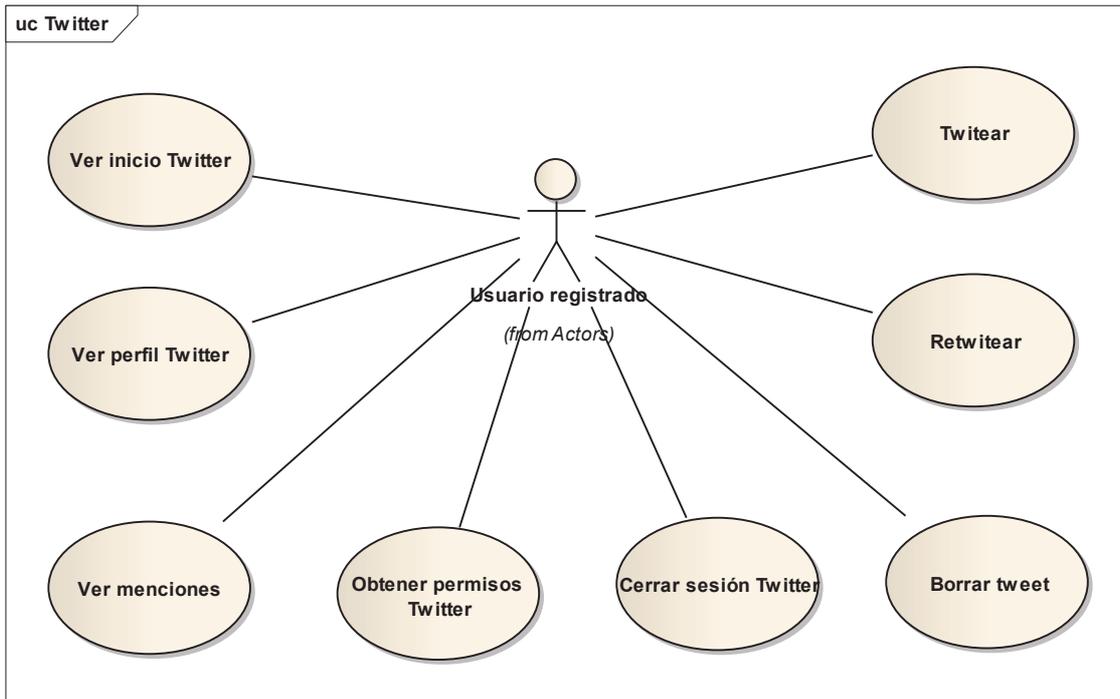


Figura 5.3. Casos de uso sobre el módulo de Twitter

Nombre del Caso de Uso
Obtener permisos Twitter
Descripción
Una vez logueado, el usuario deberá autorizar a Twitter para que la aplicación obtenga los permisos necesarios para acceder a la información y realizar modificaciones en la cuenta del usuario.

Nombre del Caso de Uso
Ver inicio Twitter
Descripción
Una vez logueado y obtenida la autorización de Twitter, el usuario podrá acceder a ver su inicio en Twitter.

Nombre del Caso de Uso
Ver perfil Twitter
Descripción
Una vez logueado y obtenida la autorización de Twitter, el usuario podrá acceder a ver su perfil en Twitter.

Nombre del Caso de Uso
Ver menciones
Descripción

Una vez logueado y obtenida la autorización de Twitter, el usuario podrá acceder a ver sus menciones en Twitter.

Nombre del Caso de Uso

Twitear

Descripción

Una vez logueado y obtenida la autorización de Twitter, el usuario podrá twitear.

Nombre del Caso de Uso

Retwitear

Descripción

Una vez logueado y obtenida la autorización de Twitter, el usuario podrá retwitear los tweets de otros usuarios en su perfil.

Nombre del Caso de Uso

Borrar tweet

Descripción

Una vez logueado y obtenida la autorización de Twitter, el usuario podrá borrar cualquiera de sus tweets.

Nombre del Caso de Uso

Cerrar sesión Twitter

Descripción

Una vez logueado y obtenida la autorización de Twitter, el usuario podrá borrar la autorización de Twitter cerrando así la sesión.

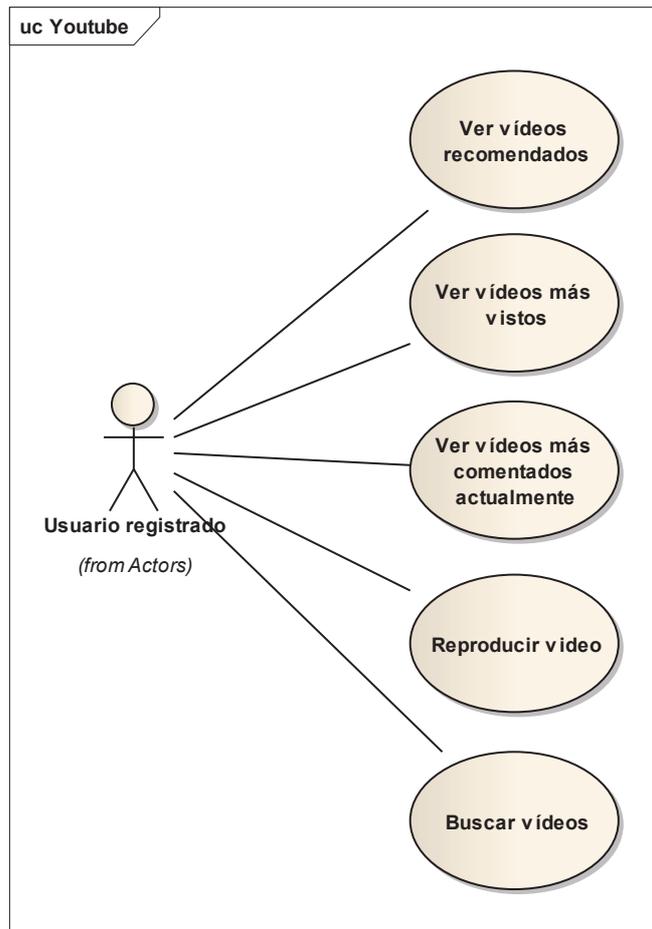


Figura 5.4. Casos de uso sobre el módulo de Youtube

Nombre del Caso de Uso	
Ver vídeos recomendados	
Descripción	
La aplicación mostrará al usuario los videos recomendados.	

Nombre del Caso de Uso	
Ver vídeos más vistos	
Descripción	
La aplicación mostrará al usuario los videos más vistos en España.	

Nombre del Caso de Uso	
Ver vídeos más comentados actualmente	
Descripción	
La aplicación mostrará al usuario los videos más comentados actualmente.	

Nombre del Caso de Uso	
Buscar vídeos	
Descripción	
La aplicación permitirá al usuario buscar vídeos en Youtube.	

Nombre del Caso de Uso
Reproducir vídeos
Descripción
La aplicación permitirá al usuario reproducir vídeos en Youtube que previamente ha buscado.

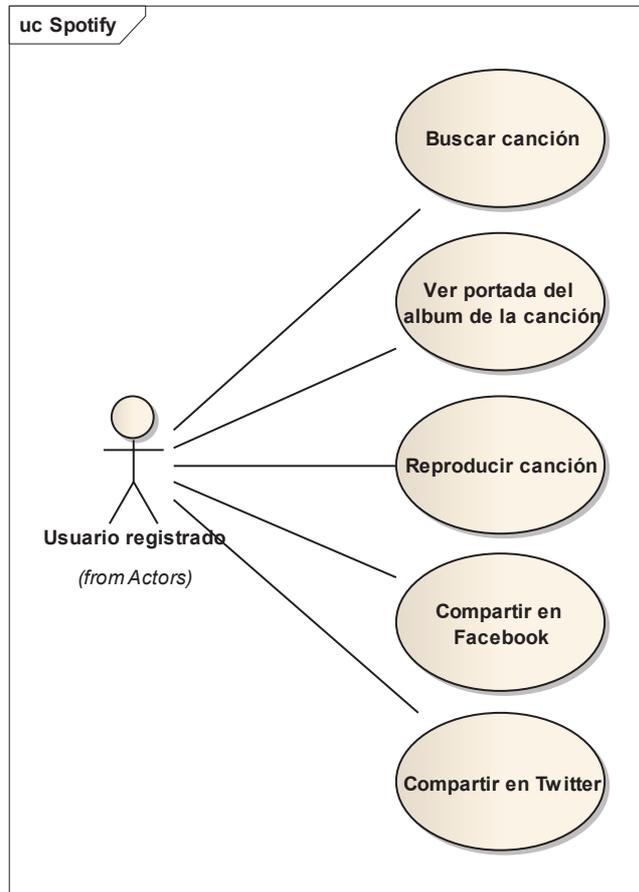


Figura 5.5. Casos de uso sobre el módulo de Spotify

Nombre del Caso de Uso
Buscar canción
Descripción
El usuario podrá buscar canciones en Spotify.

Nombre del Caso de Uso
Ver portada del álbum de la canción
Descripción
El usuario podrá visualizar la portada del álbum de la canción que haya buscado.

Nombre del Caso de Uso
Reproducir canción
Descripción

El usuario podrá reproducir la canción que haya buscado. Para poderse realizar la reproducción de la canción el usuario deberá tener instalado previamente la aplicación de escritorio de Spotify.

Nombre del Caso de Uso

Compartir en Facebook

Descripción

El usuario podrá compartir en Facebook la canción que haya buscado previamente.

Nombre del Caso de Uso

Compartir en Twitter

Descripción

El usuario podrá compartir en Twitter la canción que haya buscado previamente.
--

5.3 Identificación de los Subsistemas en la Fase de Análisis

5.3.1 Descripción de los Subsistemas

En esta sección debemos enumerar todos los subsistemas que identifiquemos inicialmente en la aplicación. Los subsistemas son agrupaciones de paquetes y clases que tienen un objetivo propósito común. Ejemplos de subsistemas pueden ser todas las clases que manejen la base de datos (subsistema “base de datos”), clases que agrupen un conjunto de servicios relacionados, clases del cliente de esos servicios, etc.

Autenticación

Este subsistema será el encargado de realizar todas las gestiones relacionadas con el usuario. Estas son:

- Crear usuario
- Modificar usuario
- Eliminar usuario
- Hacer login

De este subsistema se encargará el propio Joomla.

Facebook

Este subsistema es el encargado de ejecutar todas las funciones relacionadas con el módulo de Facebook, tales como

Twitter

Este subsistema es el encargado de ejecutar todas las funciones relacionadas con el módulo de Twitter, tales como:

- Twitear.
- Borrar tweet.
- Retwitear.
- Etc.

Youtube

Este subsistema es el encargado de ejecutar todas las funciones relacionadas con el módulo de Youtube, tales como:

- Buscar videos.
- Reproducir videos.
- Etc.

Spotify

Este subsistema es el encargado de ejecutar todas las funciones relacionadas con el módulo de Spotify, tales como:

- Buscar canción.
- Reproducir canción.
- Compartir canción.
- Etc.

5.3.2 Descripción de los Interfaces entre Subsistemas

Los subsistemas de Facebook, Twitter, Youtube y Spotify no se comunican entre ellos ya que cada uno es independiente del otro. Cada uno de ellos se comunica con su correspondiente API para conseguir los datos necesarios en cada momento.

5.4 Diagrama de Clases Preliminar del Análisis

5.4.1 Diagrama de Clases

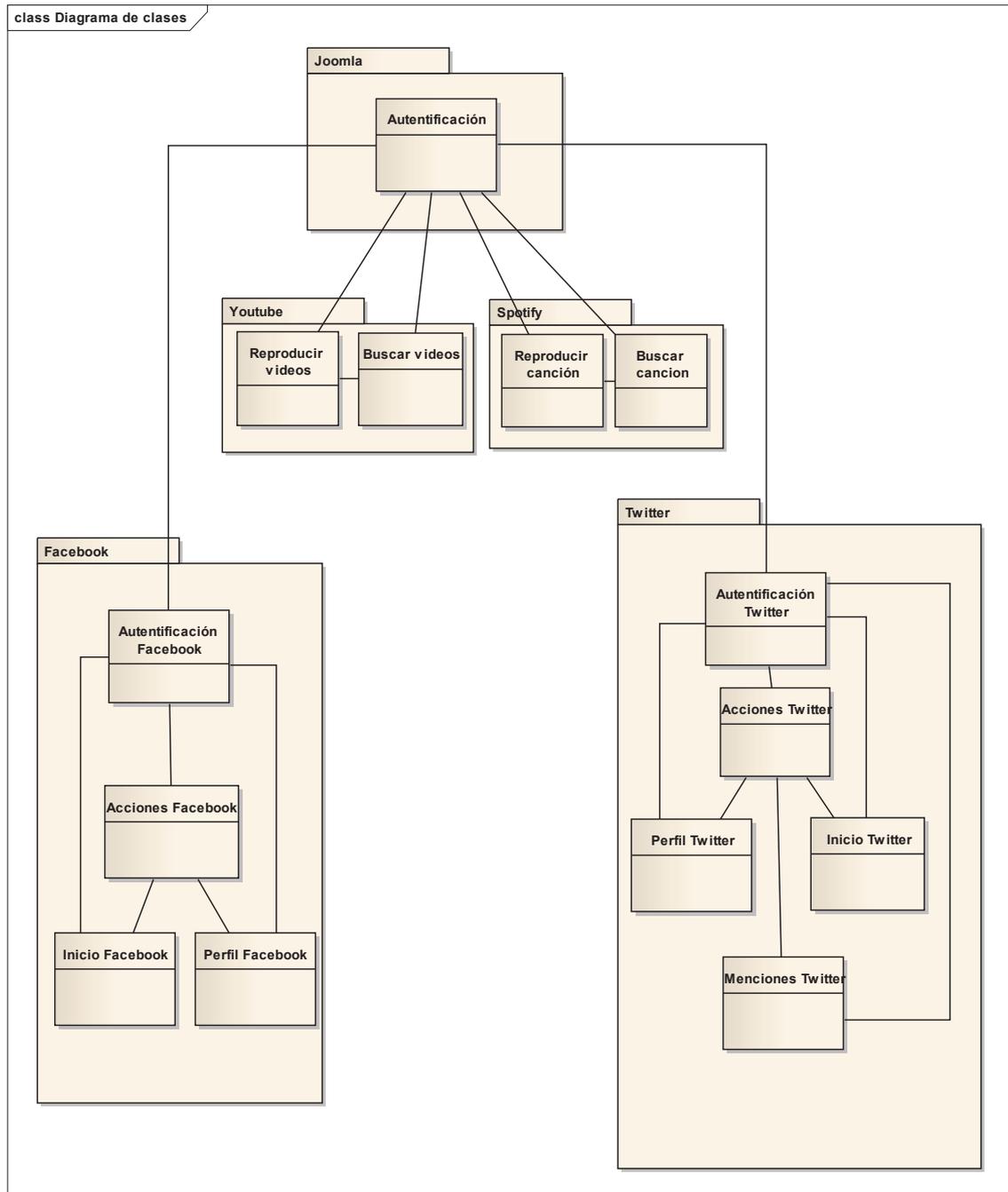


Figura 5.6. Diagrama de clases

5.4.2 Descripción de las Clases

5.4.2.1 Facebook

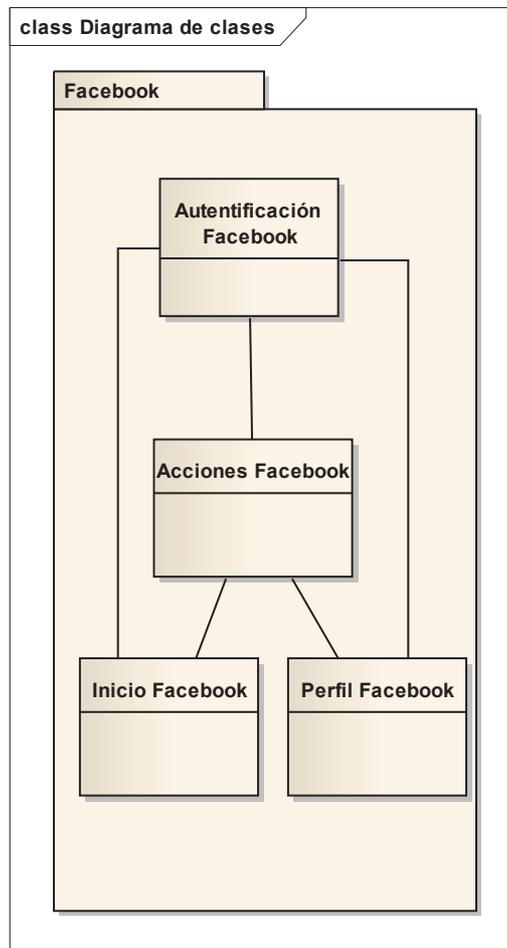


Figura 5.7. Módulo Facebook

Nombre de la Clase
Autenticación Facebook
Descripción
Será la encargada de comprobar si el usuario está registrado en Flypper y autenticado en Facebook. Si no lo está conseguirá el permiso para poder trabajar con el API de Facebook.
Responsabilidades
Atributos Propuestos
id_usuario : id del usuario registrado en Flypper
oauth_token : token de autorización que permite a Flypper utilizar el API de Facebook
Métodos Propuestos
obtener_autorización : Obtiene la autorización para poder utilizar el API de Facebook

Nombre de la Clase
Inicio Facebook
Descripción
Será la encargada de consultar el API de Facebook y mostrar los datos de inicio del usuario.
Responsabilidades
Atributos Propuestos
id_usuario : id del usuario registrado en Flypper oauth_token : token de autorización que permite a Flypper utilizar el API de Facebook
Métodos Propuestos
ver_usuario : Muestra los datos de usuario. ver_timeline : Muestra el time line del usuario.

Nombre de la Clase
Perfil Facebook
Descripción
Será la encargada de consultar el API de Facebook y mostrar los datos de perfil del usuario.
Responsabilidades
Atributos Propuestos
id_usuario : id del usuario registrado en Flypper oauth_token : token de autorización que permite a Flypper utilizar el API de Facebook
Métodos Propuestos
ver_usuario : Muestra los datos de usuario. ver_perfil : Muestra la información del perfil.

Nombre de la Clase
Acciones Facebook
Descripción
Será la encargada de realizar las acciones de compartir publicaciones, hacer me gusta, comentar publicaciones y borrar publicaciones en Facebook.
Responsabilidades
Atributos Propuestos
id_usuario : id del usuario registrado en Flypper oauth_token : token de autorización que permite a Flypper utilizar el API de Facebook
Métodos Propuestos
compartir_publicacion : esta función permite compartir una publicación. me_gusta : esta función permite hacer me gusta en las publicaciones. comentar_publicacion : permite comentar una publicación. borrar_publicacion : permite borrar una publicación.

5.4.2.2 Twitter

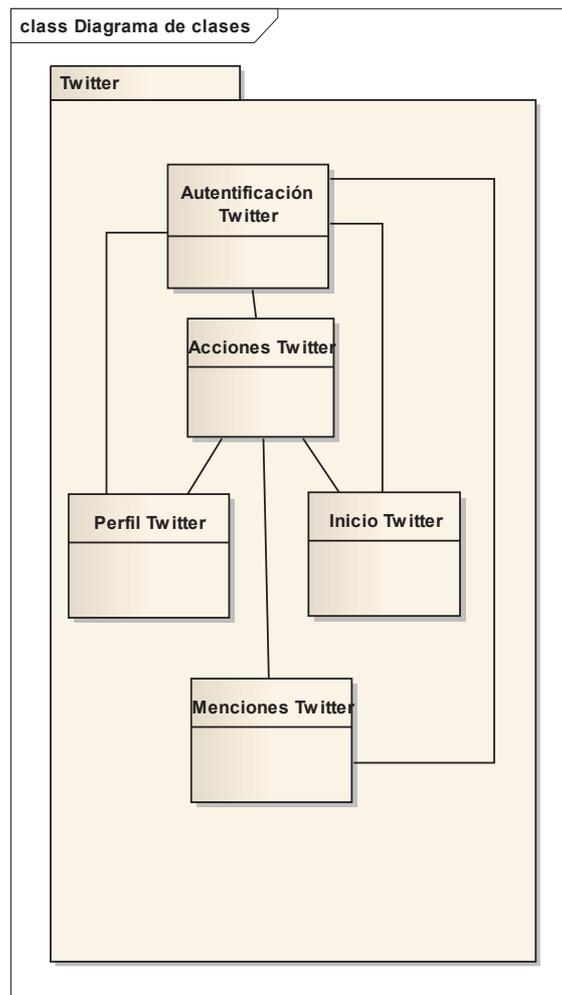


Figura 5.8. Módulo Twitter

Nombre de la Clase
Autenticación Twitter
Descripción
Será la encargada de comprobar si el usuario está registrado en Flypper y autenticado en Twitter. Si no lo está conseguirá el permiso para poder trabajar con el API de Twitter.
Responsabilidades
Atributos Propuestos
id_usuario: id del usuario registrado en Flypper oauth_token: token de autorización que permite a Flypper utilizar el API de Twitter
Métodos Propuestos
obtener_autorización: Obtiene la autorización para poder utilizar el API de Twitter

Nombre de la Clase
Inicio Twitter
Descripción
Será la encargada de consultar el API de Twitter y mostrar el inicio del usuario de Twitter.

Responsabilidades
Atributos Propuestos
id_usuario: id del usuario registrado en Flypper oauth_token: token de autorización que permite a Flypper utilizar el API de Twitter
Métodos Propuestos
ver_inicio: Muestra la información del inicio.

Nombre de la Clase
Perfil Twitter
Descripción
Será la encargada de consultar el API de Twitter y mostrar el perfil del usuario de Twitter.
Responsabilidades
Atributos Propuestos
id_usuario: id del usuario registrado en Flypper oauth_token: token de autorización que permite a Flypper utilizar el API de Twitter
Métodos Propuestos
ver_perfil: Muestra la información del perfil.

Nombre de la Clase
Menciones Twitter
Descripción
Será la encargada de consultar el API de Twitter y mostrar las menciones del usuario de Twitter.
Responsabilidades
Atributos Propuestos
id_usuario: id del usuario registrado en Flypper oauth_token: token de autorización que permite a Flypper utilizar el API de Twitter
Métodos Propuestos
ver_menciones: Muestra las menciones del usuario.

Nombre de la Clase
Acciones Twitter
Descripción
Será la encargada de realizar las acciones de twitear, hacer retweets y eliminar tweets en Twitter.
Responsabilidades
Atributos Propuestos
id_usuario: id del usuario registrado en Flypper oauth_token: token de autorización que permite a Flypper utilizar el API de Twitter
Métodos Propuestos
twitear: permite twitear un tweet retwitear: permite retwitear un tweet borrar_tweet: permite eliminar un tweet

5.4.2.3 Youtube

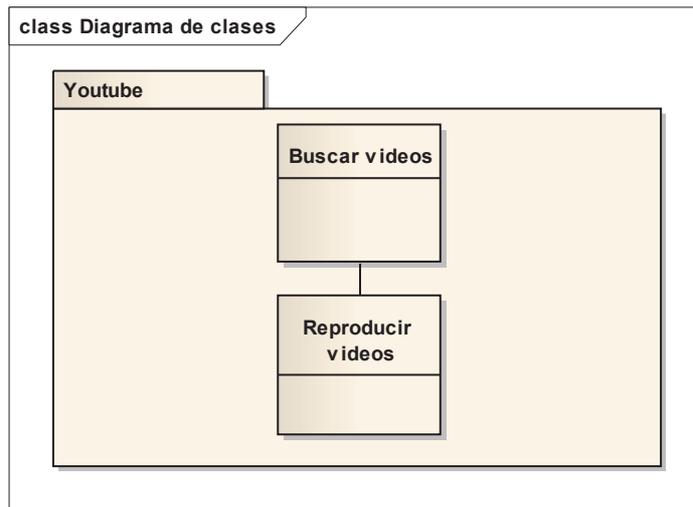


Figura 5.9. Módulo Youtube

Nombre de la Clase
Reproducir videos
Descripción
Será la encargada de reproducir un video en Youtube.
Responsabilidades
Atributos Propuestos
Métodos Propuestos
reproducir_video: permite reproducir un video

Nombre de la Clase
Buscar videos
Descripción
Será la encargada de buscar un video en Youtube.
Responsabilidades
Atributos Propuestos
Métodos Propuestos
buscar_video: permite buscar un video

5.4.2.4 Spotify

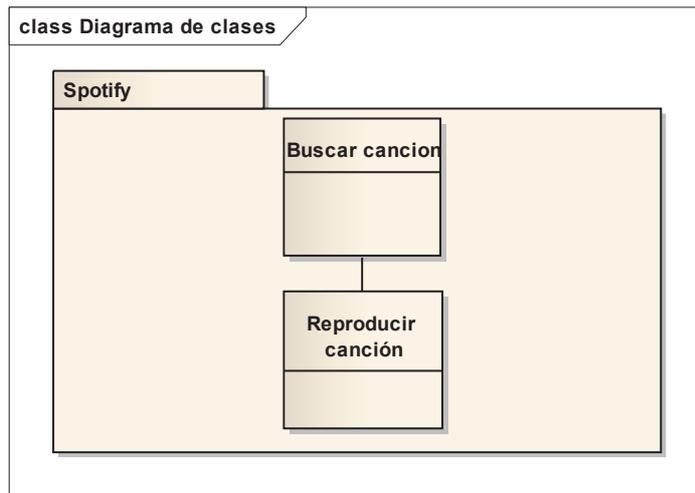


Figura 5.10. Módulo Spotify

Nombre de la Clase	
Reproducir canción	
Descripción	Será la encargada de reproducir una canción en Spotify.
Responsabilidades	
Atributos Propuestos	
Métodos Propuestos	
	buscar_cancion: permite buscar una canción

Nombre de la Clase	
Buscar canción	
Descripción	Será la encargada de buscar una canción en Spotify.
Responsabilidades	
Atributos Propuestos	
Métodos Propuestos	
	buscar_cancion: permite buscar una canción

5.5 Análisis de Casos de Uso y Escenarios

El análisis de los casos de uso se va a realizar desde un punto de vista funcional. Ya que así se permite enumerar más directamente la funcionalidad del sistema. A continuación se muestran los diagramas.

5.5.1 Obtener permisos Facebook

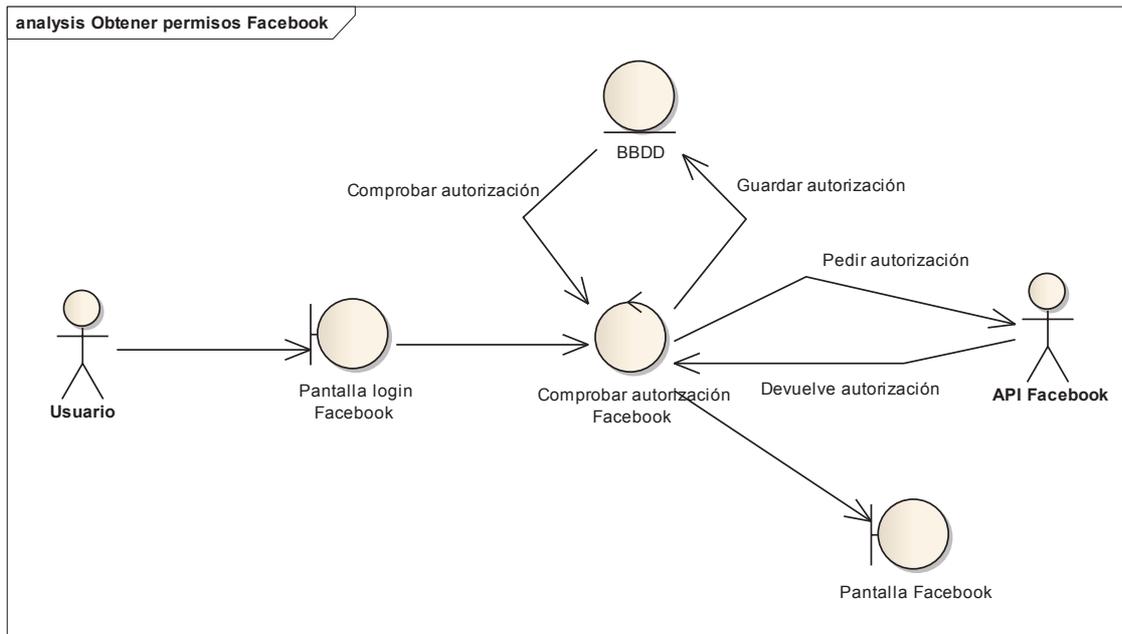


Figura 5.11. Diagrama de robustez: Obtener permisos Facebook

Obtener permisos Facebook	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La aplicación ha obtenido una autorización de Facebook para poder utilizar el API.
Actores	Usuario registrado y API de Facebook
Descripción	<p>El usuario registrado:</p> <ol style="list-style-type: none"> 1. Accederá a la pantalla de login de Facebook. 2. La aplicación comprobará si ya tiene la autorización de Facebook. 3. Si no la tiene, la aplicación pedirá al API de Facebook la autorización. 4. Guardará el examen. 5. La aplicación guardará en la base de datos la autorización. 6. Se le devolverá al usuario la pantalla del inicio de Facebook.
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La aplicación ya tiene la autorización de Facebook. <ul style="list-style-type: none"> ○ Ir al paso 6. • Escenario Alternativo 2: La aplicación no puede obtener la

	<p>autorización del API de Facebook.</p> <ul style="list-style-type: none"> ○ Ir al paso 1 ● Escenario Alternativo 3: La aplicación no puede obtener la autorización de la base de datos. <ul style="list-style-type: none"> ○ Ir al paso 3
Excepciones	<ul style="list-style-type: none"> ● No se puede obtener la autorización por ninguno de los medios <ul style="list-style-type: none"> ○ Mostrar pantalla login de Facebook.
Notas	

5.5.2 Ver inicio de Facebook

Ver inicio de Facebook	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza su información de inicio de Facebook.
Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña "Inicio". 2. La aplicación muestra el inicio de Facebook.
Variaciones (escenarios secundarios)	
Excepciones	<ul style="list-style-type: none"> ● No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.3 Ver perfil de Facebook

Ver perfil de Facebook	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza su información de perfil de Facebook.
Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña "Perfil". 2. La aplicación muestra el perfil de Facebook.
Variaciones (escenarios secundarios)	
Excepciones	<ul style="list-style-type: none"> ● No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.4 Hacer “me gusta”

Hacer “me gusta”	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario hace me gusta en una publicación de Facebook.
Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en el botón “Me gusta”. 2. La aplicación hace un “Me gusta” en la publicación de Facebook. 3. El botón cambia a “Ya no me gusta”.
Variaciones (escenarios secundarios)	<p>Escenario alternativo 1: La publicación ya había sido marcada previamente como “Me gusta”.</p> <ol style="list-style-type: none"> 1. El usuario hace click en el botón “Ya no me gusta”. 2. La aplicación hace un “Ya no me gusta” en la publicación de Facebook. 3. El botón cambia a “Me gusta”.
Excepciones	
Notas	

5.5.5 Ver publicación

Ver publicación	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza los comentarios de la publicación.
Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la publicación. 2. La aplicación muestra los comentarios de la publicación y un espacio para comentar la publicación.
Variaciones (escenarios secundarios)	<p>Escenario alternativo 1: La publicación no tiene ningún comentario.</p> <ol style="list-style-type: none"> 1. La aplicación muestra un espacio para comentar la publicación.
Excepciones	
Notas	

5.5.6 Publicar

Publicar	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La publicación queda compartida en Facebook

Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario escribe algo para publicar. 2. La aplicación lo comparte en Facebook.
Variaciones (escenarios secundarios)	
Excepciones	Si hay algún error la aplicación no comparte la publicación y muestra la pantalla de inicio.
Notas	

5.5.7 Comentar publicación

Comentar publicación	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El comentario queda compartido en la publicación de Facebook
Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario escribe algo para publicar. 2. La aplicación lo comparte en Facebook.
Variaciones (escenarios secundarios)	
Excepciones	Si hay algún error la aplicación no comparte la publicación y muestra la pantalla de inicio.
Notas	

5.5.8 Borrar publicación

Borrar publicación	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente y ha debido compartir alguna publicación.
Poscondiciones	El usuario borra una publicación que ha compartido anteriormente.
Actores	Usuario registrado y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en el botón para borrar la publicación. 2. La aplicación elimina la publicación.
Variaciones (escenarios secundarios)	
Excepciones	
Notas	

5.5.9 Cerrar sesión Facebook

Cerrar sesión Facebook	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La autorización de Facebook queda borrada de la base de datos
Actores	Usuario registrado
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click “cerrar sesión”. 2. La aplicación elimina la autorización de Facebook de la base de datos. 3. La aplicación muestra la pantalla de inicio de Facebook.
Notas	

5.5.10 Obtener permisos Twitter

El diagrama de robustez es el mismo que el de la Figura 5.11.

Obtener permisos Twitter	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La aplicación ha obtenido una autorización de Twitter para poder utilizar el API.
Actores	Usuario registrado y API de Twitter
Descripción	<p>El usuario registrado:</p> <ol style="list-style-type: none"> 1. Accederá a la pantalla de login de Twitter. 2. La aplicación comprobará si ya tiene la autorización de Twitter. 3. Si no la tiene, la aplicación pedirá al API de Twitter la autorización. 4. Guardará el examen. 5. La aplicación guardará en la base de datos la autorización. 6. Se le devolverá al usuario la pantalla del inicio de Twitter.
Variaciones (escenarios secundarios)	<ul style="list-style-type: none"> • Escenario Alternativo 1: La aplicación ya tiene la autorización de Twitter. <ul style="list-style-type: none"> ○ Ir al paso 6. • Escenario Alternativo 2: La aplicación no puede obtener la autorización del API de Twitter. <ul style="list-style-type: none"> ○ Ir al paso 1 • Escenario Alternativo 3: La aplicación no puede obtener la autorización de la base de datos. <ul style="list-style-type: none"> ○ Ir al paso 3
Excepciones	<ul style="list-style-type: none"> • No se puede obtener la autorización por ninguno de los medios <ul style="list-style-type: none"> ○ Mostrar pantalla login de Twitter.
Notas	-

5.5.11 Ver inicio de Twitter

Ver inicio de Twitter	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza su información de inicio de Twitter.
Actores	Usuario registrado y API de Twitter.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña "Inicio". 2. La aplicación muestra el inicio de Twitter.
Variaciones (escenarios secundarios)	
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.12 Ver perfil de Twitter

Ver perfil de Twitter	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza su información de perfil de Twitter.
Actores	Usuario registrado y API de Twitter.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña "Perfil". 2. La aplicación muestra el perfil de Twitter.
Variaciones (escenarios secundarios)	
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.13 Ver menciones

Ver menciones	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza sus menciones en Twitter.
Actores	Usuario registrado y API de Twitter.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña "Menciones". 2. La aplicación muestra sus menciones en Twitter.
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.14 Twitear

Twitear	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El tweet queda twiteado en Twitter
Actores	Usuario registrado y API de Twitter.
Descripción	<ol style="list-style-type: none"> 1. El usuario escribe algo para retwitear. 2. La aplicación comparte el tweet en Twitter.
Excepciones	Si hay algún error la aplicación no comparte el tweet y muestra la pantalla de inicio.
Notas	

5.5.15 Retwitear

Retwitear	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El tweet queda retwiteado en Twitter
Actores	Usuario registrado y API de Twitter.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en el tweet que quiere retwitear. 2. La aplicación lo retwitea en Twitter.
Excepciones	Si hay algún error la aplicación no retwitea el tweet y muestra la pantalla de inicio.
Notas	

5.5.16 Borrar tweet

Borrar tweet	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente y ha debido compartir algún tweet.
Poscondiciones	El usuario borra un tweet que ha compartido anteriormente.
Actores	Usuario registrado y API de Twitter.
Descripción	<ol style="list-style-type: none"> 3. El usuario hace click en el botón para borrar el tweet. 4. La aplicación elimina el tweet.
Excepciones	
Notas	

5.5.17 Cerrar sesión Twitter

Cerrar sesión Twitter	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La autorización de Twitter queda borrada de la base de datos
Actores	Usuario registrado

Descripción	<ol style="list-style-type: none"> 1. El usuario hace click “cerrar sesión”. 2. La aplicación elimina la autorización de Twitter de la base de datos. 3. La aplicación muestra la pantalla de inicio de Twitter.
Notas	

5.5.18 Ver videos recomendados

Ver videos recomendados	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza los videos recomendados por Youtube.
Actores	Usuario registrado y API de Youtube.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña “Recomendados”. 2. La aplicación muestra los videos recomendados por Youtube.
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.19 Ver videos más vistos

Ver videos más vistos	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza los videos más vistos en España en Youtube.
Actores	Usuario registrado y API de Youtube.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña “Más vistos”. 2. La aplicación muestra los videos más vistos en España por Youtube.
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.20 Ver videos más comentados

Ver videos más comentados	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza los videos más comentados en Youtube.
Actores	Usuario registrado y API de Youtube.

Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la pestaña "Trending". 2. La aplicación muestra los videos más comentados en Youtube.
Variaciones (escenarios secundarios)	
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error.
Notas	

5.5.21 Buscar videos

Buscar videos	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza los videos que ha buscado en Youtube.
Actores	Usuario registrado y API de Youtube.
Descripción	<ol style="list-style-type: none"> 1. El usuario introduce el término a buscar y hace click en "Buscar". 2. La aplicación muestra los videos que ha buscado el usuario en Youtube.
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error. • No hay resultados para la búsqueda. <ul style="list-style-type: none"> ○ Mostrar mensaje de error
Notas	

5.5.22 Reproducir videos

Reproducir videos	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La aplicación reproduce el video que el usuario seleccione
Actores	Usuario registrado y API de Youtube.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en el vídeo que desea reproducir. 2. La aplicación reproduce el video.
Variaciones (escenarios secundarios)	
Excepciones	Las excepciones posibles son controladas por Youtube
Notas	

5.5.23 Buscar canción

Buscar canción	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza las canciones que ha buscado en Spotify.
Actores	Usuario registrado y API de Spotify.
Descripción	<ol style="list-style-type: none"> 1. El usuario introduce el término a buscar y hace click en "Buscar". 2. La aplicación muestra las canciones que ha buscado el usuario en Spotify.
Variaciones (escenarios secundarios)	
Excepciones	<ul style="list-style-type: none"> • No se puede obtener los datos: <ul style="list-style-type: none"> ○ Mostrar mensaje de error. • No hay resultados para la búsqueda. <ul style="list-style-type: none"> ○ Mostrar mensaje de error
Notas	

5.5.24 Ver portada del álbum de la canción

Ver portada del álbum de la canción	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	El usuario visualiza la portada del álbum de la canción.
Actores	Usuario registrado y API de Spotify.
Descripción	<ol style="list-style-type: none"> 3. El usuario hace click en la canción que desee ver la portada. 4. La aplicación muestra la portada del video de la canción.
Variaciones (escenarios secundarios)	
Excepciones	Las excepciones son controladas por Spotify.
Notas	

5.5.25 Reproducir canción

Reproducir canción	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La aplicación reproduce la canción que el usuario seleccione
Actores	Usuario registrado y API de Spotify.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la canción que desea reproducir.

	2. La aplicación reproduce la canción.
Variaciones (escenarios secundarios)	
Excepciones	Las excepciones posibles son controladas por Spotify.
Notas	

5.5.26 Compartir en Facebook

Compartir en Facebook	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La aplicación comparte la canción que el usuario desee en Facebook.
Actores	Usuario registrado, API de Spotify y API de Facebook.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la canción que desea compartir. 2. La aplicación comparte la canción en Facebook.
Variaciones (escenarios secundarios)	
Excepciones	Las excepciones posibles son controladas por Facebook.
Notas	

5.5.27 Compartir en Twitter

Compartir en Twitter	
Precondiciones	El usuario ha tenido que loguearse en Flypper previamente.
Poscondiciones	La aplicación comparte la canción que el usuario desee en Twitter.
Actores	Usuario registrado, API de Spotify y API de Twitter.
Descripción	<ol style="list-style-type: none"> 1. El usuario hace click en la canción que desea compartir. 2. La aplicación comparte la canción en Twitter.
Variaciones (escenarios secundarios)	
Excepciones	Las excepciones posibles son controladas por Twitter.
Notas	

5.6 Análisis de Interfaces de Usuario

5.6.1 Descripción de la Interfaz

En la aplicación existirán dos tipos de interfaces, una para usuarios anónimos y otra para usuarios registrados.

Dependiendo de si el usuario está autenticado o no, se mostrará un menú con unas características u otras. Además, a los usuarios logueados, se les permitirá utilizar los diferentes módulos de la aplicación.

- Pantalla de usuario anónimo

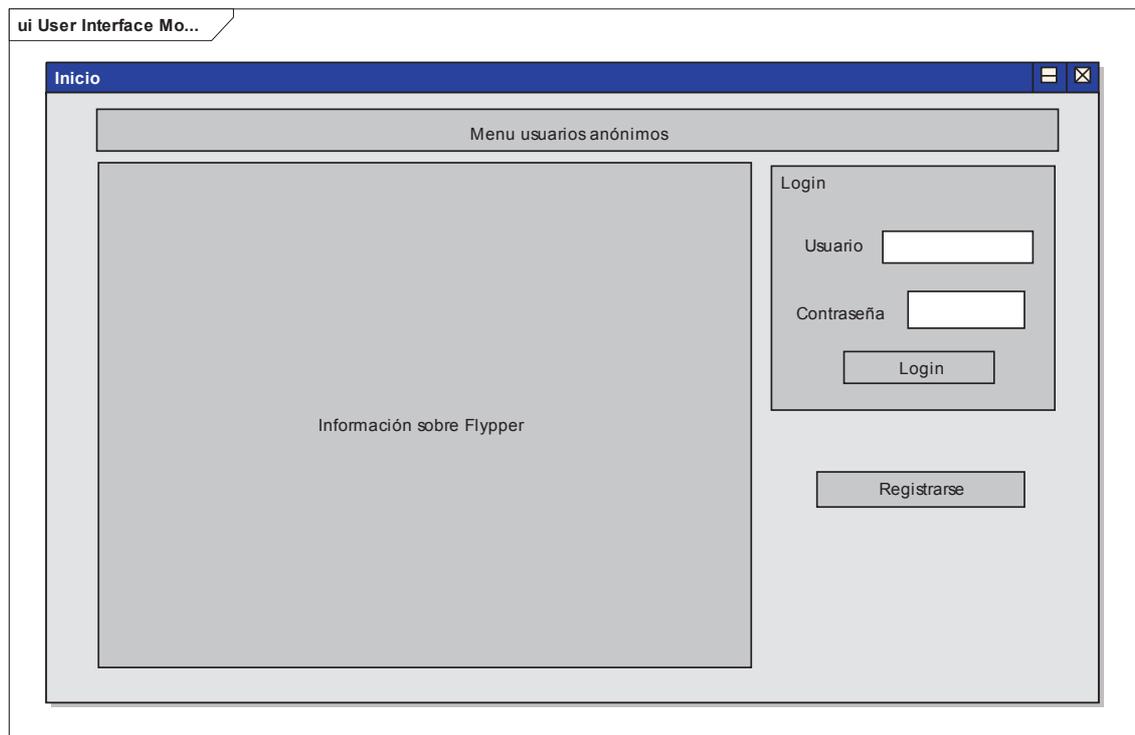


Figura 5.12. Boceto de la interfaz de inicio

- Pantalla de usuario registrado

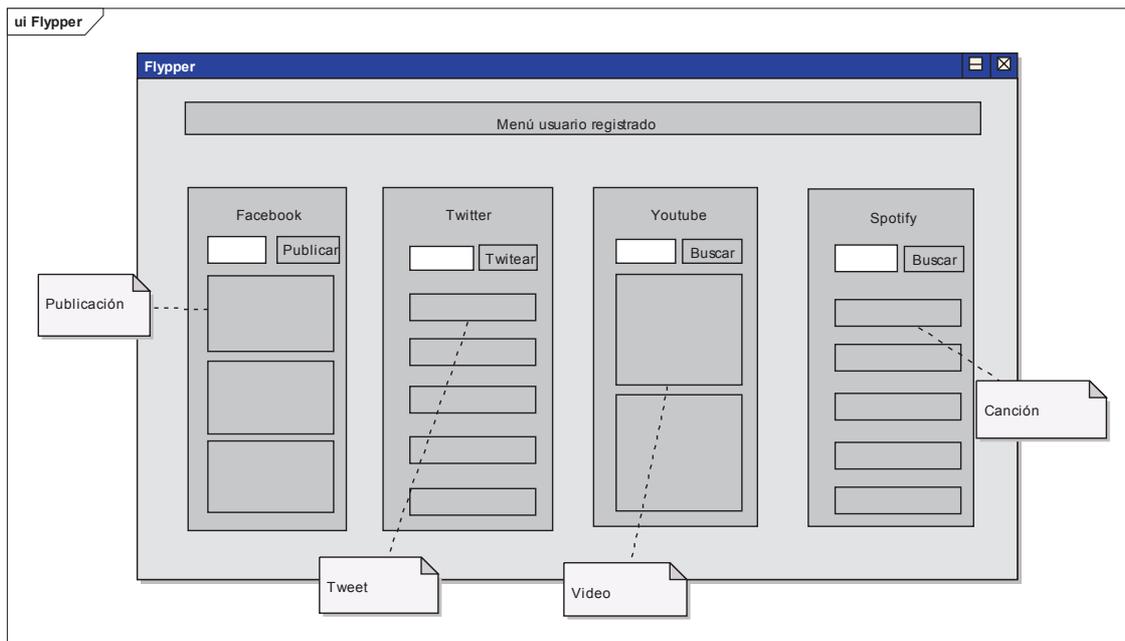


Figura 5.13. Boceto de la interfaz de la pantalla del usuario registrado

5.6.2 Descripción del Comportamiento de la Interfaz

Los datos pertenecientes a campos de formularios que sean erróneos se resaltarán en color rojo acompañándose de un mensaje de error.

Una vez que el login sea correcto se le redirigirá al usuario a la pantalla donde se encuentran todos los módulos de la aplicación.

Cuando se realice una operación con éxito se mostrará un mensaje en verde para informar al usuario que la operación se ha realizado satisfactoriamente.

5.6.3 Diagrama de Navegabilidad

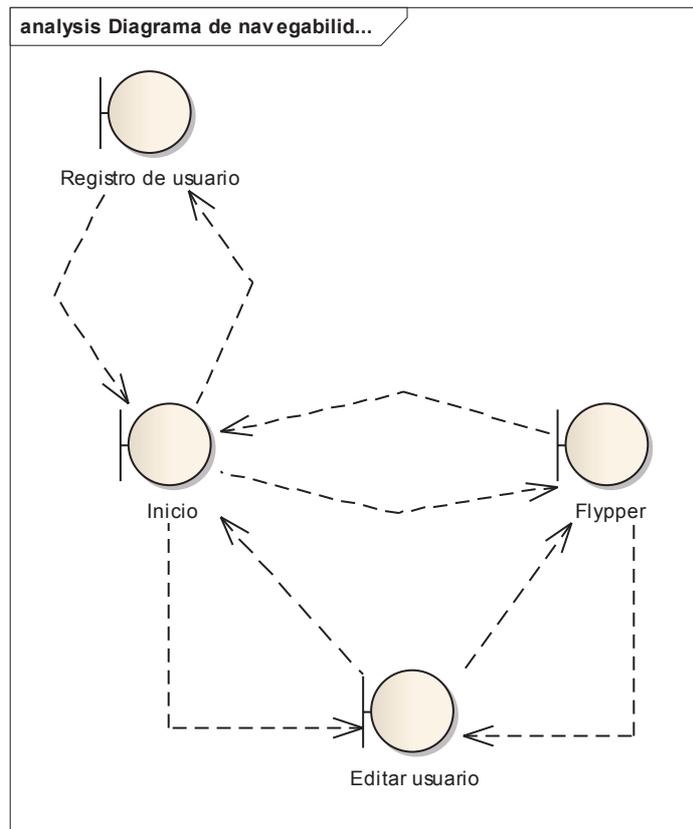


Figura 5.14. Diagrama de navegabilidad

5.7 Especificación del Plan de Pruebas

<i>Caso de Uso 1: Obtener permiso Facebook</i>	
Prueba	Resultado Esperado
Hacer click en iniciar sesión en Facebook y obtener la autorización.	Queda guardada la autorización en la base de datos del sistema.
Prueba	Resultado Esperado
Ocorre algo en la autorización y da error.	Se muestra al usuario la pantalla de iniciar sesión en Facebook.

<i>Caso de Uso 2: Ver inicio Facebook</i>	
Prueba	Resultado Esperado
Hacer click en ver inicio en Facebook.	Ver el inicio del usuario en Facebook.

<i>Caso de Uso 3: Ver perfil Facebook</i>	
Prueba	Resultado Esperado
Hacer click en ver perfil en Facebook.	Ver el perfil del usuario en Facebook.

<i>Caso de Uso 4: Ver publicación</i>	
Prueba	Resultado Esperado
Hacer click en una publicación.	Se despliega un apartado con los comentarios de la publicación y para poder hacer comentarios en la publicación.

<i>Caso de Uso 5: Publicar</i>	
Prueba	Resultado Esperado
Escribir algo para publicar y hacer click en "Publicar"	Aparece un "cargando" y se muestran todas las publicaciones, incluida la que se acaba de realizar.
Prueba	Resultado Esperado
No escribir nada para publicar y hacer click en "Publicar"	La aplicación no hace nada.

<i>Caso de Uso 6: Hacer me gusta</i>	
Prueba	Resultado Esperado
Hacer click en me gusta en una publicación.	El botón de "me gusta" cambia a "ya no me gusta".

<i>Caso de Uso 7: Comentar publicación</i>	
Prueba	Resultado Esperado
Escribir algo en una publicación.	La publicación se recarga y aparece el comentario.

Caso de Uso 8: Borrar publicación

Prueba	Resultado Esperado
Hacer click en la "x" que aparece en la publicación	La publicación queda eliminada.

Caso de Uso 9: Cerrar sesión

Prueba	Resultado Esperado
Hacer click en cerrar sesión en Facebook.	La autorización queda eliminada de la base de datos de la aplicación y al usuario se le muestra la pantalla de iniciar sesión en Facebook.

Caso de Uso 10: Obtener permiso Twitter

Prueba	Resultado Esperado
Hacer click en iniciar sesión en Twitter y obtener la autorización.	Queda guardada la autorización en la base de datos del sistema.
Prueba	Resultado Esperado
Ocorre algo en la autorización y da error.	Se muestra al usuario la pantalla de iniciar sesión en Twitter.

Caso de Uso 11: Ver inicio Twitter

Prueba	Resultado Esperado
Hacer click en ver inicio en Twitter.	Ver el inicio del usuario en Twitter.

Caso de Uso 12: Ver perfil Twitter

Prueba	Resultado Esperado
Hacer click en ver perfil en Twitter.	Ver el perfil del usuario en Twitter.

Caso de Uso 13: Ver menciones

Prueba	Resultado Esperado
Hacer click en menciones en Twitter.	Ver las menciones del usuario en Twitter.

Caso de Uso 14: Twitear

Prueba	Resultado Esperado
Escribir algo para compartir y hacer click en "Tweet".	Aparece un "cargando" y se muestran todos los tweets incluida la que se acaba de realizar.
Prueba	Resultado Esperado
No escribir nada para publicar y hacer click en "Tweet".	La aplicación no hace nada.
Prueba	Resultado Esperado
Escribir un tweet de más de 140 caracteres.	La aplicación no permite escribir más de 140 caracteres.

<u>Caso de Uso 15: Retwitear</u>	
Prueba	Resultado Esperado
Hacer click en el icono de retwitear.	El mensaje queda retwiteado y se le informa al usuario con un mensaje.

<u>Caso de Uso 16: Borrar tweet</u>	
Prueba	Resultado Esperado
Hacer click en la "x" que aparece en el tweet	El tweet queda eliminado.

<u>Caso de Uso 17: Cerrar sesión</u>	
Prueba	Resultado Esperado
Hacer click en cerrar sesión en Twitter.	La autorización queda eliminada de la base de datos de la aplicación y al usuario se le muestra la pantalla de iniciar sesión en Twitter.

<u>Caso de Uso 18: Ver recomendados</u>	
Prueba	Resultado Esperado
Hacer click en ver recomendados en Youtube.	Se muestran los vídeos recomendados por Youtube.

<u>Caso de Uso 19: Ver más vistos</u>	
Prueba	Resultado Esperado
Hacer click en ver más vistos en Youtube.	Se muestran los vídeos más vistos en España en Youtube.

<u>Caso de Uso 20: Ver trending</u>	
Prueba	Resultado Esperado
Hacer click en ver trending en Youtube.	Se muestran los vídeos trending en Youtube.

<u>Caso de Uso 21: Buscar video</u>	
Prueba	Resultado Esperado
Realizar una búsqueda de un video en Youtube.	Aparece una lista de videos.
No introducir nada en el campo de búsqueda.	La aplicación no hace nada

<u>Caso de Uso 22: Reproducir video</u>	
Prueba	Resultado Esperado
Hacer click en un video para	El video se reproduce.

reproducirlo.	
---------------	--

Caso de Uso 23: Buscar canción

Prueba	Resultado Esperado
Realizo una búsqueda de una canción en Spotify.	Aparece una lista de canciones.
No introducir nada en el campo de búsqueda.	La aplicación no hace nada.

Caso de Uso 24: Ver portada

Prueba	Resultado Esperado
Hago click en una de las canciones.	Se despliega un apartado con la portada del álbum de la canción.

Caso de Uso 25: Reproducir canción

Prueba	Resultado Esperado
Hago click en el play de una canción para reproducirla con el Spotify cerrado.	Se abre automáticamente el Spotify y te reproduce la canción.
Hago click en el play de una canción para reproducirla con el Spotify abierto.	Se reproduce la canción.

Caso de Uso 26: Compartir en Facebook

Prueba	Resultado Esperado
Hacer click en compartir en Facebook.	La canción queda compartida en Facebook.

Caso de Uso 27: Compartir en Twitter

Prueba	Resultado Esperado
Hacer click en compartir en Twitter.	La canción queda compartida en Twitter.

Capítulo 6. Diseño del Sistema

6.1 Arquitectura del Sistema

La aplicación no está programada con orientación a objetos, por tanto vamos a considerar los archivos php como clases.

6.1.1 Diagramas de Paquetes

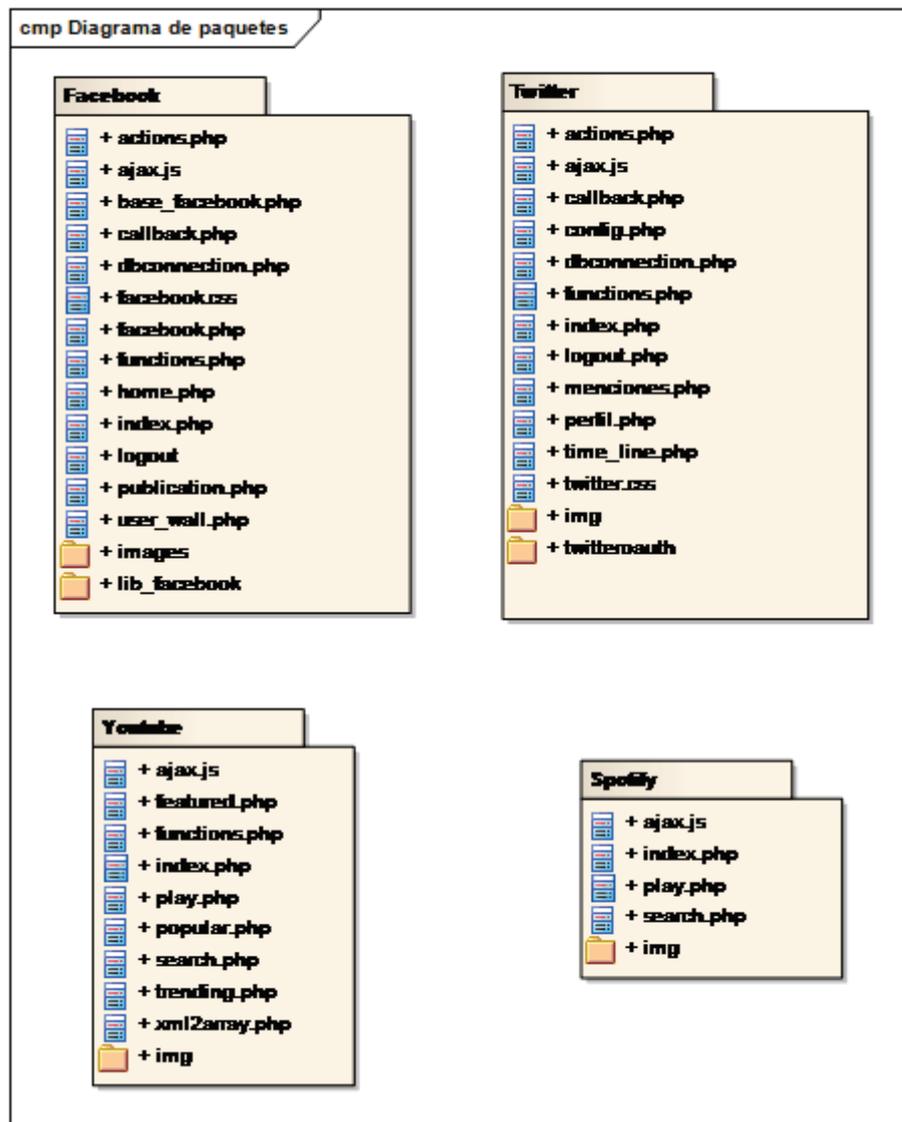


Figura 6.1. Diagrama de paquetes

6.1.1.1 Facebook

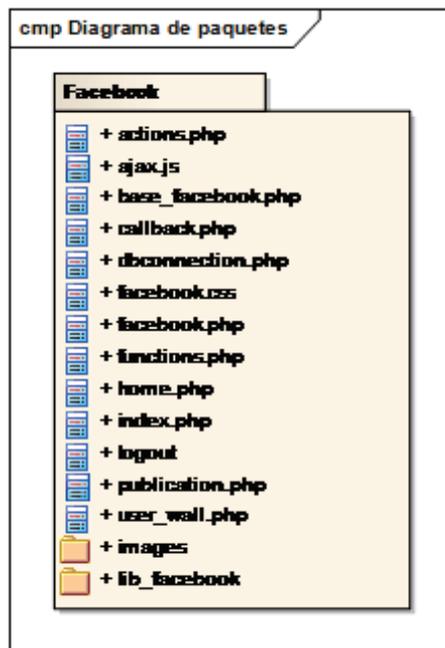


Figura 6.2. Paquete Facebook

Este paquete contiene todos los archivos necesarios para el funcionamiento del módulo de Facebook dentro de la aplicación.

6.1.1.2 Twitter

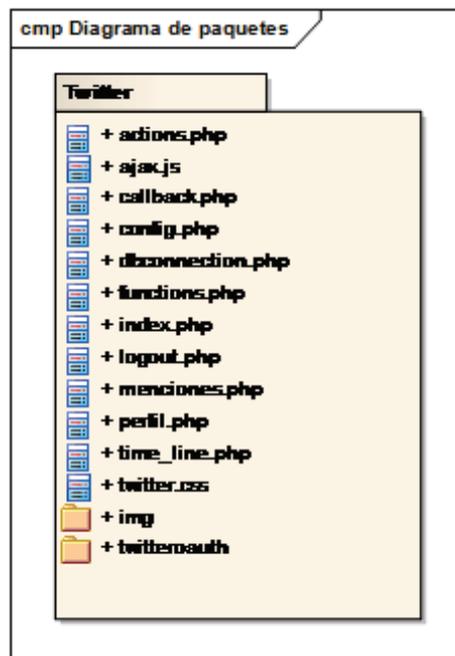


Figura 6.3. Paquete Twitter

Este paquete contiene todos los archivos necesarios para el funcionamiento del módulo de Twitter dentro de la aplicación.

6.1.1.3 Youtube

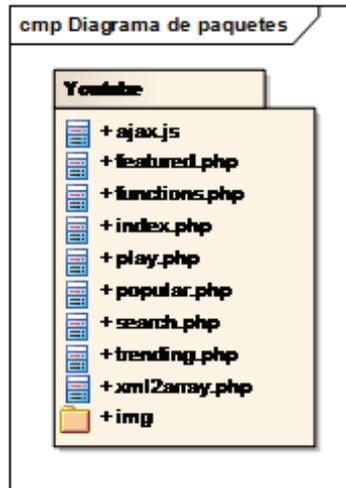


Figura 6.4. Paquete Youtube

Este paquete contiene todos los archivos necesarios para el funcionamiento del módulo de Youtube dentro de la aplicación.

6.1.1.4 Spotify

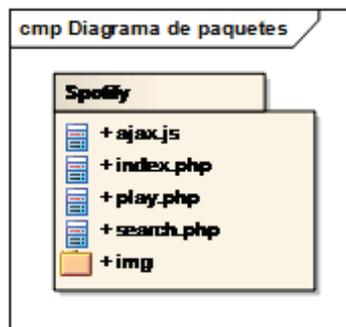


Figura 6.5. Paquete Spotify

Este paquete contiene todos los archivos necesarios para el funcionamiento del módulo de Spotify dentro de la aplicación.

6.1.2 Diagramas de Componentes

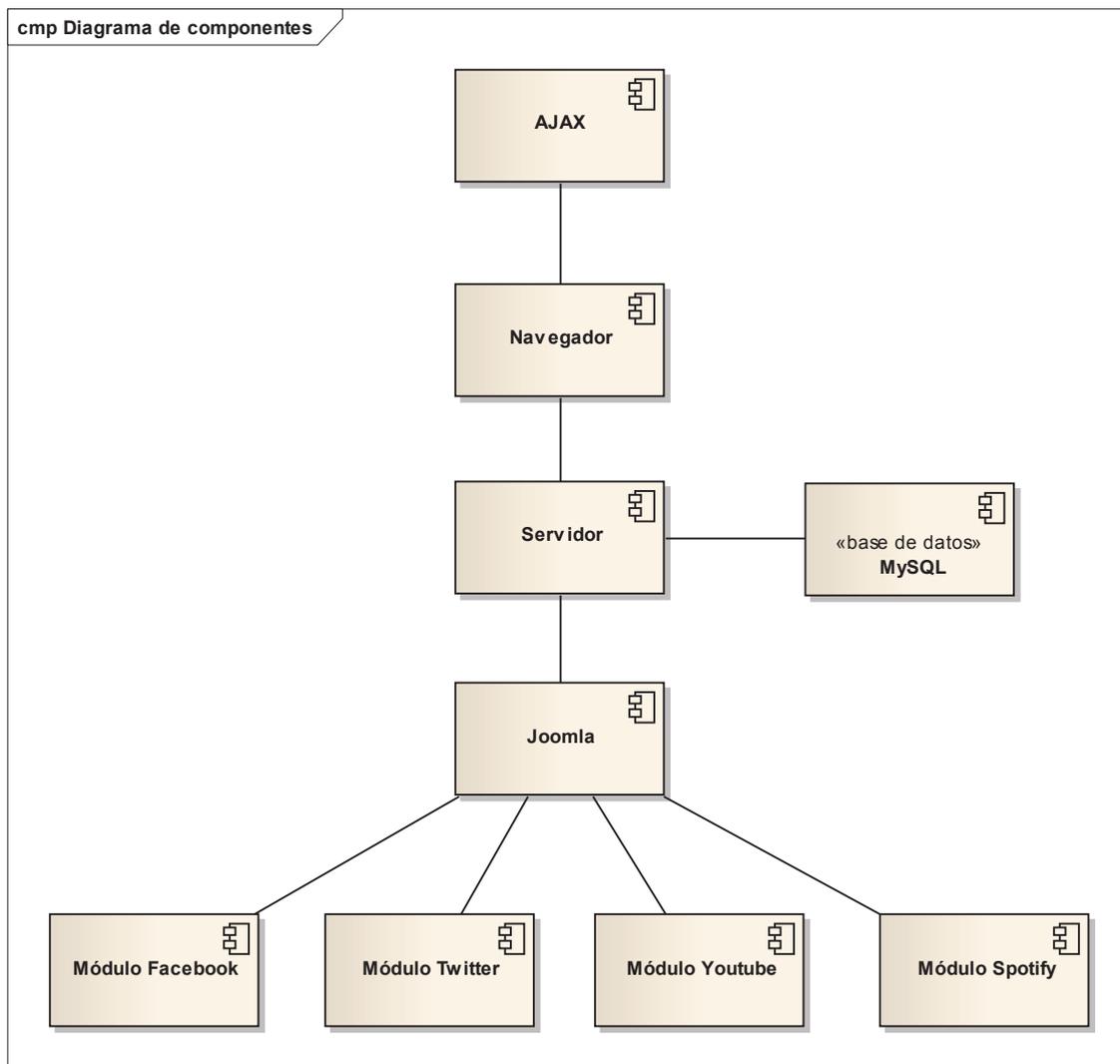


Figura 6.6. Diagrama de componentes

6.1.3 Diagramas de Despliegue

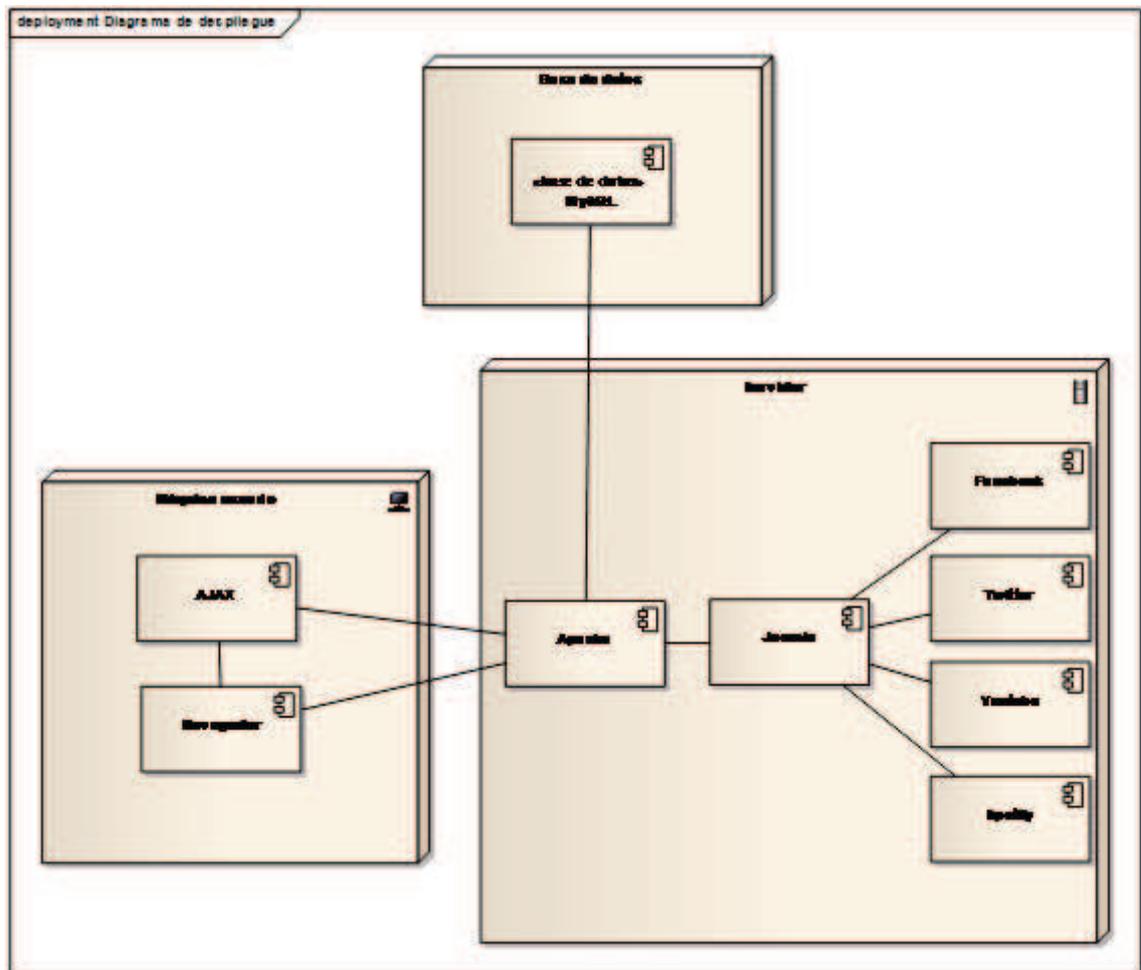


Figura 6.7. Diagrama de despliegue

6.1.3.1 Máquina usuario

Máquina desde donde el usuario utiliza la aplicación.

6.1.3.2 Servidor

Máquina donde se encuentra alojada y se ejecuta la aplicación.

6.1.3.3 Base de datos

Base de datos de la aplicación.

6.2 Diseño de Clases

6.2.1 Diagrama de Clases

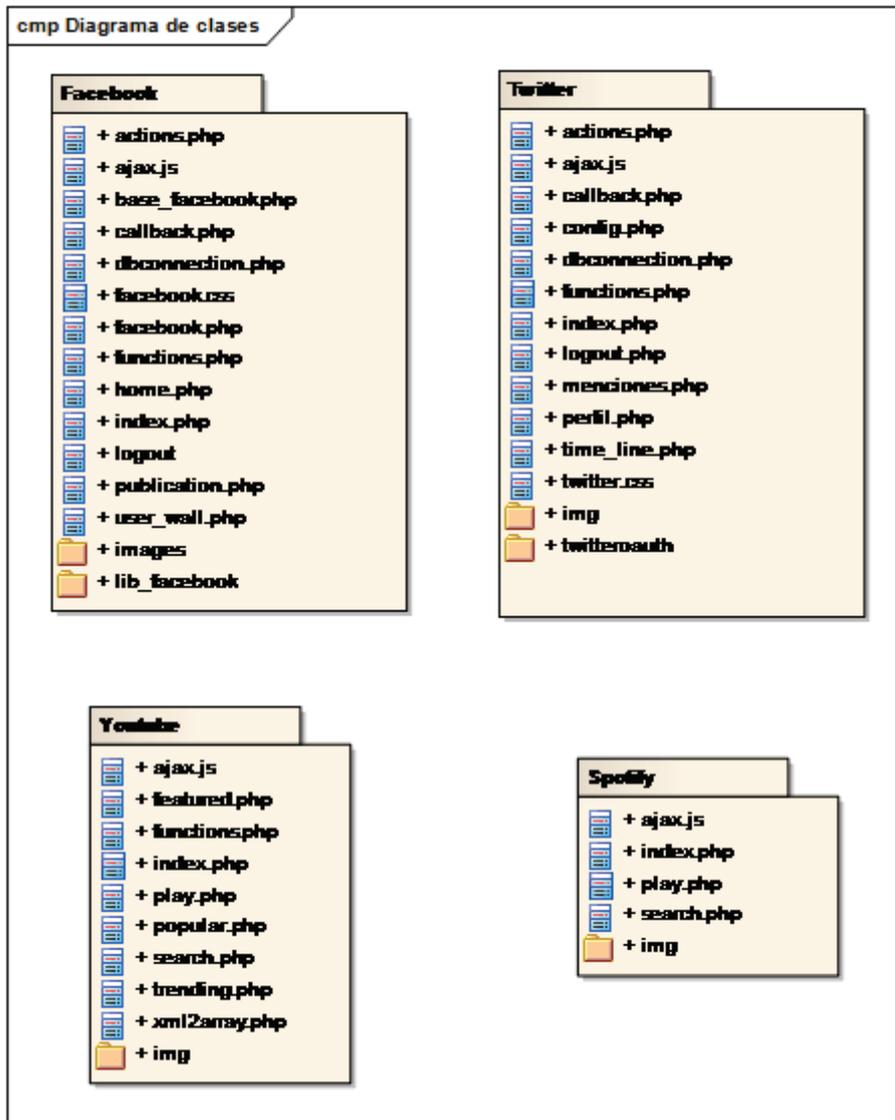


Figura 6.8. Diagrama de clases

6.3 Diagramas de Interacción y Estados

6.3.1 Obtener permisos Facebook

6.3.1.1 Diagrama de iteración

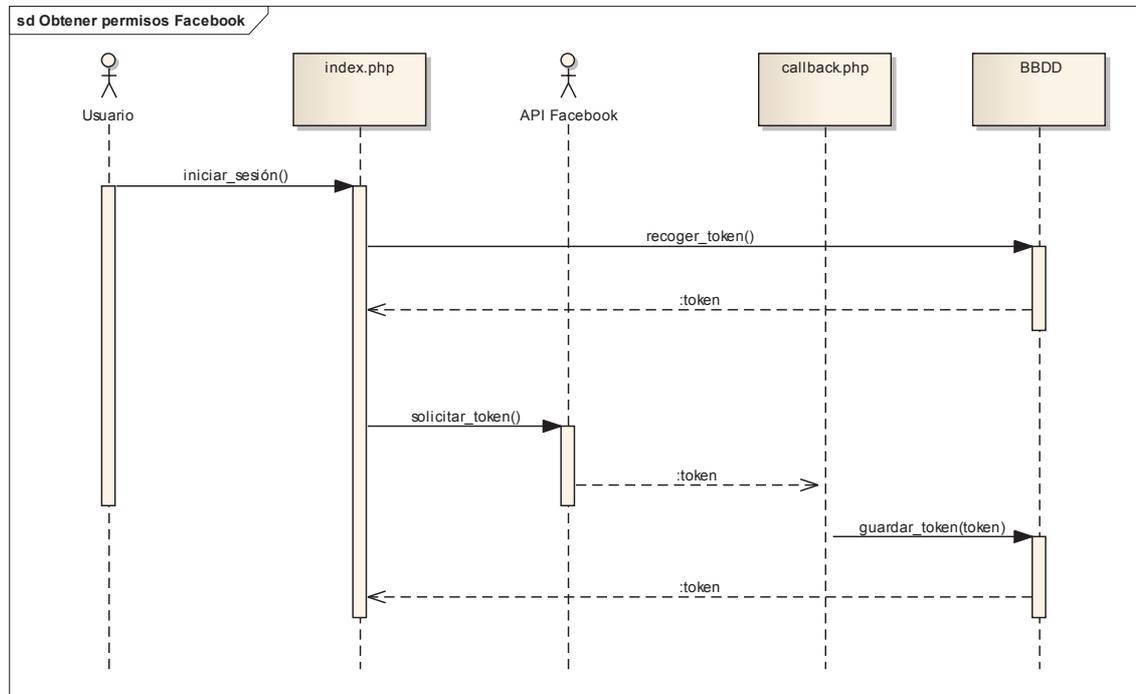


Figura 6.9. Diagrama de secuencia para obtener los permisos de Facebook

6.3.2 Ver inicio de Facebook

6.3.2.1 Diagrama de iteración

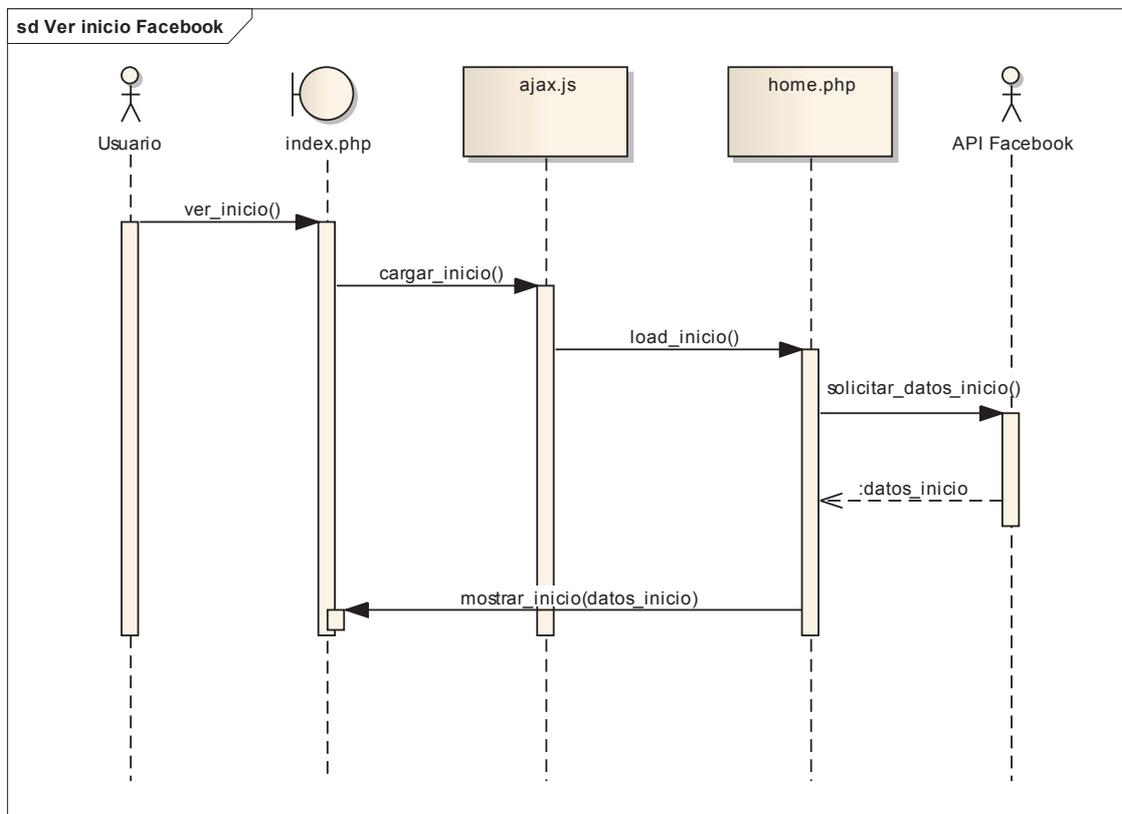


Figura 6.10. Diagrama de secuencia para ver el inicio de Facebook

6.3.3 Ver perfil de Facebook

El funcionamiento es el mismo que el del caso anterior “6.3.2 Ver inicio de Facebook” pero consultando el perfil en este caso.

6.3.4 Hacer “me gusta”

6.3.4.1 Diagrama de iteración

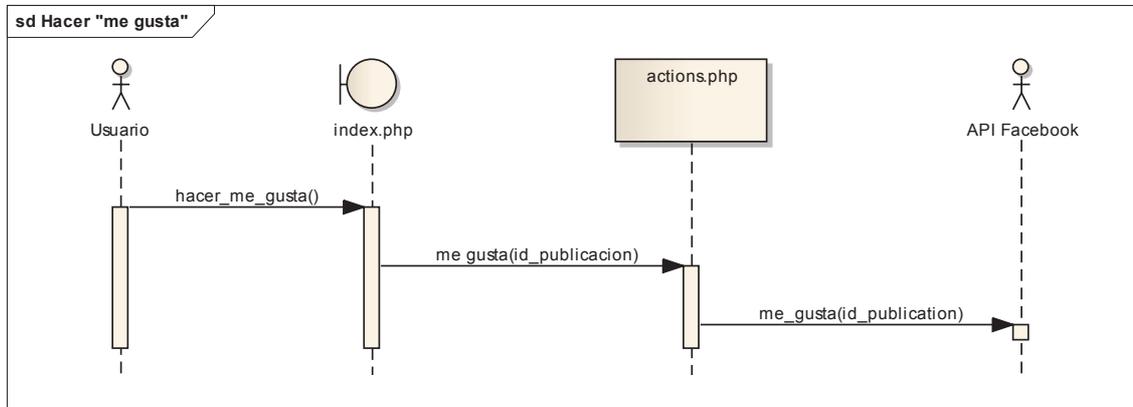


Figura 6.11. Diagrama de secuencia para hacer “me gusta”

6.3.5 Ver publicación

El funcionamiento es el mismo que el del caso “Ver inicio de Facebook” pero consultando los datos de la publicación.

6.3.6 Publicar

6.3.6.1 Diagrama de iteración

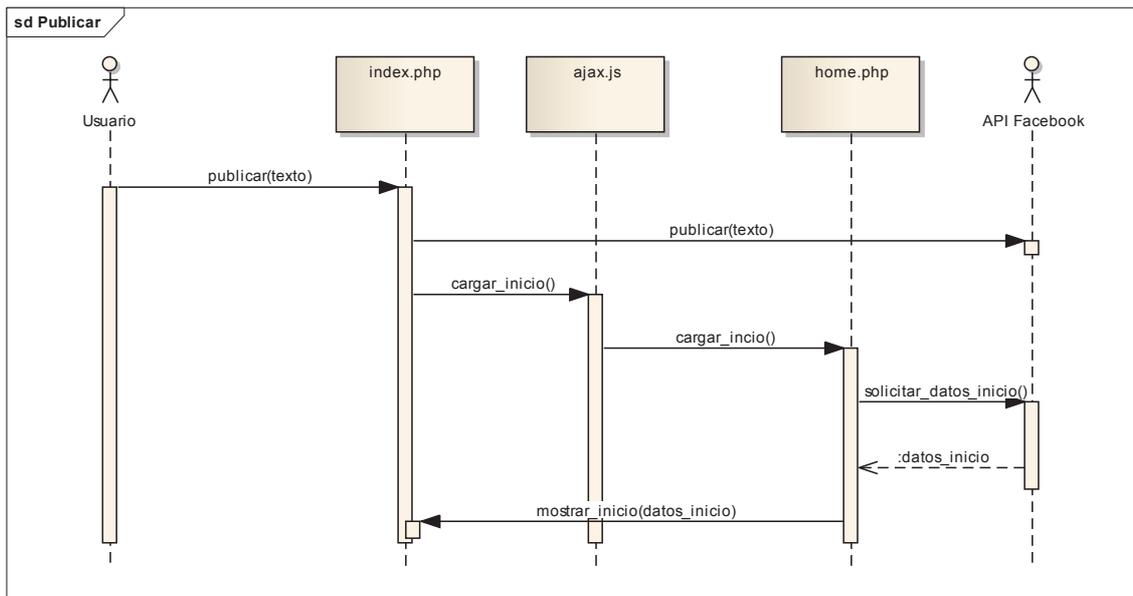


Figura 6.12. Diagrama de secuencia para publicar en Facebook

6.3.7 Comentar publicación

El funcionamiento es similar a “6.3.6 Publicar”.

6.3.8 Borrar publicación

6.3.8.1 Diagrama de iteración

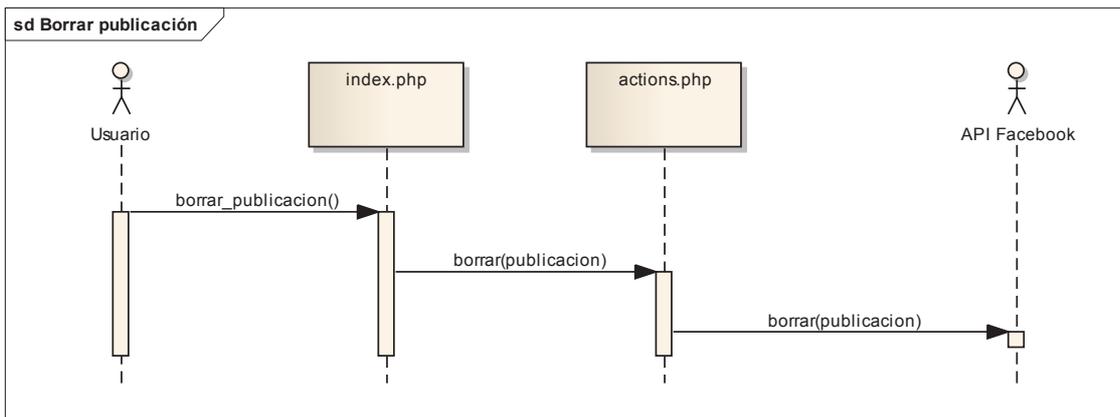


Figura 6.13. Diagrama de secuencia para borrar una publicación de Facebook

6.3.9 Cerrar sesión Facebook

6.3.9.1 Diagrama de iteración

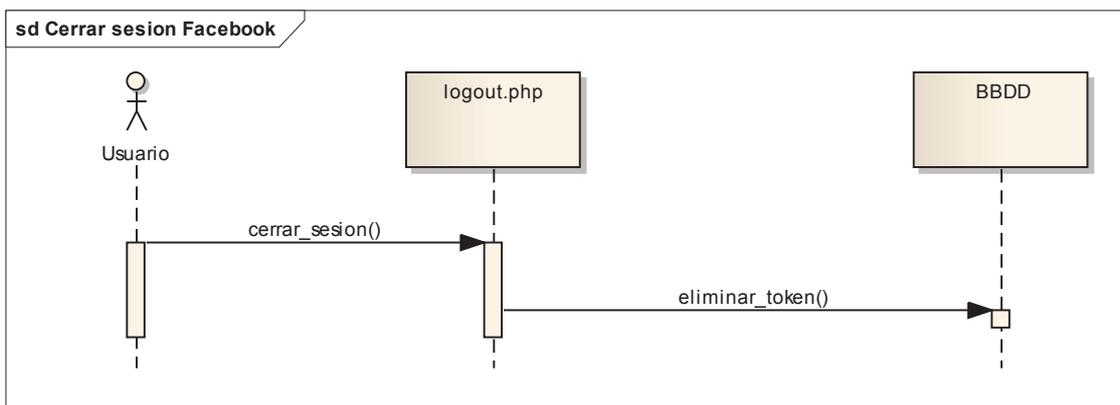


Figura 6.14. Diagrama de secuencia para cerrar sesión de Facebook

6.3.10 Obtener permisos Twitter

Funciona de la misma manera que “6.3.1 Obtener permisos Facebook” pero consultando al API de Twitter en vez de la de Facebook.

6.3.11 Ver inicio de Twitter

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Twitter.

6.3.12 Ver perfil de Twitter

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Twitter.

6.3.13 Ver menciones

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Twitter.

6.3.14 Twitear

El funcionamiento es el mismo que el del caso “6.3.6 Publicar” pero consultando el API de Twitter y ejecutando la acción de twitear.

6.3.15 Retwitear

El funcionamiento es el mismo que el del caso “6.3.6 Publicar” pero consultando el API de Twitter y ejecutando la acción de retwitear.

6.3.16 Borrar tweet

El funcionamiento es el mismo que el del caso “6.3.8 Borrar publicación” pero consultando el API de Twitter.

6.3.17 Cerrar sesión Twitter

El funcionamiento es el mismo que el del caso “6.3.9 Cerrar sesión Facebook”.

6.3.18 Ver videos recomendados

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Youtube y utilizando el archivo featured.php.

6.3.19 Ver videos más vistos

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Youtube y utilizando el archivo popular.php.

6.3.20 Ver videos más comentados

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Youtube y utilizando el archivo trending.php.

6.3.21 Buscar videos

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Youtube y utilizando el archivo search.php

6.3.22 Buscar canción

El funcionamiento es el mismo que el del caso “6.3.2 Ver inicio de Facebook” pero consultando el API de Spotify y utilizando el archivo search.php

6.4 Diseño de la Base de Datos

6.4.1 Descripción del SGBD Usado

Para el SGBD se ha utilizado MySQL en su versión 5.1.38. La base de datos está alojada en un servidor web.

Se ha elegido MySQL para la aplicación ya que es un producto con reconocido prestigio, fiabilidad, velocidad, rendimiento, facilidad de administración y conexión con otros productos, bien documentados, con una buena evolución y soporte. Este sistema gestor de bases de datos está siendo utilizado en muchos entornos productivos y nos da la suficiente confianza. Además es gratuito.

6.4.2 Integración del SGBD en Nuestro Sistema

Debido a que se utiliza Joomla como base del proyecto, creamos una base de datos MySQL e insertamos las tablas de Joomla en ella.

Joomla utiliza una base de datos relacional con numerosas tablas divididas en los siguientes bloques:

- Bloque de contenido: es el encargado de realizar la gestión del contenido de los artículos publicados en Joomla.
- Bloque de componentes, módulos y plugins: estas tablas realizan la gestión de todos los módulos, componentes y plugins instalados sobre el Joomla. Principalmente se encargan de guardar posiciones de los mismos y el estado en el que se encuentran (activado o desactivado).
- Bloque de plantillas: guarda el estado de las diferentes plantillas aplicables a Joomla.
- Bloque de menú: guarda el estado de los menús.
- Bloque de usuarios y control de acceso: bloque encargado de gestionar la disponibilidad, el orden y el contenido de los menús.
- Bloque de log y estadísticas: bloque que realiza el control de logs el sistema.

Los módulos de Facebook y Twitter solamente necesitan guardar los tokens de autorización, por lo tanto, se ha modificado la tabla jos_users perteneciente al bloque de usuarios y control de acceso, añadiendo 3 campos nuevos llamados `oauth_token`, `oauth_token_secret` y `oauth_token_fb`.

Además se han añadido un nuevo bloque de tablas correspondientes al componente que permite añadir, cerrar o arrastrar los módulos. Estas tablas permiten guardar las preferencias de los usuarios, en cuanto a la colocación o la utilización de unos módulos u otros, de manera individual.

Para conectar nuestros módulos con la base de datos se utiliza una conexión de MySQL para PHP.

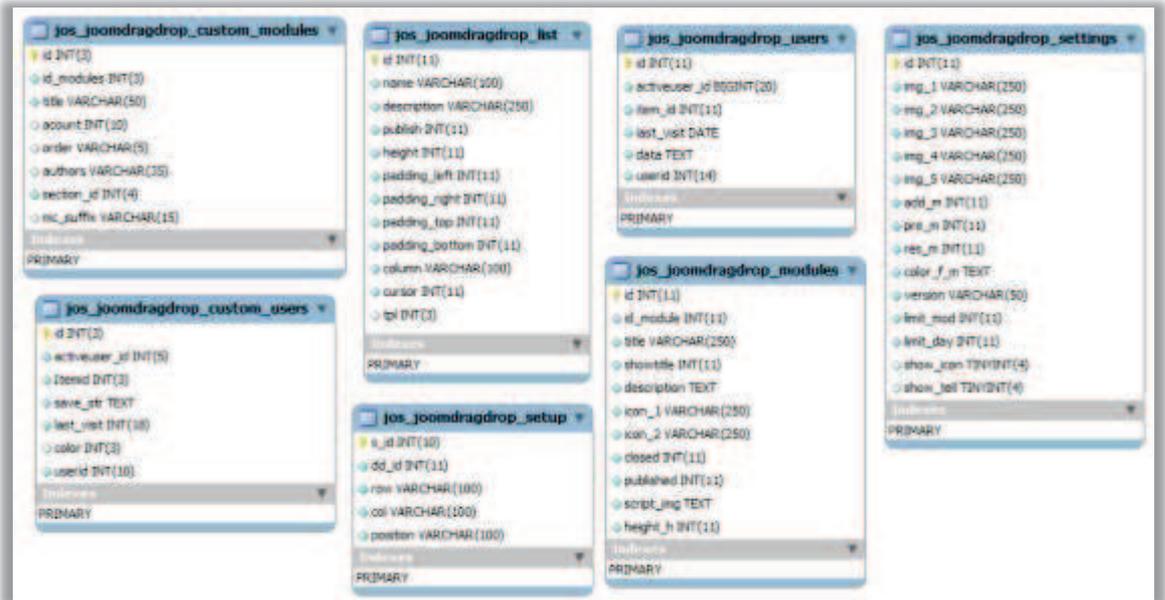


Figura 6.16. Tablas del componente Drag&Drop

6.5 Diseño de la Interfaz

6.5.1 Pantalla principal

Pantalla de presentación de Flypper. Representa la parte más comercial de la aplicación. En ella se describen algunas características que posee la web.



Figura 6.17. Pantalla principal Flypper

6.5.2 Pantalla de login

Para acceder a la aplicación el usuario deberá realizar el login.

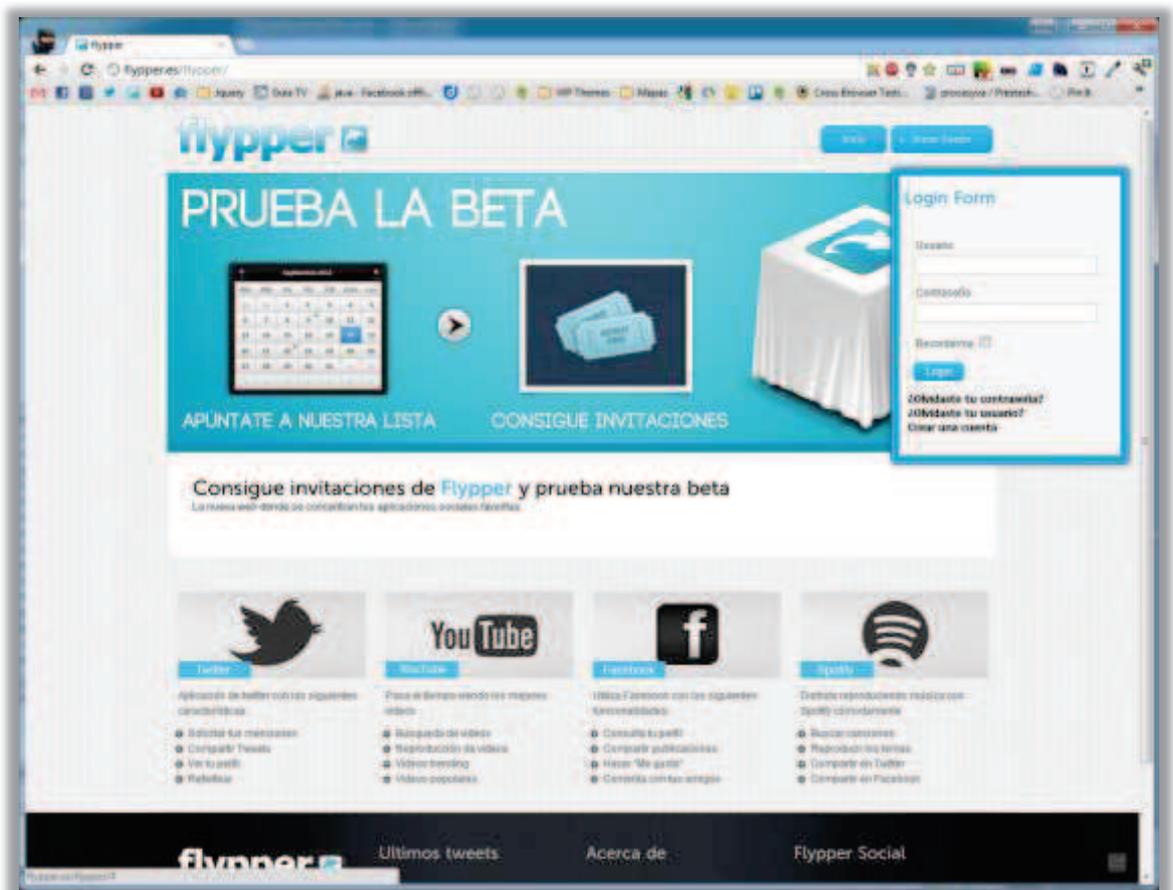


Figura 6.18. Pantalla de login

6.5.3 Pantalla Flypper

Una vez que el usuario se haya logueado, la aplicación mostrará Flypper con toda su funcionalidad.



Figura 6.19. Pantalla de Flypper

6.6 Especificación Técnica del Plan de Pruebas

Las pruebas se han llevado a cabo en la siguiente máquina:

- Sistema operativo: Windows 7 Home Premium 64 bits.
- Procesador: Intel Core i7 2670QM 2.2GHz.
- Memoria RAM: 4 Gb DDR3

Todas ellas se han realizado a través de internet, mediante el navegador Google Chrome.

6.6.1 Pruebas Unitarias

Una prueba unitaria es una forma de probar el correcto funcionamiento de un módulo de código, o en este caso una clase individual que cumple con una función concreta. Esto sirve para asegurar que cada uno de los módulos funcione correctamente por separado.

En este apartado se realizarán pruebas sobre las clases más importantes dentro de cada uno de los módulos que componen la aplicación.

6.6.2 Pruebas de Usabilidad y Accesibilidad

Las pruebas de usabilidad tratan de evaluar la aplicación en su funcionamiento real. Debemos realizar un conjunto de pruebas para verificar que las distintas partes del programa se pueden usar adecuadamente.

Se han distinguido varios tipos de usuario:

- Usuarios que tengan experiencia con redes sociales.
- Usuarios que no tengan experiencia con redes sociales.

Las pruebas se han realizado con un ordenador portátil, por lo que el lugar ha sido diferente, eso sí, con conexión a internet.

Para la realización de estas pruebas se realizarán unos cuestionarios y unas pruebas guiadas con la que evaluaremos la aplicación en su funcionamiento real.

6.6.2.1 Diseño de Cuestionarios

6.6.2.1.1 Cuestionario de Evaluación

Elaboramos un cuestionario para que los usuarios lo realicen y determinar así distintos aspectos del mismo y de su interacción con la aplicación. Los puntos a tocar son:

- **1º: Preguntas de carácter general** a través de las cuales se determine el tipo de usuario y su nivel de conocimiento informático.
- **2º: Actividades guiadas** para hacer con nuestra aplicación.

- **3º: Batería de preguntas cortas** con los distintos aspectos de la aplicación que se pretendan evaluar.
- **4º: Observaciones**, para que el usuario aporte todo lo que considere oportuno de nuestra aplicación.

6.6.2.1.2 Cuestionario para el Responsable de las Pruebas

Además, vamos a desarrollar un cuestionario para que el responsable de las pruebas pueda anotar distintos aspectos que observe durante la realización de las pruebas.

6.6.2.2 *Actividades de las Pruebas de Usabilidad*

6.6.2.2.1 Preguntas de carácter general

¿Usa un ordenador frecuentemente?
<ol style="list-style-type: none">1. Todos los días2. Varias veces a la semana3. Ocasionalmente4. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
<ol style="list-style-type: none">1. Es parte de mi trabajo o profesión2. Lo uso básicamente para ocio3. Solo empleo aplicaciones estilo Office4. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez aplicaciones web de redes sociales ?
<ol style="list-style-type: none">1. Sí, he empleado aplicaciones similares2. No, aunque si empleo otros programas que me ayudan a realizar tareas similares3. No, nunca
¿Qué busca Vd. Principalmente en un programa?
<ol style="list-style-type: none">1. Que sea fácil de usar2. Que sea intuitivo3. Que sea rápido4. Que tenga todas las funciones necesarias5. Que tenga buen diseño

6.6.2.2.2 Actividades guiadas

Hacer un compendio de actividades que se puedan hacer con la aplicación para que nuestros usuarios las hagan y comenten los problemas y dificultades que según su opinión presenta la aplicación. Posibles actividades a probar son:

- Loguearse en el módulo de Facebook.
- Ver perfil del usuario de Facebook.
- Compartir una publicación.
- Hacer me gusta en una publicación.
- Eliminar una publicación.
- Loguearse en el módulo de Twitter.
- Twitear.
- Ver las menciones del usuario.
- Borrar un tweet.
- Retwitear.
- Buscar un video en Youtube.
- Reproducir el video en Youtube.
- Buscar una canción en Spotify.
- Reproducir una canción en Spotify.
- Compartir la canción en Facebook.

6.6.2.2.3 Preguntas Cortas sobre la Aplicación y Observaciones

Realizaremos el siguiente cuestionario de preguntar cortas:

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Sabe donde está dentro de la aplicación?</i>				
<i>¿Existe ayuda para las funciones en caso de que tenga dudas?</i>				
<i>¿Le resulta sencillo el uso de la aplicación?</i>				
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
<i>¿Funciona cada tarea como Vd. Espera?</i>				
<i>¿El tiempo de respuesta de la aplicación es muy grande?</i>				
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
<i>El tipo y tamaño de letra es</i>				
<i>Los iconos e imágenes usados son</i>				
<i>Los colores empleados son</i>				
Diseño de la Interfaz	Si	No	A veces	
<i>¿Le resulta fácil de usar?</i>				
<i>¿El diseño de las pantallas es claro y atractivo?</i>				
<i>¿Cree que el programa está bien estructurado?</i>				
Observaciones				
Cualquier comentario del usuario				

6.6.2.2.4 Cuestionario para el Responsable de las Pruebas

Aspecto Observado	Notas
<i>El usuario comienza a trabajar de forma rápida por las tareas</i>	
<i>Tiempo en realizar cada tarea</i>	
<i>Errores leves cometidos</i>	
<i>Errores graves cometidos</i>	
<i>Encuentran rápidamente lo que se les pide</i>	
<i>El usuario está satisfecho con la aplicación.</i>	

6.6.2.3 Pruebas de Accesibilidad

Para la realización de las pruebas de accesibilidad seguiremos las pautas que marcan las normas WCAG. Se procederá a realizar una revisión preliminar para identificar los posibles problemas de accesibilidad que podamos encontrar; y una evaluación de conformidad para comprobar si nuestro sitio web cumple con los estándares de accesibilidad WCAG.

6.6.3 Pruebas de Rendimiento

El sistema desarrollado consumirá una determinada cantidad de recursos que se debe procurar que sean los menores posibles.

Se debe pues medir el tiempo que tardan las operaciones más importantes realizadas por Flypper y comprobar si es posible mejorarlas de algún modo.

Si se descubre algún fragmento de código con tiempos de ejecución muy grandes se tratará de remediar esta situación.

Capítulo 7. Implementación del Sistema

7.1 Estándares y Normas Seguidos

7.1.1 CSS 3

Estándar del W3C, encargado de formular la especificación de las hojas de estilo, que sirve para dar estilo a los documentos html.

Se ha comprobado su cumplimiento en el validador de CSS del W3C.



Figura 7.1. CSS válido

7.1.2 XHTML 1.0 Transitional

Estándar del W3C, encargado de formular la especificación del lenguaje de marcado, que sirve para formar una estructura en la página web e introducir el contenido de la misma.

Se ha comprobado su cumplimiento en el validador de CSS del W3C.

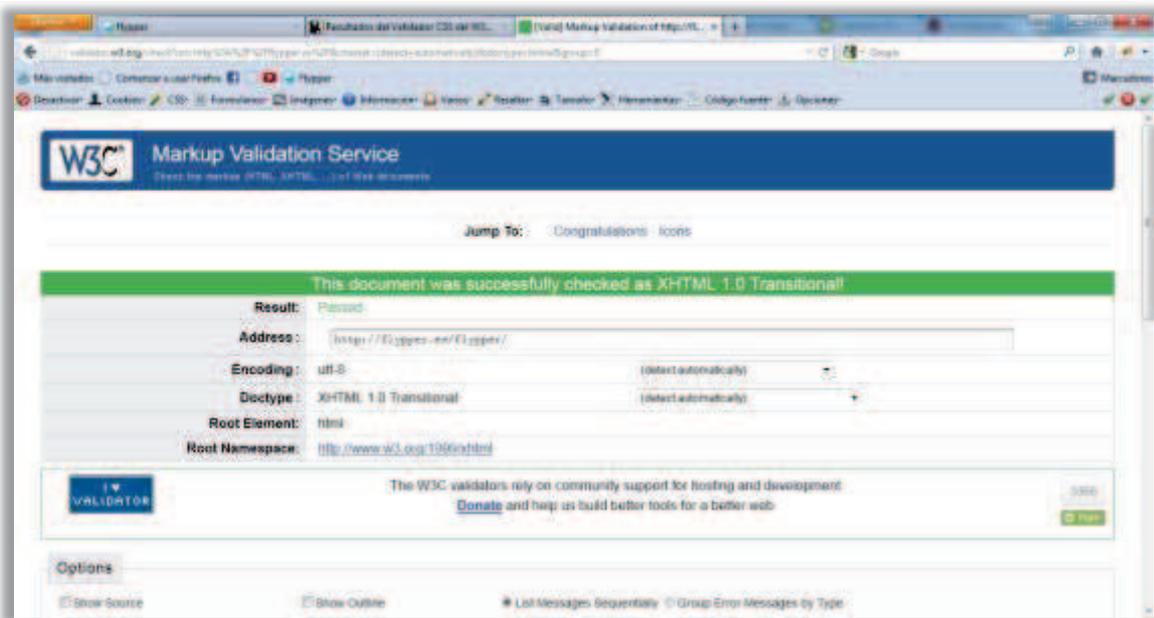


Figura 7.2. HTML válido

7.2 Lenguajes de Programación

7.2.1 HTML

HTML5 [**html5**] (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: un «clásico» HTML (text/html), la variante conocida como HTML5 y una variante XHTML conocida como sintaxis XHTML5 que deberá ser servida como XML (XHTML) (application/xhtml+xml).

7.2.2 CSS

CSS [**css**] (Cascading Style Sheet) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML). Se ha utilizado en su versión 3.

7.2.3 PHP

PHP [**php**] es un lenguaje de programación interpretado (Lenguaje de alto rendimiento), diseñado originalmente para la creación de páginas web dinámicas. Se usa principalmente para la interpretación del lado del servidor (server-side scripting).

7.2.4 JavaScript

JavaScript [**javascript**] es un lenguaje de programación que se utiliza para crear páginas web dinámicas, ya que sus programas se prueban directamente en un navegador sin necesidad de realizar procesos intermedios.

7.2.5 jQuery

jQuery [**jquery**] es una biblioteca de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web.

7.2.6 AJAX

AJAX, [**ajax**] acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

7.2.7 cURL

cURL [**curl**] es una herramienta para usar en un intérprete de comandos para transferir archivos con sintaxis URL, soporta FTP, FTPS, HTTP,HTTPS, TFTP, SCP, SFTP, Telnet, DICT, FILE y LDAP. Se ha utilizado para realizar consultas sobre las diferentes APIs.

7.2.8 SDK Facebook para PHP

Software Development Kit [**SDK.Facebook**] (Kit de desarrollo de software) de Facebook oficial para PHP. Proporciona funcionalidad para realizar llamadas al API de Facebook.

7.2.9 Librería OAuth Twitter para PHP

Librería [**Librería.Twitter**] OAuth de Twitter para PHP. Creada por Abraham Williams. Se ha utilizado para interactuar con el API de Twitter en la versión 0.2.

7.3 Herramientas y Programas Usados para el Desarrollo

7.3.1 NetBeans

NetBeans [**netbeans**] es un entorno de desarrollo integrado libre, gratuito y sin restricciones de uso. Existe además un número importante de módulos para extenderlo. Se ha utilizado la versión 7.1.1 para el desarrollo de la aplicación.

7.3.2 Enterprise Architect 7

Enterprise Architect [**enterprise**] es una herramienta de modelado UML potente y completa. Permite realizar diagramas UML, así como una amplia gama de características para el desarrollo de software, incluyendo modelado de bases de datos, requisitos, diseño de interfaces, presentación de informes HTML...

La versión utilizada ha sido 7.5.844

7.3.3 Microsoft Office 2007

Microsoft Office 2007 [**office**] es una herramienta ofimática utilizada para el desarrollo de la documentación final de este proyecto. Principalmente se ha hecho uso de las herramientas Microsoft Word, Microsoft Excel y Microsoft Power Point.

7.3.4 HeidiSQL

HeidiSQL [**heidi**] es interfaz gráfica para SQL que te permite gestionar tus bases de datos de forma cómoda e intuitiva.

7.3.5 Sublime Text 2

Sublime Text 2 [**sublime**] es un editor de código al estilo TextMate, Kate o Redcar, su interfaz es limpia e intuitiva y soporta el uso de Snippets, Plugins y sistemas de construcción de código. Se ha utilizado la versión 2 para realizar cambios en los archivos de manera ágil.

7.3.6 Google Chrome

Navegador web [**chrome**] desarrollado por Google de código abierto. Posee herramientas para desarrolladores web. Se ha utilizado la versión 19.0.1084.56 para la realización de las pruebas.

7.3.7 FileZilla

Filezilla [**filezilla**] es un cliente FTP gratuito. Se ha utilizado la versión 3.5.3 para subir los archivos de la aplicación al servidor donde se aloja la aplicación.

7.4 Creación del Sistema

7.4.1 Descripción Detallada de las Clases

La aplicación está formada por 4 módulos totalmente diferenciados. A continuación se describirá de qué está compuesto cada módulo y para qué sirve cada una de sus partes.

7.4.2 Facebook

7.4.2.1 *lib_facebook*

Es una librería en php que se encarga de la creación de un objeto de Facebook que nos facilita la tarea a la hora de obtener los códigos de autenticación, realizar acciones contra el API u obtener las URLs necesarias para conseguir los permisos.

7.4.2.2 *index.php*

Archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.

7.4.2.3 *user_wall.php*

Archivo que obtiene y muestra los datos del perfil de Facebook del usuario. En él se puede hacer me gusta en las diferentes publicaciones, borrarlas o comentarlas.

7.4.2.4 *home.php*

Archivo que obtiene y muestra los datos del perfil de Facebook del usuario. En él se puede hacer me gusta en las diferentes publicaciones, borrarlas o comentarlas.

7.4.2.5 *publication.php*

Archivo para mostrar los comentarios de una publicación. Además permite comentar en la misma. En él se puede hacer me gusta en las diferentes publicaciones, borrarlas o comentarlas.

7.4.2.6 *logout.php*

Archivo para realizar la desconexión de la aplicación. Además elimina de la base de datos el token de autorización del usuario.

7.4.2.7 *actions.php*

Archivo donde se encuentran las diferentes acciones que se pueden realizar en la aplicación (me gusta, publicar, comentar, borrar, etc.).

7.4.2.8 *ajax.js*

Archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Facebook sin tener que recargar la página.

7.4.2.9 *callback.php*

Archivo encargado de recoger los datos que Facebook devuelve una vez realizada la petición de autorización. Además guarda el token de autorización en la base de datos para no tener que volver a pedir al usuario la autorización.

7.4.2.10 *dbconnection.php*

Archivo que realiza la conexión a la base de datos.

7.4.2.11 *functions.php*

Archivo donde se encuentra la función para parsear las URLs de la aplicación.

7.4.2.12 *facebook.css*

Archivo de estilo para el módulo de Facebook.

7.4.3 Twitter

7.4.3.1 *twitteroauth*

Es una librería en php que se encarga de la creación de un objeto de Twitter que nos facilita la tarea a la hora de obtener los códigos de autenticación, realizar acciones contra el API u obtener las URLs necesarias para conseguir los permisos.

7.4.3.2 *index.php*

Archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.

7.4.3.3 *time_line.php*

Archivo que obtiene y muestra los datos del time line de Twitter del usuario. En él se puede eliminar tus propios tweets o retwitear los tweets de otros usuarios.

7.4.3.4 *perfil.php*

Archivo que obtiene y muestra los datos del time line de Twitter del usuario. En él se puede eliminar tus tweets.

7.4.3.5 *menciones.php*

Archivo que obtiene y muestra las menciones del usuario en Twitter.

7.4.3.6 *logout.php*

Archivo para realizar la desconexión de la aplicación. Además elimina de la base de datos el token de autorización del usuario.

7.4.3.7 *actions.php*

Archivo donde se encuentran las diferentes acciones que se pueden realizar en la aplicación (Twitear, borrar un tweet o retwitear).

7.4.3.8 *ajax.js*

Archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Twitter sin tener que recargar la página.

7.4.3.9 *callback.php*

Archivo encargado de recoger los datos que Twitter devuelve una vez realizada la petición de autorización. Además guarda el token de autorización en la base de datos para no tener que volver a pedir al usuario la autorización.

7.4.3.10 *dbconnection.php*

Archivo que realiza la conexión a la base de datos.

7.4.3.11 *functions.php*

Archivo donde se encuentra la función para parsear las URLs de la aplicación.

7.4.3.12 *twitter.css*

Archivo de estilo para el módulo de Twitter.

7.4.4 Youtube

7.4.4.1 *index.php*

Archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.

7.4.4.2 *featured.php*

Archivo que obtiene y muestra los videos recomendados por Youtube.

7.4.4.3 *popular.php*

Archivo que obtiene y muestra los videos más vistos en Youtube España.

7.4.4.4 *trending.php*

Archivo que obtiene y muestra los videos más comentados en el momento en Youtube.

7.4.4.5 *search.php*

Archivo encargado de realizar la búsqueda de videos y de mostrar los resultados obtenidos en Youtube.

7.4.4.6 *play.php*

Archivo encargado de realizar la reproducción del vídeo solicitado.

7.4.4.7 *ajax.js*

Archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Youtube sin tener que recargar la página.

7.4.4.8 *functions.php*

Archivo donde se encuentra la función para obtener la id del usuario a partir de una URL y otra función para pasar los segundos a minutos y segundos.

7.4.4.9 *youtube.css*

Archivo de estilo para el módulo de Youtube.

7.4.5 Spotify

7.4.5.1 *index.php*

Archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.

7.4.5.2 *search.php*

Archivo encargado de realizar la búsqueda de videos y de mostrar los resultados obtenidos en Youtube.

7.4.5.3 *play.php*

Archivo encargado de realizar la reproducción del vídeo solicitado.

7.4.5.4 *ajax.js*

Archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Spotify sin tener que recargar la página.

7.4.5.5 *spotify.css*

Archivo de estilo para el módulo de Spotify.

Capítulo 8. Desarrollo de las Pruebas

8.1 Pruebas Unitarias

Caso de Uso 1: Obtener permiso Facebook	
Prueba	Resultado Esperado
Hacer click en iniciar sesión en Facebook y obtener la autorización.	Queda guardada la autorización en la base de datos del sistema.
Resultado Obtenido	
	Token de autorización guardado con éxito.
Prueba	Resultado Esperado
Ocurre algo en la autorización y da error.	Se muestra al usuario la pantalla de iniciar sesión en Facebook.
Resultado Obtenido	
	Pantalla de iniciar sesión mostrada

Caso de Uso 2: Ver inicio Facebook	
Prueba	Resultado Esperado
Hacer click en ver inicio en Facebook.	Ver el inicio del usuario en Facebook.
Resultado Obtenido	
	Pantalla de inicio de Facebook mostrada

Caso de Uso 3: Ver perfil Facebook	
Prueba	Resultado Esperado
Hacer click en ver perfil en Facebook.	Ver el perfil del usuario en Facebook.
Resultado Obtenido	
	Pantalla de perfil de Facebook mostrada

Caso de Uso 4: Ver publicación

Prueba	Resultado Esperado
Hacer click en una publicación.	Se despliega un apartado con los comentarios de la publicación y para poder hacer comentarios en la publicación.
	Resultado Obtenido
	Se despliega la publicación.

Caso de Uso 5: Publicar	
Prueba	Resultado Esperado
Escribir algo para publicar y hacer click en "Publicar"	Aparece un "cargando" y se muestran todas las publicaciones, incluida la que se acaba de realizar.
	Resultado Obtenido
	Aparecen todas las publicaciones, incluida la nueva.
Prueba	Resultado Esperado
No escribir nada para publicar y hacer click en "Publicar"	La aplicación no hace nada.
	Resultado Obtenido
	La aplicación no realiza ningún cambio.

Caso de Uso 6: Hacer me gusta	
Prueba	Resultado Esperado
Hacer click en me gusta en una publicación.	El botón de "me gusta" cambia a "ya no me gusta".
	Resultado Obtenido
	El botón cambia de estado

Caso de Uso 7: Comentar publicación	
Prueba	Resultado Esperado
Escribir algo en una publicación.	La publicación se recarga y aparece el comentario.
	Resultado Obtenido
	Comentario creado con éxito

<u>Caso de Uso 8: Borrar publicación</u>	
Prueba	Resultado Esperado
Hacer click en la "x" que aparece en la publicación	La publicación queda eliminada.
Resultado Obtenido	
	Publicación eliminada con éxito

<u>Caso de Uso 9: Cerrar sesión</u>	
Prueba	Resultado Esperado
Hacer click en cerrar sesión en Facebook.	La autorización queda eliminada de la base de datos de la aplicación y al usuario se le muestra la pantalla de iniciar sesión en Facebook.
Resultado Obtenido	
	Token de autorización borrado de la base de datos.

<u>Caso de Uso 10: Obtener permiso Twitter</u>	
Prueba	Resultado Esperado
Hacer click en iniciar sesión en Twitter y obtener la autorización.	Queda guardada la autorización en la base de datos del sistema.
Resultado Obtenido	
	Token de autorización guardado en la base de datos.
Prueba	Resultado Esperado
Ocurre algo en la autorización y da error.	Se muestra al usuario la pantalla de iniciar sesión en Twitter.
Resultado Obtenido	
	Se muestra la pantalla de inicio en Twitter

<u>Caso de Uso 11: Ver inicio Twitter</u>	
Prueba	Resultado Esperado
Hacer click en ver inicio en Twitter.	Ver el inicio del usuario en Twitter.
Resultado Obtenido	
	Se muestra la pantalla de inicio de Twitter

--	--

Caso de Uso 12: Ver perfil Twitter	
Prueba	Resultado Esperado
Hacer click en ver perfil en Twitter.	Ver el perfil del usuario en Twitter.
	Resultado Obtenido
	Se muestra la pantalla de perfil de Twitter

Caso de Uso 13: Ver menciones	
Prueba	Resultado Esperado
Hacer click en menciones en Twitter.	Ver las menciones del usuario en Twitter.
	Resultado Obtenido
	Se muestra la pantalla de las menciones del usuario

Caso de Uso 14: Twitear	
Prueba	Resultado Esperado
Escribir algo para compartir y hacer click en "Tweet".	Aparece un "cargando" y se muestran todos los tweets incluida la que se acaba de realizar.
	Resultado Obtenido
	Aparecen todas los tweets, incluido el nuevo.
Prueba	Resultado Esperado
No escribir nada para publicar y hacer click en "Tweet".	La aplicación no hace nada.
	Resultado Obtenido
	La aplicación no realiza ningún cambio.
Prueba	Resultado Esperado
Escribir un tweet de más de 140 caracteres.	La aplicación no permite escribir más de 140 caracteres.
	Resultado Obtenido
	La aplicación no permite escribir más caracteres.

Caso de Uso 15: Retwitear	
Prueba	Resultado Esperado
Hacer click en el icono de retwitear.	El mensaje queda retwiteado y se le informa al usuario con un mensaje.
	Resultado Obtenido
	Tweet retwiteado con éxito.

Caso de Uso 16: Borrar tweet	
Prueba	Resultado Esperado
Hacer click en la "x" que aparece en el tweet	El tweet queda eliminado.
	Resultado Obtenido
	Tweet eliminado con éxito.

Caso de Uso 17: Cerrar sesión	
Prueba	Resultado Esperado
Hacer click en cerrar sesión en Twitter.	La autorización queda eliminada de la base de datos de la aplicación y al usuario se le muestra la pantalla de iniciar sesión en Twitter.
	Resultado Obtenido
	Token de autorización borrado de la base de datos.

Caso de Uso 18: Ver recomendados	
Prueba	Resultado Esperado
Hacer click en ver recomendados en Youtube.	Se muestran los vídeos recomendados por Youtube.
	Resultado Obtenido
	Pantalla con los videos recomendados.

Caso de Uso 19: Ver más vistos	
Prueba	Resultado Esperado
Hacer click en ver más vistos en Youtube.	Se muestran los vídeos más vistos en España en Youtube.
	Resultado Obtenido
	Pantalla con los videos más vistos en España.

Caso de Uso 20: Ver trending	
Prueba	Resultado Esperado
Hacer click en ver trending en Youtube.	Se muestran los vídeos trending en Youtube.
	Resultado Obtenido
	Pantalla con los videos trending.

Caso de Uso 21: Buscar video	
Prueba	Resultado Esperado
Realizar una búsqueda de un video en Youtube.	Aparece una lista de videos.
	Resultado Obtenido
	Se muestran los resultados de la búsqueda.
Prueba	Resultado Esperado
No introducir nada en el campo de búsqueda.	La aplicación no hace nada
	Resultado Obtenido
	La aplicación no realiza ningún cambio.

Caso de Uso 22: Reproducir video	
Prueba	Resultado Esperado
Hacer click en un video para reproducirlo.	El video se reproduce.
	Resultado Obtenido
	El video se reproduce.

Caso de Uso 23: Buscar canción	
Prueba	Resultado Esperado

Realizo una búsqueda de una canción en Spotify.	Aparece una lista de canciones.
	Resultado Obtenido
	Se muestra el resultado de la búsqueda.
Prueba	Resultado Esperado
No introducir nada en el campo de búsqueda.	La aplicación no hace nada.
	Resultado Obtenido
	La aplicación no realiza ningún cambio.

<u>Caso de Uso 24: Ver portada</u>	
Prueba	Resultado Esperado
Hago click en una de las canciones.	Se despliega un apartado con la portada del álbum de la canción.
	Resultado Obtenido
	Muestra la portada de la canción.

<u>Caso de Uso 25: Reproducir canción</u>	
Prueba	Resultado Esperado
Hago click en el play de una canción para reproducirla con el Spotify cerrado.	Se abre automáticamente el Spotify y te reproduce la canción.
	Resultado Obtenido
	Se reproduce la canción.
Prueba	Resultado Esperado
Hago click en el play de una canción para reproducirla con el Spotify abierto.	Se reproduce la canción.
	Resultado Obtenido
	Se reproduce la canción.

Caso de Uso 26: Compartir en Facebook	
Prueba	Resultado Esperado
Hacer click en compartir en Facebook.	La canción queda compartida en Facebook.
Resultado Obtenido	
	Se comparte la canción en Facebook.

Caso de Uso 27: Compartir en Twitter	
Prueba	Resultado Esperado
Hacer click en compartir en Twitter.	La canción queda compartida en Twitter.
Resultado Obtenido	
	Se comparte la canción en Twitter.

8.2 Pruebas de Usabilidad y Accesibilidad

8.2.1 Pruebas de Usabilidad

El cuestionario se ha realizado a un total de 10 personas. 5 de ellas no tenían experiencia con redes sociales y las otras 5 sí.

8.2.1.1 Resultados de las pruebas de los usuarios con experiencia en redes sociales

¿Usa un ordenador frecuentemente?
<ol style="list-style-type: none">5. Todos los días 5 personas6. Varias veces a la semana7. Ocasionalmente8. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
<ol style="list-style-type: none">5. Es parte de mi trabajo o profesión 3 personas6. Lo uso básicamente para ocio 3 personas7. Solo empleo aplicaciones estilo Office8. Únicamente leo el correo y navego ocasionalmente
¿Ha usado alguna vez aplicaciones web de redes sociales ?
<ol style="list-style-type: none">4. Sí, he empleado aplicaciones similares 5 personas5. No, aunque si empleo otros programas que me ayudan a realizar tareas similares6. No, nunca
¿Qué busca Vd. Principalmente en un programa?
<ol style="list-style-type: none">6. Que sea fácil de usar 2 personas7. Que sea intuitivo8. Que sea rápido 2 personas9. Que tenga todas las funciones necesarias 1 persona10. Que tenga buen diseño

Se realizan las siguientes pruebas guiadas con los usuarios:

- Loguearse en el módulo de Facebook.
- Ver perfil del usuario de Facebook.
- Compartir una publicación.
- Hacer me gusta en una publicación.
- Eliminar una publicación.
- Loguearse en el módulo de Twitter.

- Twitear.
- Ver las menciones del usuario.
- Borrar un tweet.
- Retwitear.
- Buscar un video en Youtube.
- Reproducir el video en Youtube.
- Buscar una canción en Spotify.
- Reproducir una canción en Spotify.
- Compartir la canción en Facebook.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe donde está dentro de la aplicación?	5 personas			
¿Existe ayuda para las funciones en caso de que tenga dudas?			5 personas	
¿Le resulta sencillo el uso de la aplicación?	5 personas			
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	4 personas	1 persona		
¿El tiempo de respuesta de la aplicación es muy grande?			5 personas	
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es		5 personas		
Los iconos e imágenes usados son		5 personas		
Los colores empleados son		5 personas		
Diseño de la Interfaz	Si		No	A veces
¿Le resulta fácil de usar?	5 personas			
¿El diseño de las pantallas es claro y atractivo?	5 personas			
¿Cree que el programa está bien estructurado?	5 personas			
Observaciones				
En general la aplicación les resulta bastante fácil de usar e intuitiva. Además todos coinciden en que el diseño es atractivo.				

8.2.1.2 Resultados de las pruebas de los usuarios sin experiencia en redes sociales

¿Usa un ordenador frecuentemente?
9. Todos los días 10. Varias veces a la semana 11. Ocasionalmente 5 personas

12. Nunca o casi nunca
¿Qué tipo de actividades realiza con el ordenador?
9. Es parte de mi trabajo o profesión 10. Lo uso básicamente para ocio 11. Solo empleo aplicaciones estilo Office 1 persona 12. Únicamente leo el correo y navego ocasionalmente 4 personas
¿Ha usado alguna vez aplicaciones web de redes sociales ?
7. Sí, he empleado aplicaciones similares 8. No, aunque si empleo otros programas que me ayudan a realizar tareas similares 9. No, nunca 5 personas
¿Qué busca Vd. Principalmente en un programa?
11. Que sea fácil de usar 5 personas 12. Que sea intuitivo 13. Que sea rápido 14. Que tenga todas las funciones necesarias 15. Que tenga buen diseño

Se realizan las siguientes pruebas guiadas con los usuarios:

- Loguearse en el módulo de Facebook.
- Ver perfil del usuario de Facebook.
- Compartir una publicación.
- Hacer me gusta en una publicación.
- Eliminar una publicación.
- Loguearse en el módulo de Twitter.
- Twitear.
- Ver las menciones del usuario.
- Borrar un tweet.
- Retwitear.
- Buscar un video en Youtube.
- Reproducir el video en Youtube.
- Buscar una canción en Spotify.
- Reproducir una canción en Spotify.
- Compartir la canción en Facebook.

Facilidad de Uso	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Sabe donde está dentro de la aplicación?	5 personas			
¿Existe ayuda para las funciones en caso de que tenga dudas?			5 personas	

¿Le resulta sencillo el uso de la aplicación?	3 personas	2 personas		
Funcionalidad	Siempre	Frecuentemente	Ocasionalmente	Nunca
¿Funciona cada tarea como Vd. Espera?	3 personas	2 personas		
¿El tiempo de respuesta de la aplicación es muy grande?			5 personas	
Calidad del Interfaz				
Aspectos gráficos	Muy Adecuado	Adecuado	Poco Adecuado	Nada Adecuado
El tipo y tamaño de letra es		4 personas	1 persona	
Los iconos e imágenes usados son		5 personas		
Los colores empleados son		5 personas		
Diseño de la Interfaz		Si	No	A veces
¿Le resulta fácil de usar?		5 personas		
¿El diseño de las pantallas es claro y atractivo?		5 personas		
¿Cree que el programa está bien estructurado?		5 personas		
Observaciones				
<p>En general, parece que no resulta especialmente complicado utilizar la aplicación a los usuarios sin experiencia en redes sociales. Normalmente, tienen más problemas de entender los tecnicismos de cada aplicación (Ej. Tweet, “me gusta”, hasta, etc.) que de la propia utilización de Flypper. Además todos coinciden en que el diseño es atractivo.</p>				

8.2.1.3 Resultados del cuestionario para el responsable de las pruebas

Aspecto Observado	Notas
El usuario comienza a trabajar de forma rápida por las tareas	Al principio tardan unos minutos en observar la aplicación, pero rápidamente comienzan a trabajar por la aplicación.
Tiempo en realizar cada tarea	Las realizaciones de las tareas son rápidas.
Errores leves cometidos	No han cometido ningún error leve.
Errores graves cometidos	No han cometido ningún error grave.
Encuentran rápidamente lo que se les pide	En general, encuentran rápidamente lo que se les pide.
El usuario está satisfecho con la aplicación.	Los usuarios están satisfechos con la aplicación y coinciden en que es una buena herramienta para el día a día.

8.2.1.4 Resultados de la guía de usabilidad desarrollada por Yusef Hassan Montero

Criterios	¿Cumplido?
<u>Generales</u>	
¿Cuáles son los objetivos del sitio web? ¿Son concretos y bien definidos? ¿Los contenidos y servicios que ofrece se corresponden con esos objetivos?	SI
¿Tiene una URL correcta, clara y fácil de recordar? ¿Y las URL de sus páginas internas? ¿Son claras y permanentes?	SI
¿Muestra de forma precisa y completa qué contenidos o servicios ofrece realmente el sitio web? El diseño de la página de inicio debe ser diferente al resto de páginas y cumplir la función de 'escaparate' del sitio.	SI
¿La estructura general del sitio web está orientada al usuario? Los sitios web deben estructurarse pensando en el usuario, sus objetivos y necesidades. La estructura interna de la empresa u organización, cómo funciona o se organiza no interesan al usuario.	SI
¿El look & feel general se corresponde con los objetivos, características, contenidos y servicios del sitio web? Ciertas combinaciones de colores ofrecen imágenes más o menos formales, serias o profesionales.	SI
¿Es coherente el diseño general del sitio web? Se debe mantener una coherencia y uniformidad en las estructuras y colores de todas las páginas. Esto sirve para que el usuario no se desoriente en su navegación.	SI
¿Es reconocible el diseño general del sitio web? Cuánto más se parezca el sitio web al resto de sitios web, más fácil será de usar.	SI
¿El sitio web se actualiza periódicamente? ¿Indica cuándo se actualiza? Las fechas que se muestren en la página deben corresponderse con actualizaciones, noticias, eventos...no con la fecha del sistema del usuario.	SI
<u>Identidad e Información</u>	
¿Se muestra claramente la identidad de la empresa-sitio a través de todas las páginas?	SI
El Logotipo, ¿es significativo, identificable y suficientemente visible?	SI
El eslogan o <i>tagline</i> , ¿expresa realmente qué es la empresa y qué servicios ofrece?	-
¿Se ofrece algún enlace con información sobre la empresa, sitio web, 'webmaster',...?	-
¿Se proporciona mecanismos para ponerse en contacto con la empresa? (email, teléfono, dirección postal, fax...)	-
¿Se proporciona información sobre la protección de datos de carácter personal de los clientes o los derechos de autor de los contenidos del sitio web?	SI
En artículos, noticias, informes... ¿Se muestra claramente información sobre el autor, fuentes y fechas de creación y revisión del documento?	-
<u>Lenguaje y Redacción</u>	

¿El sitio web habla el mismo lenguaje que sus usuarios? Se debe evitar usar un lenguaje corporativista. Así mismo, hay que prestarle especial atención al idioma, y ofrecer versiones del sitio en diferentes idiomas cuando sea necesario.	SI
¿Emplea un lenguaje claro y conciso?	SI
¿Es amigable, familiar y cercano? Es decir, lo contrario a utilizar un lenguaje constantemente imperativo, mensajes crípticos, o tratar con "desprecio" al usuario.	SI
¿1 párrafo = 1 idea? Cada párrafo es un objeto informativo. Trasmite ideas, mensajes...Se deben evitar párrafos vacíos o varios mensajes en un mismo párrafo.	SI
<u>Rotulado</u>	
Los rótulos, ¿son significativos? Ejemplo: evitar rótulos del tipo "haga clic aquí".	SI
¿Usa rótulos estándar? Siempre que exista un "estándar" comúnmente aceptado para el caso concreto, como "Mapa del Sitio" o "Acerca de...".	SI
¿Usa un único sistema de organización, bien definido y claro? No se deben mezclar diferentes. Los sistemas de organización son: alfabético, geográfico, cronológico, temático, orientado a tareas, orientado al público y orientado a metáforas.	SI
¿Utiliza un sistema de rotulado controlado y preciso? Por ejemplo, si un enlace tiene el rótulo "Quiénes somos", no puede dirigir a una página cuyo encabezamiento sea "Acerca de"	-
El título de las páginas, ¿Es correcto? ¿Ha sido planificado? Relacionado con la capacidad para poder buscar y encontrar el sitio web.	SI
<u>Estructura y Navegación</u>	
La estructura de organización y navegación, ¿Es la más adecuada? Hay varios tipos de estructuras: jerárquicas, hipertextual, facetada,...	SI
En el caso de estructura jerárquica, ¿Mantiene un equilibrio entre Profundidad y Anchura?	-
En el caso de ser puramente hipertextual, ¿Están todos los clúster de nodos comunicados? Aquí se mide la distancia entre nodos.	-
¿Los enlaces son fácilmente reconocibles como tales? ¿Su caracterización indica su estado (visitados, activos,...)? Los enlaces no sólo deben reconocerse como tales, sino que su caracterización debe indicar su estado, y ser reconocidos como una unidad	SI
En menús de navegación, ¿Se ha controlado el número de elementos y de términos por elemento para no producir sobrecarga memorística? No se deben superar los 7 ± 2 elementos, ni los 2 o, como mucho, 3 términos por elemento.	SI
¿Es predecible la respuesta del sistema antes de hacer clic sobre el enlace? Relacionado con el nivel de significación del rótulo del enlace, aunque también con: el uso de globos de texto, información contextual, la barra de estado del navegador,...	SI

¿Se ha controlado que no haya enlaces que no lleven a ningún sitio? Enlaces que no lleven a ningún sitio: Los enlaces rotos, y los que enlazan con la misma página que se está visualizando (por ejemplo enlaces a la "home" desde la misma página de inicio)	SI
¿Existen elementos de navegación que orienten al usuario acerca de dónde está y cómo deshacer su navegación? ...como <i>breadcrumbs</i> , enlaces a la página de inicio,...recuerde que el logo también es recomendable que enlace con la página de inicio.	SI
Las imágenes enlace, ¿se reconocen como clicables? ¿Incluyen un atributo 'title' describiendo la página de destino? En este sentido, también hay que cuidar que no haya imágenes que parezcan enlaces y en realidad no lo sean.	SI
¿Se ha evitado la redundancia de enlaces?	SI
¿Se ha controlado que no haya páginas "huérfanas"? Páginas huérfanas: que aún siendo enlazadas desde otras páginas, éstas no enlacen con ninguna.	SI
<u>Layout de la Página</u>	
¿Se aprovechan las zonas de alta jerarquía informativa de la página para contenidos de mayor relevancia? (como por ejemplo la zona central)	SI
¿Se ha evitado la sobrecarga informativa? Esto se consigue haciendo un uso correcto de colores, efectos tipográficos y agrupaciones para discriminar información. Los grupos diferentes de objetos informativos de una página deben ser 7 ± 2 .	SI
¿Es una interfaz limpia, sin ruido visual?	SI
¿Existen zonas en "blanco" entre los objetos informativos de la página para poder descansar la vista?	SI
¿Se hace un uso correcto del espacio visual de la página? Es decir, que no se desaproveche demasiado espacio con elementos de decoración, o grandes zonas en "blanco", y que no se adjudique demasiado espacio a elementos de menor importancia.	SI
¿Se utiliza correctamente la jerarquía visual para expresar las relaciones del tipo "parte de" entre los elementos de la página? (La jerarquía visual se utiliza para orientar al usuario)	SI
¿Se ha controlado la longitud de página? Se debe evitar en la medida de lo posible el <i>scrolling</i> . Si la página es muy extensa, se debe fraccionar.	SI
<u>Búsqueda (si es necesario, por la extensión del sitio, incorporar un buscador interno)</u>	
¿Se encuentra fácilmente accesible? Es decir: directamente desde la home, y a ser posible desde todas las páginas del sitio, y colocado en la zona superior de la página.	SI
¿Es fácilmente reconocible como tal?	SI
¿Permite la búsqueda avanzada? (siempre y cuando, por las características del sitio web, fuera de utilidad que la ofreciera)	-
¿Muestra los resultados de la búsqueda de forma comprensible para el usuario?	-
¿La caja de texto es lo suficientemente ancha?	-
¿Asiste al usuario en caso de no poder ofrecer resultados para una consultada dada?	-
<u>Elementos Multimedia</u>	

¿Las fotografías están bien recortadas? ¿Son comprensibles? ¿Se ha cuidado su resolución?	SI
¿Las metáforas visuales son reconocibles y comprensibles por cualquier usuario? (prestar especial atención a usuarios de otros países y culturas)	SI
¿El uso de imágenes o animaciones proporciona algún tipo de valor añadido?	SI
¿Se ha evitado el uso de animaciones cíclicas?	SI
<u>Ayuda</u>	
Si posee una sección de Ayuda, ¿Es verdaderamente necesaria? Siempre que se pueda prescindir de ella simplificando los elementos de navegación e interacción, debe omitirse esta sección.	-
En enlace a la sección de Ayuda, ¿Está colocado en una zona visible y "estándar"? La zona de la página más normal para incluir el enlace a la sección de Ayuda, es la superior derecha.	-
¿Se ofrece ayuda contextual en tareas complejas? (transferencias bancarias, formularios de registro...)	-
Si posee FAQs, ¿Es correcta tanto la elección como la redacción de las preguntas? ¿Y las respuestas?	-
<u>Accesibilidad (debería cubrirse con los test de Accesibilidad posteriores)</u>	
¿El tamaño de fuente se ha definido de forma relativa, o por lo menos, la fuente es lo suficientemente grande como para no dificultar la legibilidad del texto?	SI
¿El tipo de fuente, efectos tipográficos, ancho de línea y alineación empleadas facilitan la lectura?	SI
¿Existe un alto contraste entre el color de fuente y el fondo?	SI
¿Incluyen las imágenes atributos 'alt' que describan su contenido?	
¿Es compatible el sitio web con los diferentes navegadores? ¿Se visualiza correctamente con diferentes resoluciones de pantalla? Se debe prestar atención a: JScript, CSS, tablas, fuentes...	SI
¿Puede el usuario disfrutar de todos los contenidos del sitio web sin necesidad de tener que descargar e instalar <i>plugins</i> adicionales?	SI
¿Se ha controlado el peso de la página? Se deben optimizar las imágenes, controlar el tamaño del código JScript...	SI
¿Se puede imprimir la página sin problemas? Leer en pantalla es molesto, por lo que muchos usuarios preferirán imprimir las páginas para leerlas. Se debe asegurar que se puede imprimir la página (no salen partes cortadas), y que el resultado es legible.	-
<u>Control y Retroalimentación</u>	
¿Tiene el usuario todo el control sobre el interfaz? Se debe evitar el uso de ventanas pop-up, ventanas que se abren a pantalla completa, banners intrusivos...	SI
¿Se informa constantemente al usuario acerca de lo que está pasando? Si el usuario tiene que esperar hasta que se termine una operación, se debe mostrar un mensaje indicándole y que debe esperar, con el tiempo de espera estimado o una barra de progreso.	SI

¿Se informa al usuario de lo que ha pasado? Por ejemplo, cuando un usuario valora un artículo o responde a una encuesta, se le debe informar de que su voto ha sido procesado correctamente.	SI
Cuando se produce un error, ¿se informa de forma clara y no alarmista al usuario de lo ocurrido y de cómo solucionar el problema? Siempre es mejor intentar evitar que se produzcan errores a tener que informar al usuario del error.	SI
¿Posee el usuario libertad para actuar? NO restringir la libertad del usuario: Uso de animaciones que no pueden ser "saltadas", páginas en las que desaparecen los botones de navegación, no impida al usuario poder usar el botón derecho de su ratón...	SI
¿Se ha controlado el tiempo de respuesta? Esto tiene que ver con el peso de cada página (accesibilidad) y tiene relación con el tiempo que tarda el servidor en finalizar una tarea y responder. El tiempo máximo que esperará un usuario son 10 segundos	SI
<u>Aclaraciones</u>	
¿Se ha evaluado adecuadamente la orientación del usuario? (Donde estoy, como volver, que he visitado, que va a pasar)	SI
¿Se ha usado correctamente la publicidad? ¹	-

8.2.2 Pruebas de Accesibilidad

8.2.2.1 Revisión Preliminar

A continuación se enumeran los programas utilizados para realizar las pruebas de accesibilidad:

- Internet Explorer
- Mozilla Firefox
- Google Chrome
- Safari
- Opera
- Web Developer Extension de Firefox
- Jawx (navegador por voz)
- Lynx (navegador de texto)

Paso 1. Selección de un grupo de páginas representativo de la aplicación

Las pruebas de accesibilidad serán probadas en las dos páginas disponibles en la aplicación.

¹ Respecto a la publicidad (normalmente en forma de banners), se puede evaluar desde varios criterios: lenguaje (lenguaje y redacción), nivel de significación de los rótulos (rotulado), jerarquía informativa y ruido visual (*layout* de la página), pop-ups y banners intrusivos (control y retroalimentación)...

Paso 2. Examinar las páginas usando un navegador gráfico

Utilizamos Web Developer Extension de Firefox para realizar las pruebas

Esta actividad comprende las siguientes operaciones:

- Desactivamos las imágenes de la aplicación y comprobamos que sigue siendo usable.

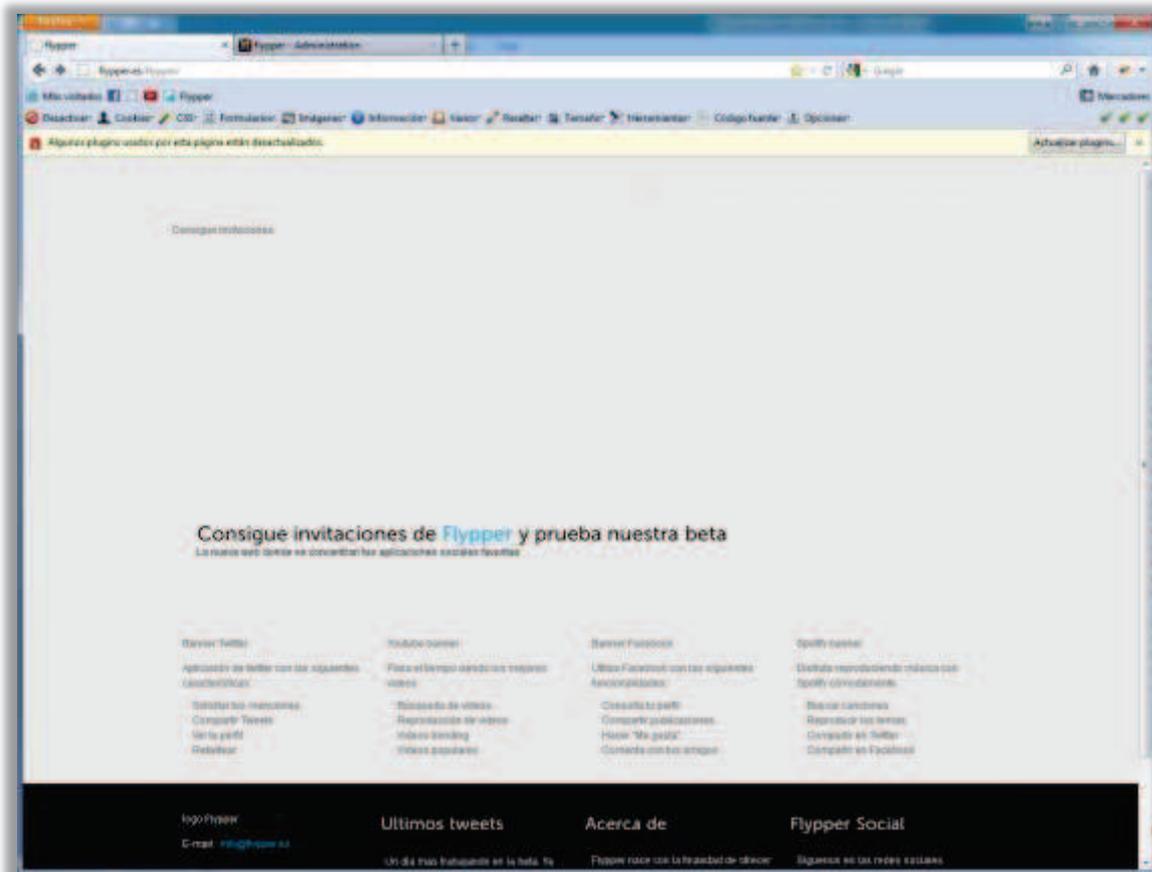


Figura 8.1. Página principal sin imágenes



Figura 8.2. Panel de módulos de Flypper sin imágenes

- Cambiamos el tamaño del texto y comprobamos que sigue siendo usable.



Figura 8.3. Página principal cambiando tamaño del texto

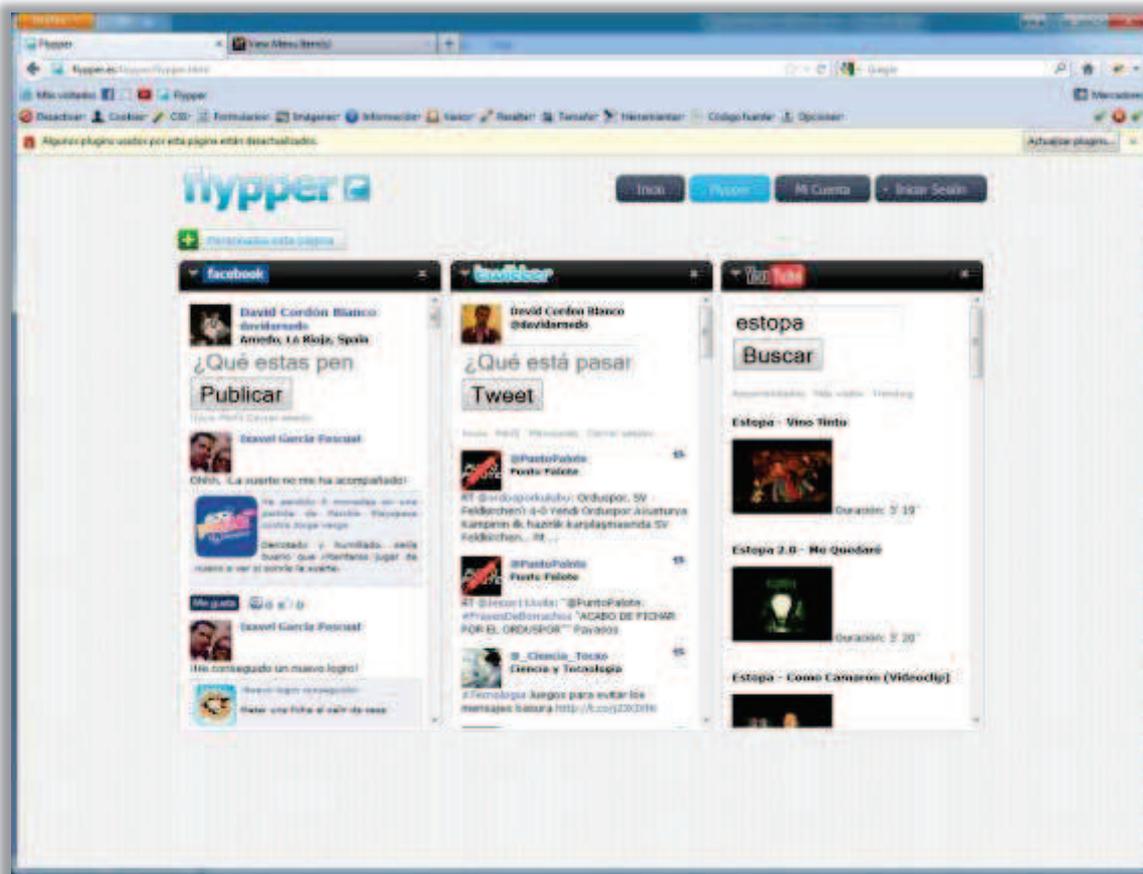


Figura 8.4. Panel de módulos de Flypper cambiando el texto

- Se ha modificado la resolución y el tamaño de la ventana. Las resoluciones que se han probado son 800x600, 1024x768, 1280x768 y 1280x800. La página es usable en todas las resoluciones anteriores.
- Se ha probado la página en monitores de 13", 15", 17" y 24". En todos los casos la página sigue siendo usable.

Paso 3. Utilizar herramientas automáticas de evaluación de accesibilidad

- Pasamos la herramienta TAW y Hera para la evaluación automática del sitio web. Se nos informa de que debemos mirar 8 posibles problemas. Una vez revisados comprobamos que se cumplen todos.

Paso 4. Resumen de resultados

Finalmente, podemos concluir que la página presenta un problema principal que es el de mostrar todo con valores absolutos en vez de relativos. Tanto la letra como el posicionamiento de la web o de las imágenes no se adaptan a las pantallas si no que se mantiene con los valores establecidos.

En una futura versión final habría que corregir estos errores.

8.2.2.2 Evaluación de Conformidad

Paso 1. Determinar el alcance de la evaluación

Definimos como nivel de conformidad A de las WCAG 1.0

Paso 2. Utilizar herramientas de evaluación automática de accesibilidad

Comprobamos las páginas con la extensión de Firefox y es apta. Además comprobamos con los validadores de la W3C.

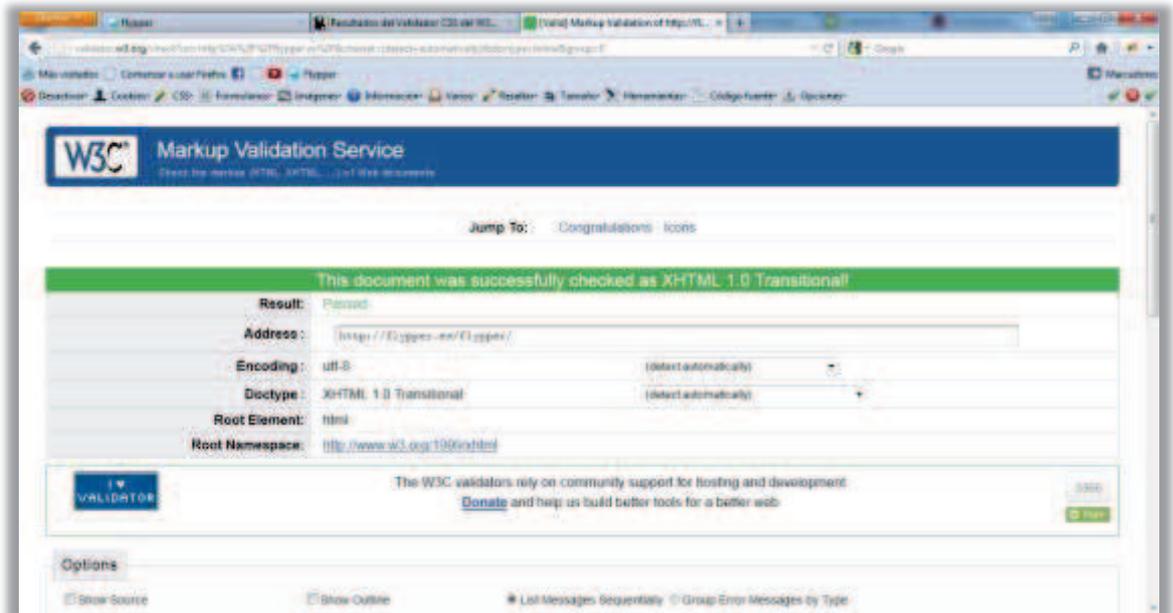


Figura 8.5. XHTML Transitional válido



Figura 8.6. CSS3 válido

Paso 3. Evaluar manualmente la muestra de páginas

Se utilizará el checklist de puntos de control de las WCAG 1.0 que aparecerá en el siguiente punto.

Paso 4. Elaborar el informe de conclusiones

- Se ha alcanzado un nivel A de las WCAG 1.0. No se alcanza nivel AA por utilizar valores absolutos en las medidas de la página. En esta fase beta se acepta nivel A, pero más adelante, cuando la aplicación sea final, se debe cumplir AA.

8.2.2.3 Checklist del WCAG 1.0

Puntos de verificación Prioridad 1:

En general (Prioridad 1)	Sí	No	N/A
1.1 Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el contenido del elemento). <i>Esto incluye:</i> imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, <i>GIFs</i> animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.	X		
2.1 Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.	X		
4.1 Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (por ejemplo, leyendas).			X
6.1 Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.	X		
6.2 Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.	X		
7.1 Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla.	X		
14.1 Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.	X		
Y si utiliza imágenes y mapas de imagen (Prioridad 1)	Sí	No	N/A
1.2 Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor.			X
9.1 Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.			X
Y si utiliza tablas (Prioridad 1)	Sí	No	N/A
5.1 En las tablas de datos, identifique los encabezamientos de fila y columna.			X
5.2 Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos.			X
Y si utiliza marcos ("frames") (Prioridad 1)	Sí	No	N/A

12.1 Titule cada marco para facilitar su identificación y navegación.	X		
Y si utiliza "applets" y "scripts" (Prioridad 1)	Sí	No	N/A
6.3 Asegure que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, <i>applets</i> u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible.			X
Y si utiliza multimedia (Prioridad 1)	Sí	No	N/A
1.3 Hasta que las aplicaciones de usuario puedan leer en voz alta automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la información importante de la banda visual de una presentación multimedia.	X		
1.4 Para toda presentación multimedia dependiente del tiempo (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación.	X		
Y si todo lo demás falla (Prioridad 1)	Sí	No	N/A
11.4 Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.			x

8.3 Pruebas de Rendimiento

Para realizar las pruebas de rendimiento utilizamos las herramientas de desarrolladores de Google Chrome. En la pestaña “Network” podemos observar el tiempo de carga de la web. Normalmente tarda unos 9 segundos de media en cargar.

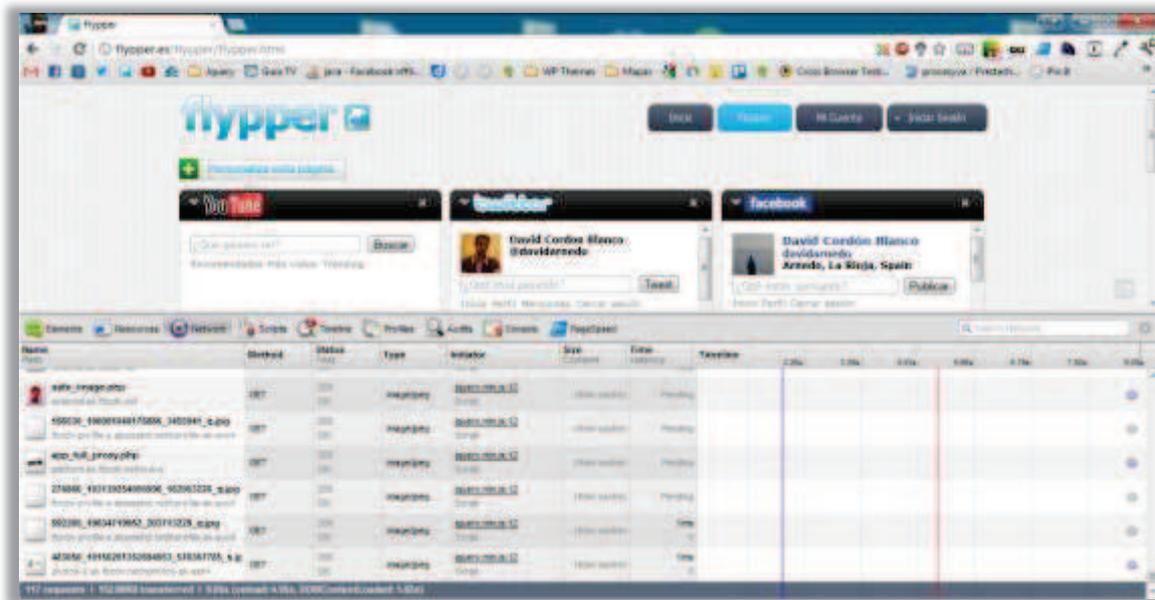


Figura 8.7. Prueba de rendimiento

Puesto que la aplicación trabaja contra varias APIs, se considera que el tiempo de respuesta de la aplicación es adecuado. Además en la aplicación aparecen imágenes de “cargando” para que el usuario espere de la manera más agradable posible y la aplicación no de sensación de no estar realizando ninguna tarea.

Capítulo 9. Manuales del Sistema

9.1 Manual de Instalación y ejecución

Para realizar la instalación de la aplicación realizamos los siguientes pasos:

- Instalamos un servidor web que soporte html5, php y Javascript. Además el servidor debe tener soportar una base de datos mySQL. Wamp Server cumple con estos requisitos. Se puede descargar del siguiente link:

<http://www.wampserver.com/en/>

- Descargamos Joomla de la siguiente dirección:

<http://www.joomla.org/download.html>

- Copiamos la carpeta de instalación de Joomla en el directorio “www” de Wamp Server.
- Realizamos la instalación de Joomla.

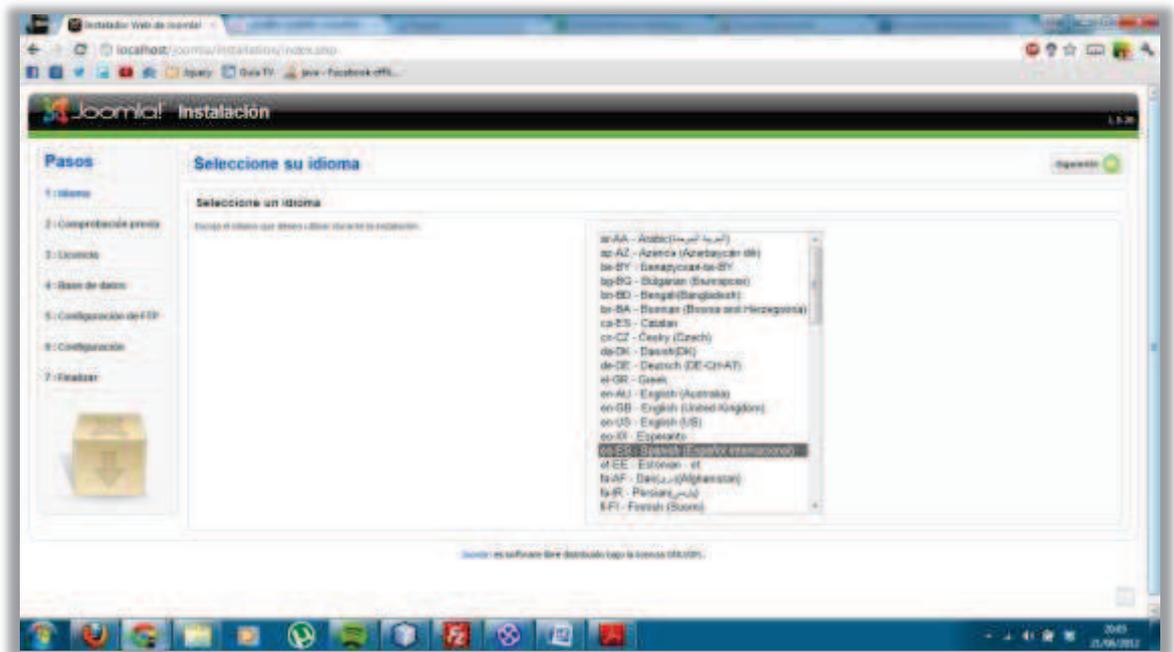


Figura 9.1. Instalación Joomla paso 1



Figura 9.2. Instalación Joomla paso 2

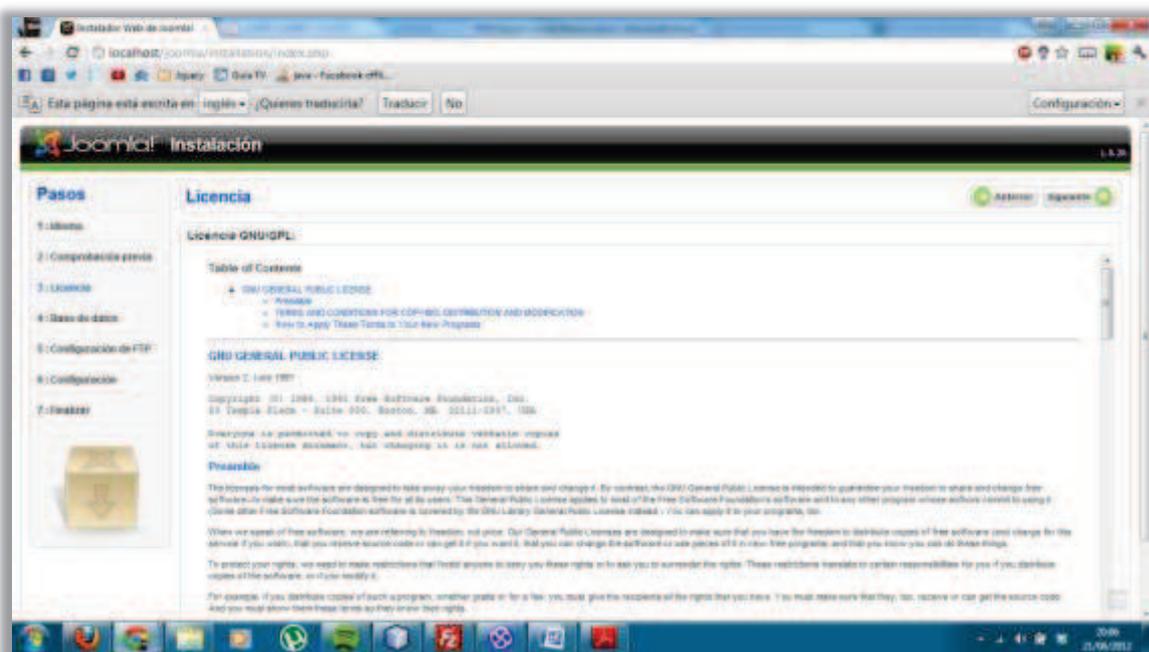


Figura 9.3. Instalación Joomla paso 3



Figura 9.4. Instalación Joomla paso 4



Figura 9.5. Instalación Joomla paso 5



Figura 9.6. Instalación Joomla paso 6

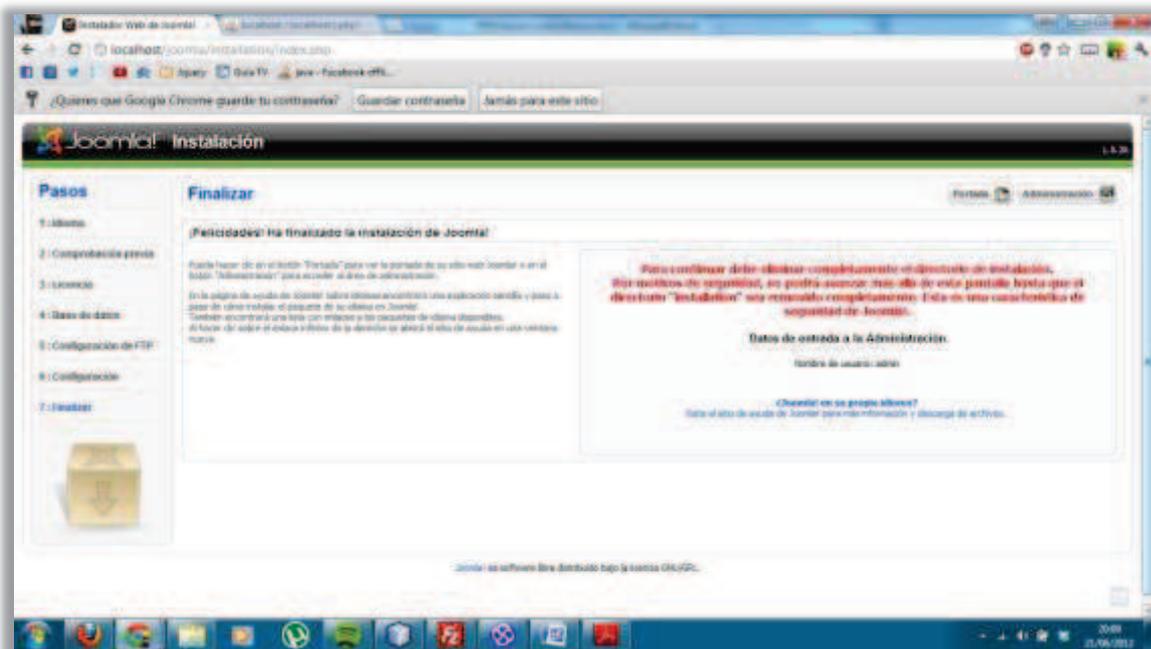


Figura 9.7. Instalación Joomla paso 7

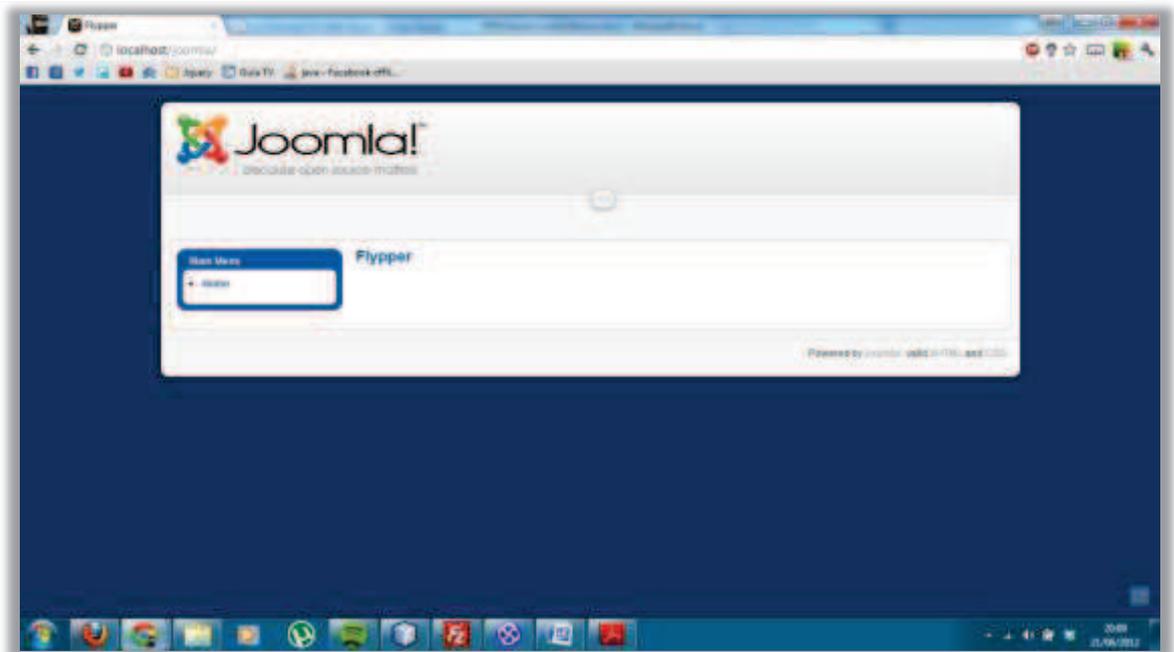


Figura 9.8. Instalación Joomla completada

- Una vez instalado el Joomla, descargamos el componente Drag&Drop del siguiente enlace:
<http://www.joomplace.com/drag-and-drop-2/drag-and-drop-2.html>
- Instalamos el componente, en la pestaña de instalación de módulos y componentes de Joomla.

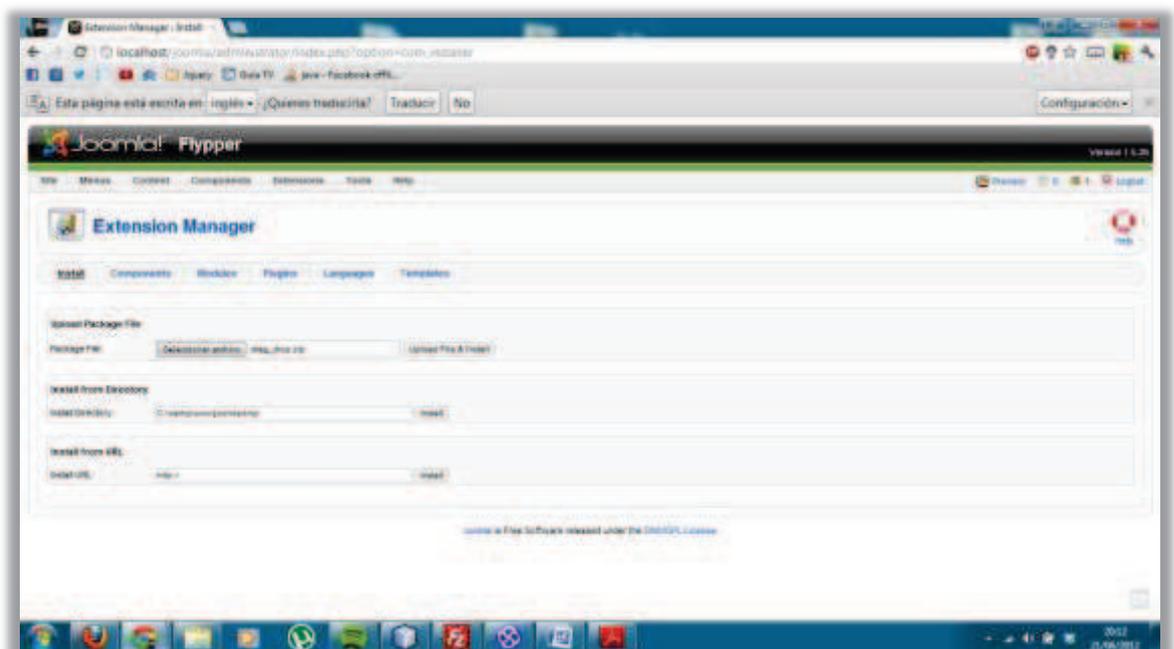


Figura 9.9. Instalación Drag&Drop

- Una vez instalado el componente copiamos los archivos de la aplicación al FTP para poderlos utilizar.

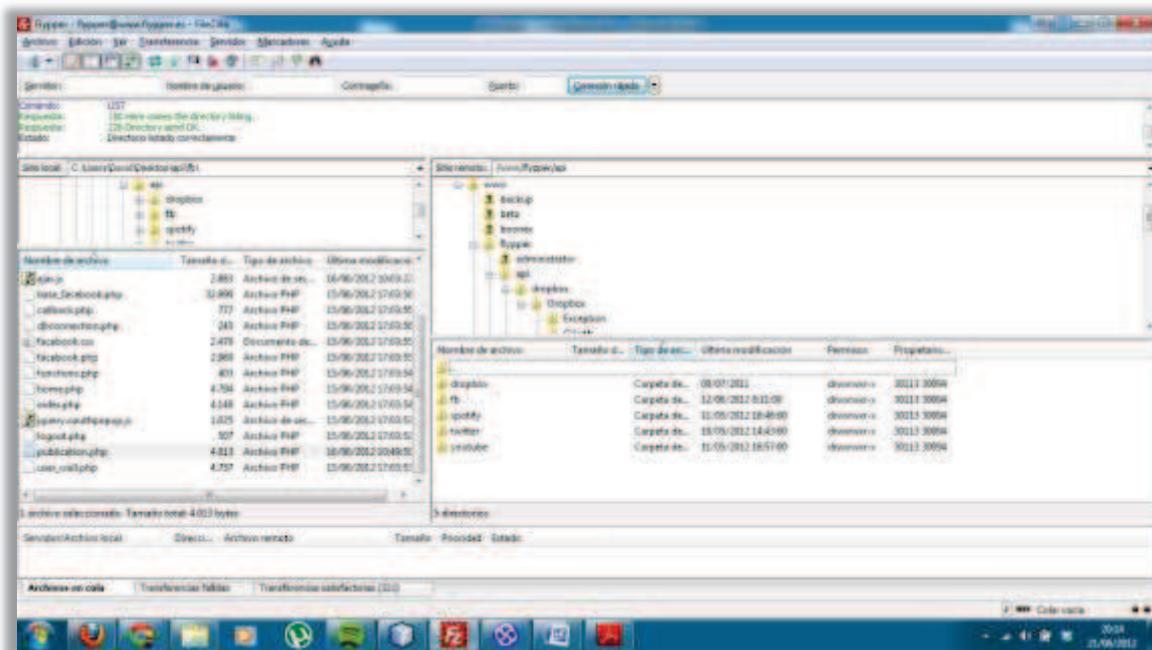


Figura 9.10. Archivos FTP

- Configuramos el componente.



Figura 9.11. Configuración Drag&Drop

- Creamos los módulos de Facebook, Twitter, Youtube y Spotify en la pestaña de la administración de los módulos como Wrapper.

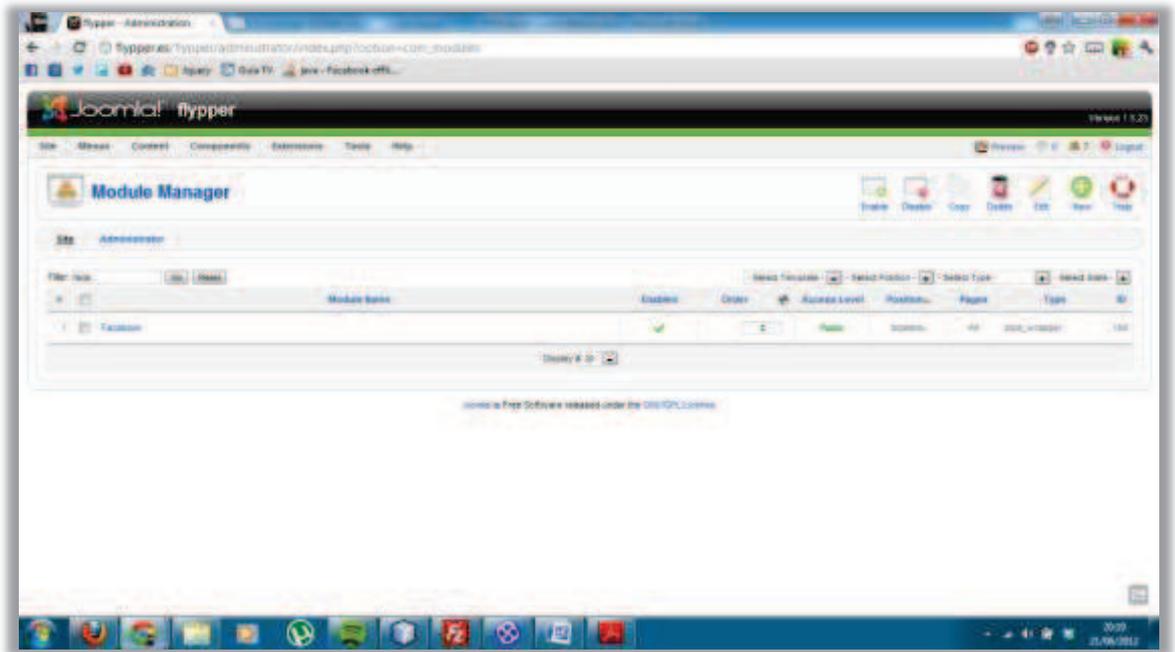


Figura 9.12. Creación módulo Facebook

- Indicamos para cada módulo donde se encuentra dentro de nuestro FTP.

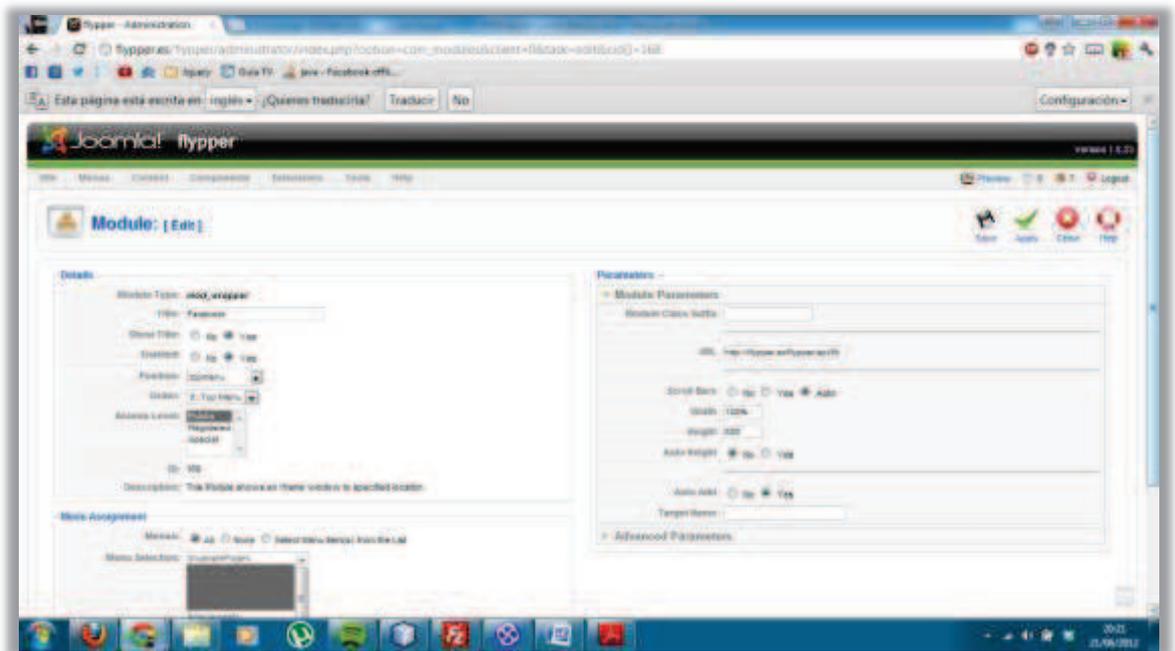


Figura 9.13. Configuración del módulo Facebook

- Añadimos los módulos que acabamos de crear al componente Drag&Drop.

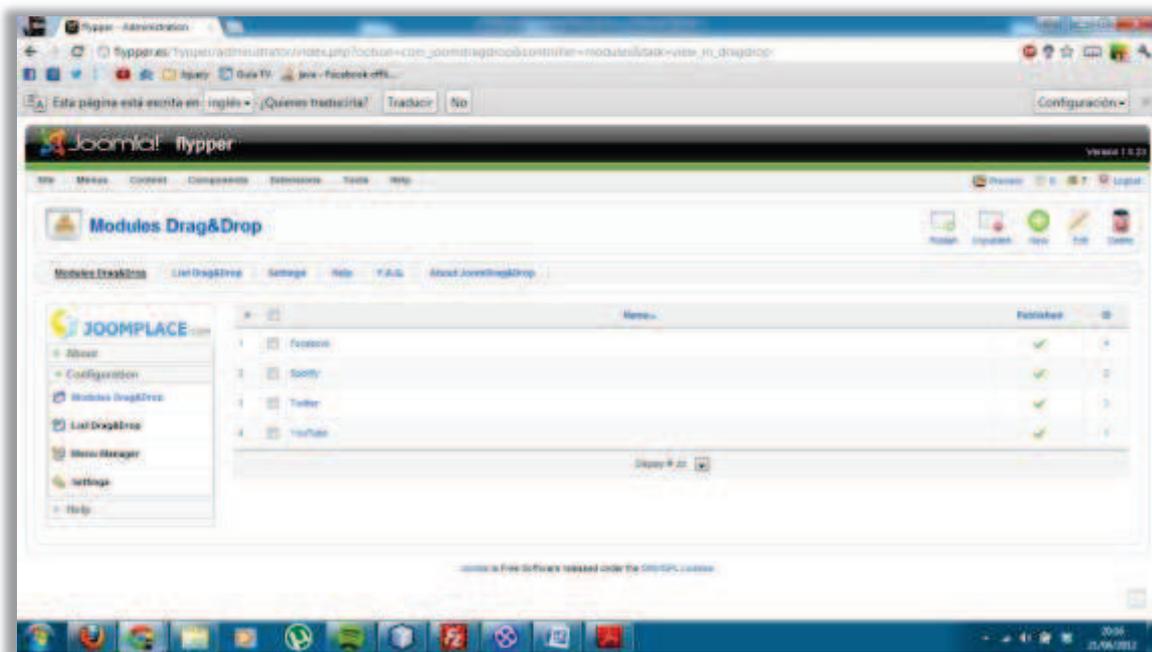


Figura 9.14. Añadir módulos al componente Drag&Drop

- Configuramos cada módulo por separado.

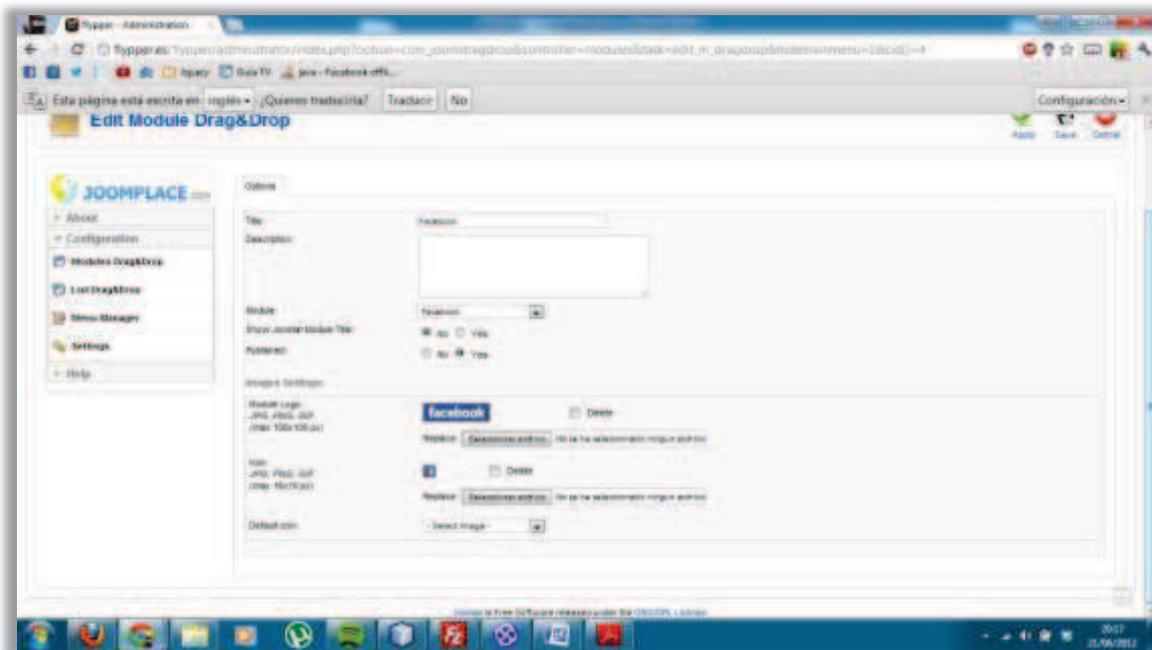


Figura 9.15. Configuración módulo Facebook en Componente Drag&Drop

9.2 Manual de Usuario

9.2.1 Usuario anónimo

Cuando el usuario entra a Flypper.es se encontrará con la siguiente pantalla de inicio.



Figura 9.16. Pantalla inicial de Flypper

En ella encontraremos información general sobre la aplicación y su funcionalidad en su fase beta. Como usuario anónimo no se muestran todas las pestañas disponibles. Para tener todas las funciones disponibles hace falta estar registrado.

Podemos registrarnos en la aplicación poniendo el ratón encima de la pestaña "Iniciar sesión" y luego en "Crear una cuenta". Introduciremos los datos en el formulario de registro. Para poder utilizar la aplicación hay que validar previamente el correo haciendo click en el enlace que la aplicación envía al usuario.

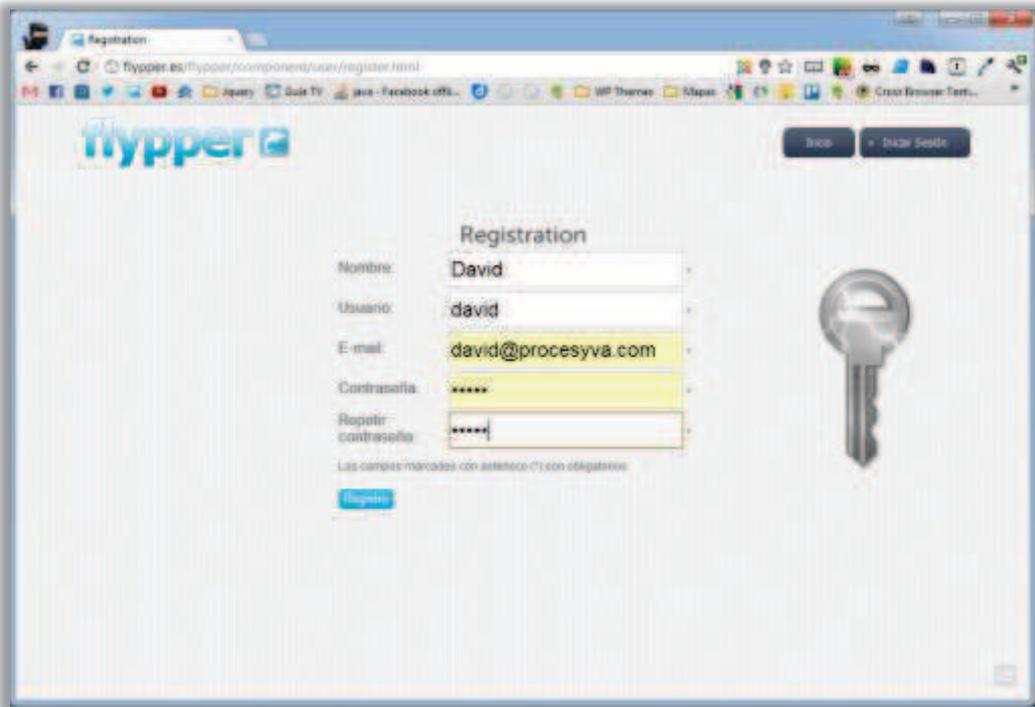


Figura 9.17. Pantalla de registro en Flypper

9.2.2 Usuario registrado

Una vez registrados realizaremos el login. Para ello ponemos el ratón sobre la pestaña “Iniciar sesión”, introducimos nuestro usuario y contraseña y hacemos click en “Login”.



Figura 9.18. Pantalla de login

Cuando hayamos iniciado sesión, se redirigirá al usuario dentro de Flypper y aparecerán 2 nuevas pestañas que son “Mi cuenta”, donde podremos cambiar los datos del usuario, y “Flypper” donde se encuentra toda la funcionalidad de la aplicación.

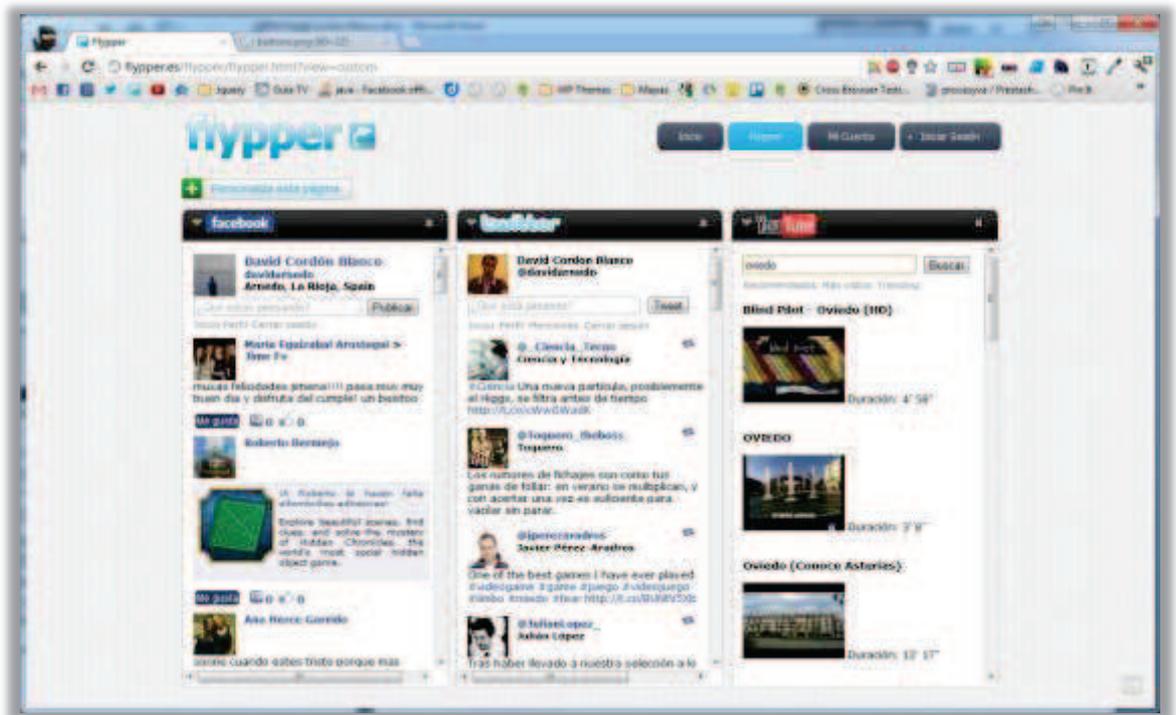


Figura 9.19. Pantalla principal de Flypper

Una vez dentro de Flypper podremos modificar el aspecto de la página arrastrando, minimizando, añadiendo o borrando los módulos existentes.

- Para arrastrar se hace click en la parte superior de cada módulo y se arrastra a la posición deseada.

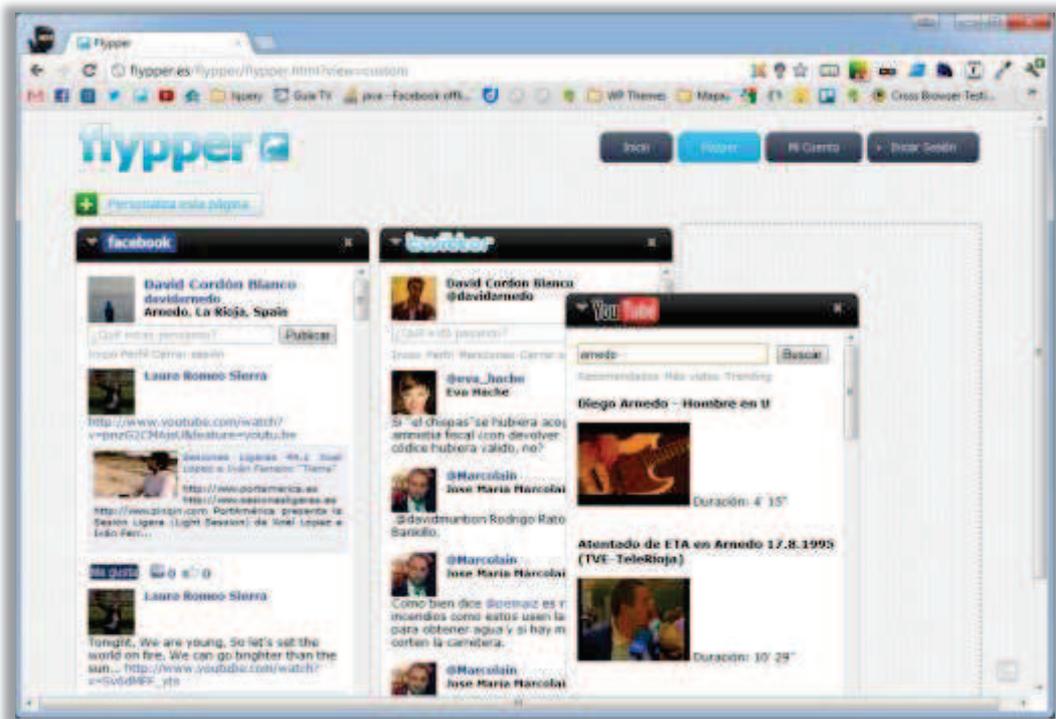


Figura 9.20. Acción de arrastrar módulos

- Para minimizar hay que hacer click sobre el triángulo que apunta hacia abajo en la parte superior izquierda del módulo.



Figura 9.21. Acción de minimizar un módulo

- Para eliminar un módulo que no deseemos utilizar pulsaremos la “x” situada en la parte superior derecha del módulo.



Figura 9.22. Acción de cerrar un módulo

- Para añadir un nuevo módulo hay que hacer click sobre el botón llamado “Personaliza esta página”.

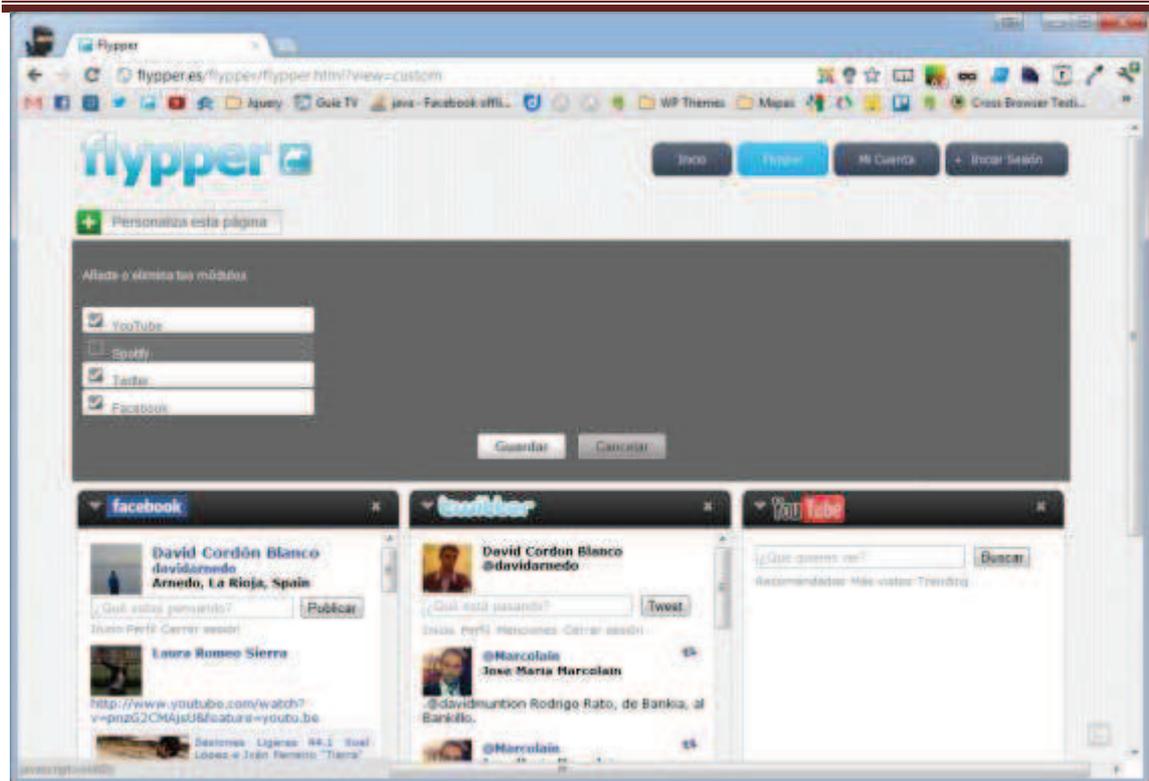


Figura 9.23. Acción de añadir un módulo

A continuación se van a describir la utilización de los diferentes módulos.

9.2.2.1 Módulo Facebook

La primera vez que el usuario entre a Flypper deberá conceder los permisos de Facebook a la aplicación para que pueda utilizar el módulo. Para ello se hace click en el botón de conectar con Facebook.



Figura 9.24. Pantalla iniciar sesión en Facebook

Aparecerá un pop up para introducir el usuario y contraseña de Facebook.

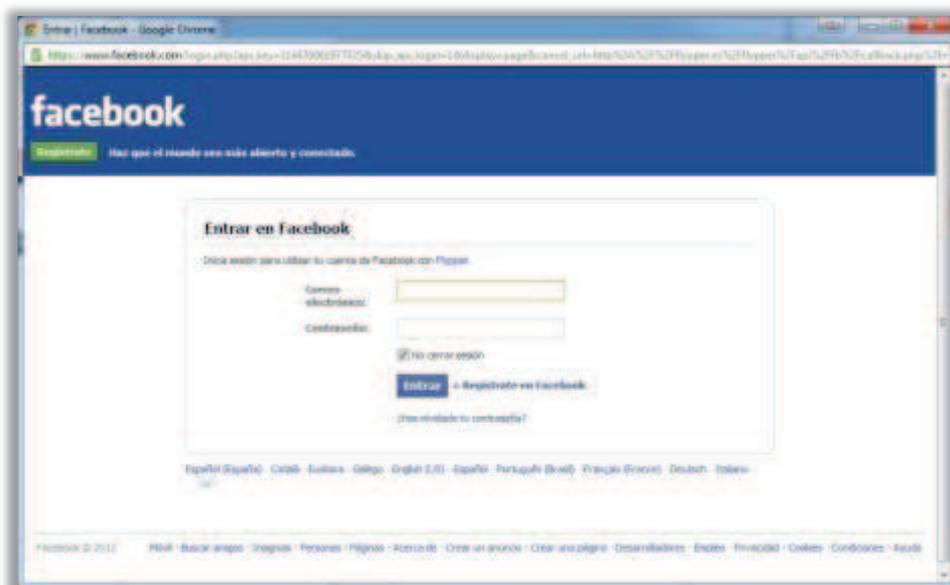


Figura 9.25. Pantalla login Facebook

Una vez que el usuario haga login en Facebook, aparecerá una pantalla para conceder los permisos a Flypper.



Figura 9.26. Ir a la aplicación de Flypper en Facebook



Figura 9.27 Concesión de permisos a Flypper en Facebook

Cuando se hayan concedido los permisos a la aplicación se mostrará la pantalla de inicio de Facebook. En esta pantalla podremos acceder al inicio y el perfil del usuario haciendo click en las pestañas correspondientes. Para compartir una publicación deberemos escribir en el apartado correspondiente y hacer click en “Publicar”.



Figura 9.28. Pantalla de inicio de Facebook



Figura 9.29. Pantalla de perfil de Facebook

Para comentar una publicación deberemos hacer click en la que deseamos comentar. Se desplegará un apartado donde podremos ver los comentarios de esa publicación y compartir tu propio comentario.

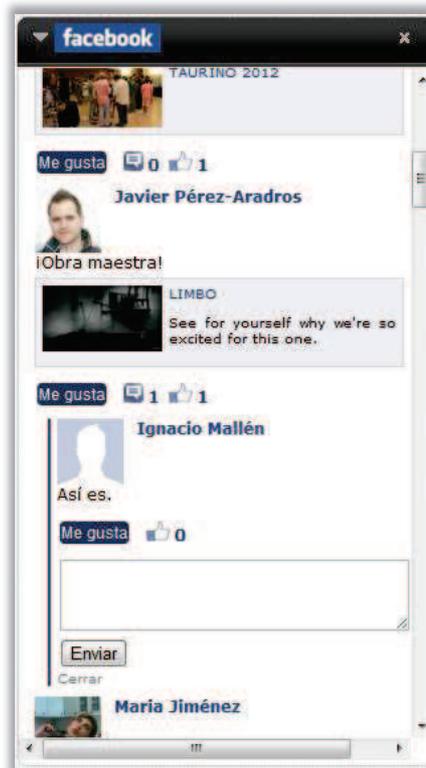


Figura 9.30. Pantalla comentar publicación

Además de todo esto se podrán eliminar las publicaciones del usuario o hacer me gusta haciendo click en el botón correspondiente de la publicación.

9.2.2.2 Módulo Twitter

La primera vez que el usuario entre a Flypper deberá conceder los permisos de Twitter a la aplicación para que pueda utilizar el módulo. Para ello se hace click en el botón de conectar con Facebook.



Figura 9.31. Pantalla iniciar sesión en Twitter

Aparecerá un pop up para introducir el usuario y contraseña de Twitter.

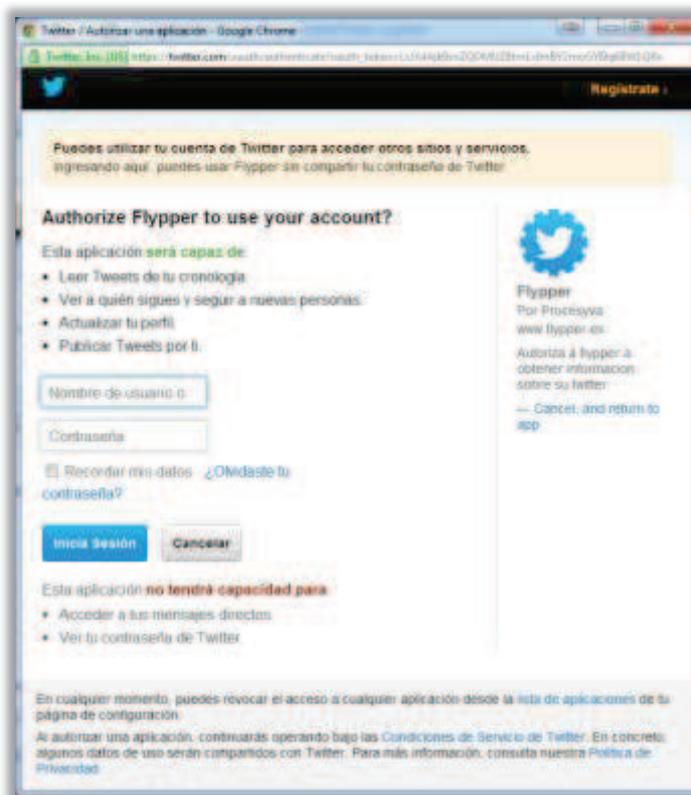


Figura 9.32. Pantalla login Twitter

Una vez que el usuario haga login en Facebook, aparecerá una pantalla para conceder los permisos a Flypper.

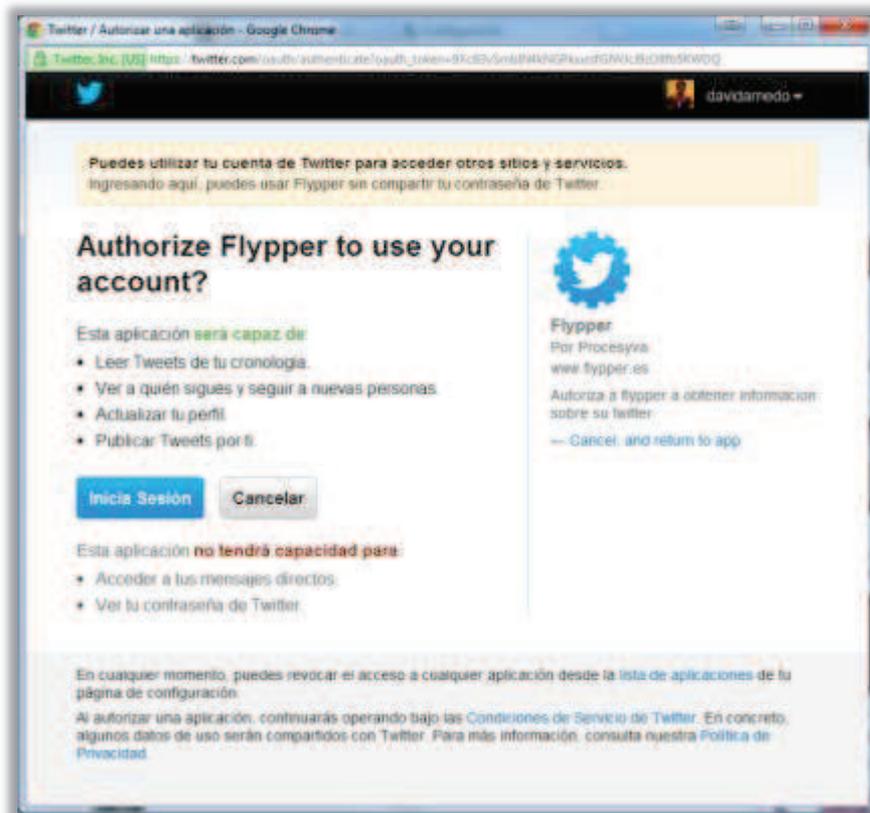


Figura 9.33. Concesión de permisos a Flypper en Twitter

Cuando se hayan concedido los permisos a la aplicación se mostrará la pantalla de inicio de Twitter. En esta pantalla podremos acceder al inicio, perfil y menciones del usuario haciendo click en las pestañas correspondientes.

Para compartir un tweet deberemos escribir en el apartado correspondiente y hacer click en "Tweet". No está permitido escribir más de 140 caracteres.

Además, podremos realizar retweets y eliminar tweets propios haciendo click en el icono correspondiente.



Figura 9.34. Pantalla de inicio de Twitter



Figura 9.35. Pantalla de perfil de Twitter



Figura 9.36. Pantalla de menciones en Twitter

9.2.2.3 Módulo Youtube

Desde las pestañas del módulo de Youtube se puede acceder a unos listados de videos tales como: recomendaciones, más vistos y trending.



Figura 9.37. Videos recomendados por Youtube



Figura 9.38. Videos más vistos en Youtube España



Figura 9.39. Videos trending en Youtube

Para realizar una búsqueda se introduce el texto de la canción o del cantante a buscar y se hace click en “Buscar”. Aparecerá un listado con todas las canciones relacionadas con la búsqueda.



Figura 9.40. Resultados de la búsqueda en Youtube

Para reproducir el vídeo hay que hacer click en el video correspondiente y se desplegará el reproductor.

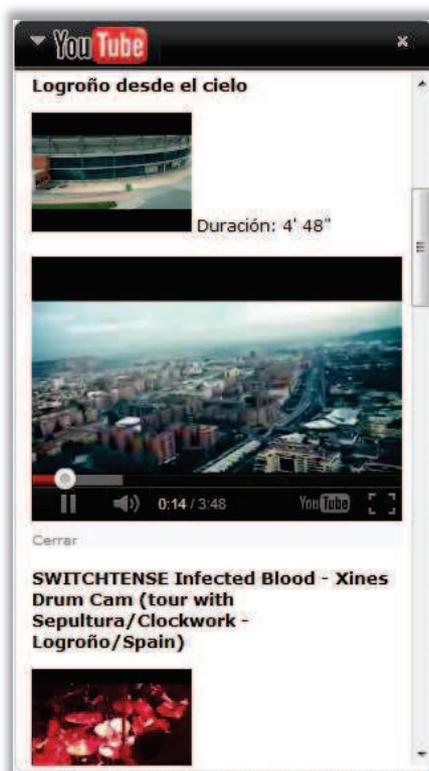


Figura 9.41. Reproducción de un video en Youtube

9.2.2.4 Módulo Spotify

Para realizar una búsqueda se introduce el texto de la canción o del cantante a buscar y se hace click en “Buscar”. Aparecerá un listado con todas las canciones relacionadas con la búsqueda.



Figura 9.42. Resultado de una búsqueda en Spotify

Tanto para reproducir la canción, como para compartir en las redes sociales de Twitter y Facebook, hay que hacer click en la canción correspondiente y se desplegará un apartado, para realizar la tarea deseada.

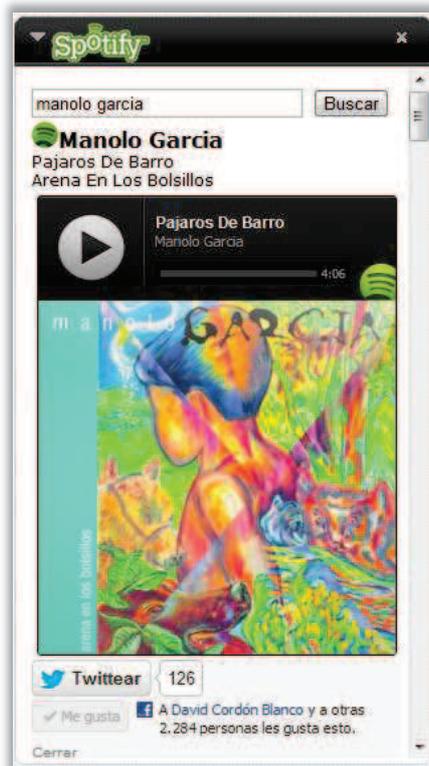


Figura 9.43. Reproductor de Spotify

9.3 Manual del Programador

La aplicación está formada por 4 módulos totalmente diferenciados. A continuación se describirá de qué está compuesto cada módulo y para qué sirve cada una de sus partes.

9.3.1 Facebook

Consta de los siguientes archivos:

- **lib_facebook:** es una librería en php que se encarga de la creación de un objeto de Facebook que nos facilita la tarea a la hora de obtener los códigos de autenticación, realizar acciones contra el API u obtener las URLs necesarias para conseguir los permisos.
- **index.php:** archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.
- **user_wall.php:** archivo que obtiene y muestra los datos del perfil de Facebook del usuario. En él se puede hacer me gusta en las diferentes publicaciones, borrarlas o comentarlas.
- **home.php:** archivo que obtiene y muestra los datos del perfil de Facebook del usuario. En él se puede hacer me gusta en las diferentes publicaciones, borrarlas o comentarlas.
- **publication.php:** archivo para mostrar los comentarios de una publicación. Además permite comentar en la misma. En él se puede hacer me gusta en las diferentes publicaciones, borrarlas o comentarlas.
- **logout.php:** archivo para realizar la desconexión de la aplicación. Además elimina de la base de datos el token de autorización del usuario.
- **actions.php:** archivo donde se encuentran las diferentes acciones que se pueden realizar en la aplicación (me gusta, publicar, comentar, borrar, etc.).
- **ajax.js:** archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Facebook sin tener que recargar la página.
- **callback.php:** archivo encargado de recoger los datos que Facebook devuelve una vez realizada la petición de autorización. Además guarda el token de autorización en la base de datos para no tener que volver a pedir al usuario la autorización.
- **dbconnection.php:** archivo que realiza la conexión a la base de datos.
- **functions.php:** archivo donde se encuentra la función para parsear las URLs de la aplicación.
- **facebook.css:** archivo de estilo para el módulo de Facebook.

9.3.2 Twitter

Consta de los siguientes archivos:

- **twitteroauth:** es una librería en php que se encarga de la creación de un objeto de Twitter que nos facilita la tarea a la hora de obtener los códigos de autenticación,

realizar acciones contra el API u obtener las URLs necesarias para conseguir los permisos.

- **index.php**: archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.
- **time_line.php**: archivo que obtiene y muestra los datos del time line de Twitter del usuario. En él se puede eliminar tus propios tweets o retwitear los tweets de otros usuarios.
- **perfil.php**: archivo que obtiene y muestra los datos del time line de Twitter del usuario. En él se puede eliminar tus tweets.
- **menciones.php**: archivo que obtiene y muestra las menciones del usuario en Twitter.
- **logout.php**: archivo para realizar la desconexión de la aplicación. Además elimina de la base de datos el token de autorización del usuario.
- **actions.php**: archivo donde se encuentran las diferentes acciones que se pueden realizar en la aplicación (Twitear, borrar un tweet o retwitear).
- **ajax.js**: archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Twitter sin tener que recargar la página.
- **callback.php**: archivo encargado de recoger los datos que Twitter devuelve una vez realizada la petición de autorización. Además guarda el token de autorización en la base de datos para no tener que volver a pedir al usuario la autorización.
- **dbconnection.php**: archivo que realiza la conexión a la base de datos.
- **functions.php**: archivo donde se encuentra la función para parsear las URLs de la aplicación.
- **twitter.css**: archivo de estilo para el módulo de Twitter.

9.3.3 Youtube

- **index.php**: archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.
- **featured.php**: archivo que obtiene y muestra los videos recomendados por Youtube.
- **popular.php**: archivo que obtiene y muestra los videos más vistos en Youtube España.
- **trending.php**: archivo que obtiene y muestra los videos más comentados en el momento en Youtube.
- **search.php**: archivo encargado de realizar la búsqueda de videos y de mostrar los resultados obtenidos en Youtube.
- **play.php**: archivo encargado de realizar la reproducción del vídeo solicitado.
- **ajax.js**: archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Youtube sin tener que recargar la página.
- **functions.php**: archivo donde se encuentra la función para obtener la id del usuario a partir de una URL y otra función para pasar los segundos a minutos y segundos.
- **youtube.css**: archivo de estilo para el módulo de Youtube.

9.3.4 Spotify

- **index.php**: archivo principal donde se incluyen todo lo necesario para que la aplicación funcione. Aquí se encuentra el menú principal de la aplicación encargado de llamar a los demás archivos para mostrar la información solicitada.
- **search.php**: archivo encargado de realizar la búsqueda de videos y de mostrar los resultados obtenidos en Youtube.
- **play.php**: archivo encargado de realizar la reproducción del vídeo solicitado.
- **ajax.js**: archivo encargado de realizar las llamadas asíncronas para cargar los diferentes apartados de Spotify sin tener que recargar la página.
- **spotify.css**: archivo de estilo para el módulo de Spotify.

9.3.5 Nuevos módulos

Si se desea crear nuevos módulos para la aplicación se añade una carpeta nueva siguiendo la siguiente estructura:

- **index.php**: archivo que se encarga de comprobar la autenticación y muestra unos datos u otros.
- **callback.php**: archivo que recoge el token de autorización en caso de que lo hubiese.
- **dbconnection.php**: archivo para realizar la conexión a la base de datos.
- **ajax.js**: comportamiento de las llamadas asíncronas del módulo en concreto.
- **style.css**: estilo css para el módulo.
- **functions.php**: funciones que va a utilizar el módulo.
- Además se añadirá un archivo por cada presentación de datos que se desee mostrar.

Capítulo 10. Conclusiones y Ampliaciones

y

10.1 Conclusiones

Una vez finalizado el proyecto “Aplicación web para la gestión agregada de distintas redes sociales y aplicaciones”, con nombre comercial “Flypper”, se han cumplido los siguientes objetivos:

- Creación de una herramienta donde se reúnan diferentes aplicaciones.
- Añadida la aplicación de Facebook.
- Añadida la aplicación de Twitter.
- Añadida la aplicación de Youtube.
- Añadida la aplicación de Spotify.
- Dotar a la aplicación con un diseño atractivo.
- Crear una aplicación que sea intuitiva y fácil de usar.

Una vez conseguida que la primera fase de Flypper sea funcional, se va a presentar la idea ante empresas para intentar conseguir capital de riesgo para juntar un equipo que trabaje en la aplicación y llegar al mayor número de usuarios posible.

Personalmente, este proyecto me ha permitido dirigir un proyecto desde sus inicios e ir aprendiendo en cada etapa del mismo. Además el desarrollo de la aplicación me ha permitido hacer hincapié en temas estudiados durante el máster como son la utilización de un sistema de gestión de contenido, la utilización de APIs, la utilización de lenguajes de programación como jQuery, html y css, o la utilización de tecnologías como AJAX.

Además, con todo lo aprendido, tengo la ilusión de que el proyecto siga adelante y poder trabajar en él.

10.2 Ampliaciones

Las labores de ampliación para este proyecto podrían ser las siguientes:

10.2.1 Creación de nuevos módulos

Actualmente, en la fase beta de la aplicación, existen 4 módulos correspondientes a las aplicaciones de Facebook, Twitter, Youtube y Spotify. Se podrían añadir otras muchas como el correo Gmail, un lector RSS, Dropbox o Evernote, entre otras muchas.

10.2.2 Crear un login para registrarse con redes sociales

Con el afán de hacer la aplicación mucho mas usable y sencilla para el usuario, se podría programar un login mediante las redes sociales. Con este tipo de login, el usuario no tendría que introducir sus datos personales para realizar su registro sino que, con sus propias redes sociales, el sistema recogería los datos del usuario y los guardaría en la base de datos para usarlos posteriormente.

10.2.3 Crear un acortador de URL

Hoy en día no se pueden desaprovechar caracteres en escribir una URL ya que en Twitter solamente se pueden escribir 140 caracteres. Además acortando las URL se le da un estilo mucho más cuidado a las publicaciones que se deseen compartir.

10.2.4 Crear un sistema para publicar el mismo contenido en varias redes sociales

Muchas veces se quiere compartir la misma publicación en varias redes sociales. Con un sistema de publicación masiva mejoraríamos la experiencia de la aplicación.

10.2.5 Programar la aplicación para prescindir de Joomla

Para optimizar la aplicación y eliminar código que no es necesario, se podría programar la aplicación para prescindir de Joomla. Así ocuparía mucho menos, sería más escalable y podríamos optimizar más su rendimiento.

10.2.6 Cumplir al menos accesibilidad AA

Para llegar al máximo número de usuarios y que todo el mundo pueda acceder a la aplicación sin problemas, debe llegarse a cumplirse al menos accesibilidad AA.

10.2.7 Aplicación móvil

Actualmente toda aplicación importante tiene su versión para móvil. Tener una versión para móvil amplia mucho el número de usuarios al que la aplicación puede llegar.

Capítulo 11. Presupuesto

Para el cálculo del presupuesto se han estimado 3 tipos de empleados con los siguientes salarios:

- Analista: 30€/h
- Diseñador: 25€/h
- Programador: 25€/h

Por otro lado, se ha estimado una jornada laboral de 8 horas diarias (40 semanales).

A continuación se muestra una tabla con el resumen aproximado de los costes del proyecto.

Item	Subitem	Concepto	Cantidad	Precio Unitario	Total
1		Desarrollo de la aplicación			14.880,00 €
	001	Análisis	136	30,00 €	4.080,00 €
	002	Diseño	136	25,00 €	3.400,00 €
	003	Implementación y pruebas	296	25,00 €	7.400,00 €
2		Software			465,00 €
	001	Windows 7 Professional	1	300,00 €	300,00 €
	002	Microsoft Office	1	0,00 €	0,00 €
	003	Eclipse	1	0,00 €	0,00 €
	004	Enterprise Architect	1	165,00 €	165,00 €
3		Hardware			600,00 €
	001	Ordenador Portátil Asus (desarrollo)	1	600,00 €	600,00 €
4		Servidor			185,00 €
	001	Hosting	1	160,00 €	160,00 €
	002	Dominio	1	25,00 €	25,00 €
5		Oficina			150,00 €
	001	Material de oficina	1	150,00 €	150,00 €
				Subtotal	16.280,00 €
				Beneficio (20%)	3.256,00 €
				IVA (18%)	2930,40 €
				TOTAL	22.466,40 €

Capítulo 12. Referencias Bibliográficas

12.1 Referencias en Internet

[Hootsuite] Wikipedia. "Hootsuite". <http://es.wikipedia.org/wiki/Hootsuite>

[Tweetdeck] Wikipedia. "Tweetdeck". <http://es.wikipedia.org/wiki/Tweetdeck>

[igoogle] Wikipedia. "iGoogle". <http://es.wikipedia.org/wiki/IGoogle>

[social] Wikipedia. "Social media". [http://es.wikipedia.org/wiki/Medio_social_\(social_media\)](http://es.wikipedia.org/wiki/Medio_social_(social_media))

[oauth] Wikipedia. "OAuth". <http://es.wikipedia.org/wiki/OAuth>

[html5] Wikipedia. "Html 5". <http://es.wikipedia.org/wiki/HTML5>

[css3] Wikipedia. "CSS 3". <http://es.wikipedia.org/wiki/Css>

[php] Wikipedia. "Php". <http://es.wikipedia.org/wiki/Php>

[mysql] Wikipedia. "MySQL". <http://es.wikipedia.org/wiki/Mysql>

[ajax] Wikipedia. "AJAX". <http://es.wikipedia.org/wiki/AJAX>

[jquery] Wikipedia. "jQuery". <http://es.wikipedia.org/wiki/Jquery>

[netbeans] Wikipedia. "NetBeans". <http://es.wikipedia.org/wiki/Netbeans>

[joomla] Wikipedia. "Joomla". <http://es.wikipedia.org/wiki/Joomla>

[curl] Wikipedia, "cURL". <http://es.wikipedia.org/wiki/CURL>

[SDK.Facebook] Facebook. "Php SDK Facebook". <http://developers.facebook.com/docs/reference/php/>

[Librería.Twitter] Abraham. "Twitter OAuth". <https://github.com/abraham/twitteroauth/wiki/documentation>

[enterprise] Sparxsystems. "Enterprise Architect". <http://www.sparxsystems.es/New/products/ea.html>

[office] Wikipedia. "Microsoft Office". http://es.wikipedia.org/wiki/Microsoft_Office

[heidi] HeidiSQL. "Heidi SQL". <http://www.heidisql.com/>

[sublime] Sublime Text 2. "Sublime Text 2". <http://www.sublimetext.com/>

[chrome] Wikipedia. "Google Chrome". http://es.wikipedia.org/wiki/Google_Chrome

[filezilla] Wikipedia. "Filezilla". <http://es.wikipedia.org/wiki/Filezilla>

Capítulo 13. Apéndices

13.1 Glosario y Diccionario de Datos

A continuación se enumeran los términos que se consideran más importantes usados en este proyecto:

- **AJAX:** tecnología para realizar cambios en la aplicación sin tener que recargar la página.
- **Facebook:** red social para compartir contenidos.
- **Flypper:** aplicación para la gestión agregada de distintas redes sociales y aplicaciones.
- **JQuery:** librería de Javascript que facilita la manera de interactuar con los documentos HTML.
- **Twitter:** red social para compartir contenidos.
- **Spotify:** aplicación para escuchar música.
- **Social:** término para englobar a las plataformas de comunicación donde el contenido es creado por los usuarios.
- **Usabilidad:** facilidad de uso con el que la aplicación se puede utilizar.
- **Youtube:** aplicación para ver vídeos.

13.2 Contenido Entregado en el CD-ROM

13.2.1 Contenidos

13.2.1.1.1 Introducción

A continuación se describe el contenido del CD (directorios y para qué sirve cada cosa).

13.2.1.1.2 Estructura general directorios del CD

Directorio	Contenido
<i>./ Directorio raíz del CD</i>	Contiene un fichero leeme.txt explicando toda esta estructura.
<i>./Flypper</i>	Contiene toda la estructura de directorios del proyecto para desarrollo.
<i>./instalacion</i>	Ficheros utilizados para la instalación del proyecto.
<i>./documentacion</i>	Contiene toda la documentación asociada al proyecto.
<i>./documentacion/img</i>	Directorio que contiene las imágenes utilizadas en la documentación.
<i>./documentacion/uml</i>	Ficheros que genera la herramienta la que se han generado los diagramas UML.
<i>./presentacion</i>	Directorio que contiene la presentación en <i>PowerPoint</i> .
<i>./herramientas</i>	Contiene los ficheros de instalación de las herramientas utilizadas para el desarrollo o puesta en marcha del proyecto (lógicamente sólo las que sean distribuibles).
<i>./herram/desarrollo</i>	Ficheros de instalación de las herramientas utilizadas en el desarrollo
<i>./herram/explotacion</i>	BD, servidor Web y herramientas en general.

13.2.1.1.3 Estructura de Directorios de “desarrollo”

Se muestra aquí el contenido del directorio de desarrollo de la tabla anterior, incluyendo todos los directorios que deben depender del mismo.

Directorio	Contenido
<i>./ Facebook</i>	Contiene todos los archivos pertenecientes al módulo de Facebook.
<i>./Twitter</i>	Contiene todos los archivos pertenecientes al módulo de Twitter.
<i>./Youtube</i>	Contiene todos los archivos pertenecientes al módulo de Youtube.
<i>./Spotify</i>	Contiene todos los archivos pertenecientes al módulo de Spotify.

13.2.2 Código Ejecutable e Instalación

- Instalar Joomla en un servidor.
- Copiar las carpetas de los módulos dentro del servidor.
- Crear módulos wrappers dentro de Joomla apuntando a cada uno de los módulos.
- Configurar Drag&Drop con los módulos.

13.3 Índice Alfabético

A

AJAX, 7, 9, 11, 23, 30, 31, 41, 102, 171, 175, 177

F

Facebook, 3, 7, 8, 9, 10, 11, 12, 15, 18, 20, 21, 22, 25, 26, 39, 40, 41, 42, 44, 45, 50, 51, 53, 54, 59, 60, 61, 62, 63, 70, 71, 75, 76, 78, 80, 85, 86, 87, 88, 89, 90, 91, 99, 103, 106, 107, 113, 115, 120, 121, 122, 123, 146, 154, 155, 156, 157, 159, 166, 168, 171, 175, 177

Flypper, 3, 9, 11, 12, 15, 19, 20, 26, 53, 54, 55, 56, 57, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 93, 94, 95, 96, 100, 124, 149, 150, 151, 152, 154, 155, 157, 159, 160, 171, 177

J

JQuery, 177

S

Social, 7, 11, 25, 175, 177

Spotify, 3, 9, 10, 11, 12, 15, 20, 25, 39, 41, 42, 49, 50, 51, 58, 59, 69, 70, 71, 78, 81, 90, 99, 110, 119, 122, 123, 146, 166, 167, 170, 171, 177

T

Twitter, 3, 8, 9, 10, 11, 12, 15, 20, 21, 22, 25, 26, 39, 40, 41, 42, 46, 47, 50, 51, 55, 56, 57, 64, 65, 66, 67, 71, 76, 77, 78, 80, 81, 88, 89, 91, 99, 103, 107, 108, 109, 115, 116, 117, 120, 122, 123, 146, 157, 158, 159, 160, 161, 162, 166, 169, 171, 172, 175, 177

U

Usabilidad, 9, 10, 41, 97, 98, 121, 177

Y

Youtube, 3, 9, 10, 11, 12, 15, 20, 25, 39, 41, 42, 48, 49, 51, 57, 58, 67, 68, 69, 77, 81, 89, 90, 99, 109, 110, 117, 118, 122, 123, 146, 162, 163, 164, 165, 169, 170, 171, 177

13.4 Código Fuente

13.4.1 Paquete Facebook

13.4.1.1 index.php

```

<?php
require_once './lib_facebook/facebook.php';
include 'dbconnection.php';

//Creamos la instancia de la aplicacion
if (!isset($facebook)) {
    $facebook = new Facebook(array(
        'appId' => '114470661977025',
        'secret' => '2bbalad4a08162b3fccc418625a8d409',
        'cookie' => true,
    ));
}
//Obtenemos la autorización de la BD
$oauth_token_fb = mysql_query("SELECT oauth_token_fb FROM jos_users WHERE id=" .
$id_user, $conexion);
$_SESSION['oauth_token_fb'] = mysql_result($oauth_token_fb, 0);
$facebook->setAccessToken($_SESSION['oauth_token_fb']);
// Obtenemos el ID usuario
$user = $facebook->getUser();
if ($user) {
    try {
        $user_profile = $facebook->api('/me');
    } catch (FacebookApiException $e) {
        error_log($e);
        $user = null;
    }
}

// LoginURL
$loginUrl = $facebook->getLoginUrl(array('scope' =>
'offline_access,read_stream,email,publish_stream,status_update,user_birthday,user_location,user_work_history, user_online_presence'));
?>
<!doctype html>
<html>
<head>
<title>Flypper-Facebook</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link href="facebook.css" rel="stylesheet" type="text/css" />
<script type="text/javascript"
src='http://ajax.googleapis.com/ajax/libs/jquery/1.3.2/jquery.min.js'></script>
<script type="text/javascript" src='ajax.js'></script>
<script>$(document).ready(function() {
    $('#connect').click(function() {
        $.oauthpopup({
            path: '<? echo $loginUrl; ?>',
            callback: function() {
                window.location.reload();
            }
        });
    });
});
</script>
</head>
<body>

<!-- Informacion del usuario -->

<?php if ($user): ?>
<div class="content">
<div class="user">

<p><a href='https://www.facebook.com/<? echo
$user_profile['username']; ?>' alt='<? echo $user_profile['name']; ?>'

```

```
target='_blank'></a></p>
<p id="name"><a href="https://www.facebook.com/<? echo
$user_profile['username']; ?>' target='_blank'><? echo $user_profile['name']; ?></a></p>
<p><a href="https://www.facebook.com/<? echo
$user_profile['username']; ?>' target='_blank'><? echo $user_profile['username'];
?></a></p>
<?
if (isset($user_profile['hometown']['name'])) {
    echo $user_profile['hometown']['name'];
}
?>
</div>
<form id="form" name="form"
action="javascript:cargarConParametros ('comment_wall', 'actions.php', document.form);">
    <input type="text" id="buscar" autocomplete="off" name="publish"
placeholder="¿Qué estas pensando?">
    <input type="submit" value="Publicar">
</form>
<!-- Menu -->
<div class="menu">
    <a href="#" alt="Inicio"
onclick="cargarSinParametros ('contenido', 'home.php');">Inicio</a>
    <a href="#" alt="Perfil"
onclick="cargarSinParametros ('contenido', 'user_wall.php');">Perfil</a>
    <a href="logout.php" alt="Cerrar sesión" >Cerrar sesión</a>
</div>
<div class="cargando" id="cargando">
    
</div>
<div id="contenido"></div>
<script type="text/javascript">
cargarSinParametros ("contenido", "home.php")
</script>
</div>
<?php else: ?>
    <input type="button" id="connect" />
<?php endif ?>
</body>
</html>
```

13.4.1.2 user_wall.php

```
<?
include './lib_facebook/facebook.php';
include 'functions.php';
session_start();
$facebook = new Facebook(array(
    'appId' => '114470661977025',
    'secret' => '2bba1ad4a08162b3fccc418625a8d409',
    'cookie' => true,
));
$facebook->setAccessToken($_SESSION['oauth_token_fb']);
$user = $facebook->getUser();
//Muestra el perfil en Facebook
$home = $facebook->api('/me/feed?' . $_SESSION['fb_114470661977025_access_token']);
?>
<div class="columna">
<?
foreach ($home['data'] as $new) {
    $user_like = false;
    //ID de la publicacion
    $id = $new['id'];
    ?>
    <div id="publi_<? echo $id;?>">
        <?
        //Nombre del usuario
        echo "<div class='user'>";
        echo "<a href='https://www.facebook.com/' . $new['from']['id'] . '"
target='_blank'><img src='https://graph.facebook.com/' . $new['from']['id'] . "/picture'
alt='\" . $new['from']['name'] . \"'></a>";
        echo "<a href='https://www.facebook.com/' . $new['from']['id'] . '"
target='_blank'>\" . $new['from']['name'] . \"</a>";
        //Nombre del destinatario
```

```

        if (isset($new['to']['data'])) {
            echo " > ";
            foreach ($new['to']['data'] as $to) {
                echo "<a href='https://www.facebook.com/" . $to['id'] . "'
target='_blank'>" . $to['name'] . "</a>";
            }
        }
        echo "</div>";

        //Mensaje de la publicacion
        echo "<div class='message'>";
        echo parseUrl($new['message']);
        echo "</div>";

        //Share
        if (isset($new['description']) || isset($new['picture']) || isset($new['link']))
    {
        echo "<div class='share'>";
        if (isset($new['picture'])) {
            //Imagen
            echo "<a href='" . $new['link'] . "' target='_blank'><img src='" .
$new['picture'] . "' alt='" . $new['from']['name'] . "' /></a>";
        }
        //Link
        echo "<a href='" . $new['link'] . "' target='_blank'>" . $new['name'] .
"</a>";

        //Description
        echo "<p>" . $new['description'] . "</p>";
        echo "</div>";
    }
    if ($user == $new['from']['id']) {
        ?>
        <div class='eliminar'>
            <a onclick="javascript:borrar('delete','actions.php','<? echo $id;
?>');">Eliminar</a>
        </div>
        <?
    }
    ?>
    <div class="barra">
        <?
        //Recorro el array de personas que le gusta la publicacion y si alguna es
igual al usuario hago break para no recorrer mas
        foreach ($new['likes']['data'] as $like) {
            if ($user == $like['id']) {
                ?>
                <!--Boton ya no me gusta-->
                <div id="void" style="display:none;"></div>
                <input type="button" value="Ya no me gusta" class="like"
id="dontlike_<? echo $id; ?>" onclick="javascript:like('dontlike','actions.php','<? echo
$id; ?>');" />
                <input type="button" value="Me gusta" class="like" id="like_<? echo
$id; ?>" style="display:none;" onclick="javascript:like('like','actions.php','<? echo
$id; ?>');" />
                <?
                $user_like = true;
                break;
            }
        }

        if ($user_like != true) {
            ?>
            <!--Boton me gusta-->
            <div id="void" style="display:none;"></div>
            <input type="button" value="Me gusta" class="like" id="like_<? echo $id;
?>" onclick="javascript:like('like','actions.php','<? echo $id; ?>');" />
            <input type="button" value="Ya no me gusta" class="like" id="dontlike_<?
echo $id; ?>" style="display:none;"
onclick="javascript:like('dontlike','actions.php','<? echo $id; ?>');" />
            <?
        }

        //Count de comments
        $coments = 0;
        if (isset($new['comments']['count']))
            $coments = $new['comments']['count'];
        //Count likes
    
```

```

        $likes = 0;
        if (isset($new['likes']['count']))
            $likes = $new['likes']['count'];
        ?>
        <div class="std">
            <a onclick="cargarPublication('publication.php', '<? echo $id;
?>', 'user_wall')">
                <? echo
$coments; ?>
                <? echo
$likes; ?>
            </a>
        </div>

    </div>
    <div class='cargando' id='cargando<? echo $id; ?>'>
        <img src='http://flypper.es/flypper/api/fb/img/cargando.gif' alt="Cargando">
    </div>
    <div class='slide' id='publication<? echo $id; ?>'></div>
</div>
<? }
?>
</div>

```

13.4.1.3 home.php

```

<?
include './lib_facebook/facebook.php';
include 'functions.php';
session_start();
$facebook = new Facebook(array(
    'appId' => '114470661977025',
    'secret' => '2bbalad4a08162b3fccc418625a8d409',
    'cookie' => true,
));
//Obtengo la autorización
$facebook->setAccessToken($_SESSION['oauth_token_fb']);
//ID usuario
$user = $facebook->getUser();
//Datos de inicio del usuario
$home = $facebook->api('/me/home?' . $_SESSION['fb_114470661977025_access_token']);
?>
<div class="columna">
    <?
        foreach ($home['data'] as $new) {

            $user_like = false;
            //ID de la publicacion
            $id = $new['id'];
            ?>
            <div id="publi_<? echo $id; ?>">
                <?
                    //Nombre y foto del usuario
                    echo "<div class='user'>";
                    echo "<a href='https://www.facebook.com/" . $new['from']['id'] . "'
target='_blank'><img src='https://graph.facebook.com/" . $new['from']['id'] .
"/picture'alt='" . $new['from']['name'] . "'></a>";
                    echo "<a href='https://www.facebook.com/" . $new['from']['id'] . "'
target='_blank'>" . $new['from']['name'] . "</a>";
                    //Nombre del destinatario
                    if (isset($new['to']['data'])) {
                        echo " > ";
                        foreach ($new['to']['data'] as $to) {
                            echo "<a href='https://www.facebook.com/" . $to['id'] . "'
target='_blank'>" . $to['name'] . "</a>";
                        }
                    }
                    echo "</div>";

                    //Mensaje de la publicacion
                    echo "<div class='message'>";
                    echo parseUrl($new['message']);
                    echo "</div>";
                </?
            </div>
        }
    </?

```

```

        //Share
        if (isset($new['description']) || isset($new['picture']) ||
isset($new['link'])) {
            echo "<div class='share'>";
            if (isset($new['picture'])) {
                //Imagen
                echo "<a href='" . $new['link'] . "' target='_blank'><img src='" .
$new['picture'] . "' alt='" . $new['from']['name'] . "' /></a>";
            }
            //Link
            echo "<a href='" . $new['link'] . "' target='_blank'>" . $new['name'] .
"</a>";

            //Description
            echo "<p>" . $new['description'] . "</p>";
            echo "</div>";
        }
        //Eliminar mensaje
        if ($user == $new['from']['id']) {
            ?>
            <div class='eliminar'>
                <a onclick="javascript:borrar('delete','actions.php','<? echo $id;
?>');">Eliminar</a>
            </div>
            <?
        }
        ?>
        <div class="barra">
            <?
            //Recorro el array de personas que le gusta la publicacion y si alguna
es igual al usuario hago break para no recorrer mas
            foreach ($new['likes']['data'] as $like) {
                if ($user == $like['id']) {
                    ?>
                    <!--Boton ya no me gusta-->
                    <div id="void" style="display:none;"></div>
                    <input type="button" value="Ya no me gusta" class="like"
id="dontlike_<? echo $id; ?>" onclick="javascript:like('dontlike','actions.php','<? echo
$id; ?>');"/>
                    <input type="button" value="Me gusta" class="like" id="like_<?
echo $id; ?>" style="display:none;" onclick="javascript:like('like','actions.php','<?
echo $id; ?>');"/>
                    <?
                    $user_like = true;
                    break;
                }
            }

            if ($user_like != true) {
                ?>
                <!--Boton me gusta-->
                <div id="void" style="display:none;"></div>
                <input type="button" value="Me gusta" class="like" id="like_<? echo
$id; ?>" onclick="javascript:like('like','actions.php','<? echo $id; ?>');"/>
                <input type="button" value="Ya no me gusta" class="like"
id="dontlike_<? echo $id; ?>" style="display:none;"
onclick="javascript:like('dontlike','actions.php','<? echo $id; ?>');"/>
                <?
            }

            //Count de comments
            $coments = 0;
            if (isset($new['comments']['count']))
                $coments = $new['comments']['count'];
            //Count likes
            $likes = 0;
            if (isset($new['likes']['count']))
                $likes = $new['likes']['count'];
            ?>
            <div class="std">
                <a onclick="cargarPublication('publication.php','<? echo $id;
?>', 'user_wall')">
                    <?
echo $coments; ?>
                    <? echo
$likes; ?>
                </a>
            </div>

```

```

        </div>
        <div class='cargando' id='cargando<? echo $id; ?>'>
            <img src='http://flypper.es/flypper/api/fb/img/cargando.gif'
alt="cargando">
        </div>
        <div class='slide' id='publication<? echo $id; ?>'></div>
    </div>
    <? }
    ?>
</div>

```

13.4.1.4 publication.php

```

<?
include './lib_facebook/facebook.php';
session_start();
$facebook = new Facebook(array(
    'appId' => '114470661977025',
    'secret' => '2bba1ad4a08162b3fccc418625a8d409',
    'cookie' => true,
));
$facebook->setAccessToken($_SESSION['oauth_token_fb']);
//ID del usuario
$user = $facebook->getUser();
//Obtener el id de la publicacion
$id = $_POST['id'];
//Parseo el id para que sea válido
$id_exp = explode("_", $id);
$id = $id_exp[0] . "_" . $id_exp[1];

//Obtienes la publicacion
$publication = $facebook->api('/' . $id . '?' .
$_SESSION['fb_114470661977025_access_token']);

//Comentarios
foreach ($publication['comments']['data'] as $comment) {
    ?>
    <div id="slide_<? echo $comment["id"]; ?>">
        <div class="user">
            <!--Imagen del usuario -->
            <a href="https://www.facebook.com/<? echo $comment['from']['id']; ?>"
target="_blank">
                "
            </a>
            <!--Nombre del usuario -->
            <p class="user_link"><a href="https://www.facebook.com/<? echo
$comment['from']['id']; ?>" target="_blank">
                <? echo $comment['from']['name']; ?></a>
            </p>
        </div>

        <!--Mensaje -->
        <div class="message">
            <p>
                <?
                echo $comment['message'];
                ?>
            </p>
            <?
            if ($user == $comment['from']['id']) {
                ?>
                <!--Eliminar mensaje -->
                <a onclick="javascript:borrarPublication('delete','actions.php','<? echo
$comment["id"]; ?>');">Eliminar</a>
                <?
            }
            ?>
        </div>
        <div class="barra">
            <?
            //Obtengo el comentario

```

```

        $comment_pub = $facebook->api('/', . $comment['id'] . '?' .
$_SESSION['fb_114470661977025_access_token']);
        if (isset($comment_pub['user_likes'])) {
            ?>
            <!--Boton ya no me gusta-->
            <div id="void" style="display:none;"></div>
            <input type="button" value="Ya no me gusta" class="like"
style="display:none;" id="dontlike_<? echo $comment['id']; ?>"
onclick="javascript:like('dontlike','actions.php','<? echo $comment['id']; ?>');"/>
            <input type="button" value="Me gusta" class="like" id="like_<? echo
$comment['id']; ?>" onclick="javascript:like('like','actions.php','<? echo
$comment['id']; ?>');"/>

            <?
        } else {
            ?>
            <!--Boton me gusta-->
            <div id="void" style="display:none;"></div>
            <input type="button" value="Ya no me gusta" class="like" id="dontlike_<?
echo $comment['id']; ?>" onclick="javascript:like('dontlike','actions.php','<? echo
$comment['id']; ?>');"/>
            <input type="button" value="Me gusta" class="like" style="display:none;"
id="like_<? echo $comment['id']; ?>" onclick="javascript:like('like','actions.php','<?
echo $comment['id']; ?>');"/>

            <?
        };
        if (isset($comment['likes'])) {
            $likes = $comment['likes'];
        }
        else
            $likes = 0;
        ?>
        <!-- Numero de me gusta -->
        <div class="std">
            <? echo $likes;
?>

        </div>

        </div>
    </div>
    <?
}
?>
<!-- Comentar publicacion -->
<form id="form<? echo $id; ?>" name="form<? echo $id; ?>"
action="javascript:commentPublication('actions.php','<? echo $id;
?>','comment_post',document.form<? echo $id; ?>');">
    <textarea id="text" rows="3" cols="30" name="mensaje"></textarea>
    <input type="submit" value="Enviar"/>
</form>
<div class="cerrar" onclick="cerrarPublication('<? echo $id; ?>')">
    Cerrar
</div>

```

13.4.1.5 logout.php

```

<?
include './lib_facebook/facebook.php';
require_once('dbconnection.php');
mysql_query("UPDATE jos_users SET oauth_token_fb=NULL WHERE id=".$id_user,$conexion);
header('Location: ./index.php');
?>

```

13.4.1.6 actions.php

```

<?php
session_start();
$action = $_POST['action'];
include './lib_facebook/facebook.php';
//Creamos la instancia de la aplicacion

```

```
if (!isset($facebook)) {
    $facebook = new Facebook(array(
        'appId' => '114470661977025',
        'secret' => '2bba1ad4a08162b3fccc418625a8d409',
        'cookie' => true,
    ));
}
$facebook->setAccessToken($_SESSION['oauth_token_fb']);
switch ($action) {
    //Añadir publicacion
    case "comment_wall":
        $mensaje = $_POST['message'];
        $params = array('access_token' => $_SESSION['fb_114470661977025_access_token'],
'message' => $mensaje);
        $facebook->api('/me/feed', 'POST', $params);
    //Comentar un post
    case "comment_post":
        $mensaje = $_POST['message'];
        $params = array('access_token' => $_SESSION['fb_114470661977025_access_token'],
'message' => $mensaje);
        $facebook->api('/' . $_POST['id'] . '/comments', 'POST', $params);
        break;
    //Me gusta
    case "like":
        $params = array('access_token' => $_SESSION['fb_114470661977025_access_token']);
        $facebook->api('/' . $_POST['id'] . '/likes', 'POST', $params);
        break;
    //Ya no me gusta
    case "dontlike":
        $params = array('access_token' => $_SESSION['fb_114470661977025_access_token']);
        $facebook->api('/' . $_POST['id'] . '/likes', 'DELETE', $params);
        break;
    //Eliminar objeto
    case "delete":
        $params = array('access_token' => $_SESSION['fb_114470661977025_access_token']);
        $facebook->api('/' . $_POST['id'], 'DELETE', $params);
        break;

    default:
        break;
}
?>
```

13.4.1.7 ajax.js

```
//Pop up oauth
(function(jQuery){
    jQuery.oauthpopup = function(options)
    {
        options.windowName = options.windowName || 'ConnectWithOAuth'; // should not
        include space for IE
        options.windowOptions = options.windowOptions ||
        'location=0,status=0,width=800,height=700';
        options.callback = options.callback || function(){ window.location.reload(); };
        var that = this;

        that._oauthWindow = window.open(options.path, options.windowName,
        options.windowOptions);
        that._oauthInterval = window.setInterval(function(){
            if (that._oauthWindow.closed) {
                window.clearInterval(that._oauthInterval);
                options.callback();
            }
        }, 1000);
    };
})(jQuery);
//Función para cargar un php sin enviar parámetros
function cargarSinParametros(div,pagina){
    $('#contenido').hide();
    $('#cargando').show();
    $('#'+div).load(pagina, function(){
        $('#cargando').hide();
        $('#contenido').fadeIn('slow');
    });
}
//Función para cargar un php enviando parámetros
function cargarConParametros(action,pagina,comment){
    if(comment.elements[0].value != ''){
        $('#contenido').hide();
        $('#cargando').show();
        $('#contenido').load(pagina, {
            message:comment.elements[0].value,
            action:action
        }, function(){
            $('#contenido').load('home.php', function(){
                $('#cargando').hide();
                $('#form')[0].reset();
                $('#contenido').fadeIn('slow');
            });
        });
    }
}
//Función para borrar mensajes
function borrar(action,pagina,id){
    $('#publi_'+id).hide();
    $('#void').load(pagina, {
        action:action,
        id:id
    });
}
//Función me gusta
function like(action,pagina,id){
    if (action=='dontlike') {
        $('#dontlike_'+id).css("display", "none");
        $('#like_'+id).css("display", "block");
        $('#void').load(pagina, {
            action:action,
            id:id
        });
    }
    else{
        $('#like_'+id).css("display", "none");
        $('#dontlike_'+id).css("display", "block");
        $('#void').load(pagina, {
            action:action,
            id:id
        });
    }
}
}
```

```
//Función para ver una publicación
function cargarPublication(pagina, id_comment, option){
    option = option || "home";
    if ($("#publication"+id_comment).is(":hidden")) {
        $("#cargando"+id_comment).show();
        $("#publication"+id_comment).load(pagina, {
            id:id_comment,
            option:option
        }, function(){
            $("#cargando"+id_comment).hide();
            $("#publication"+id_comment).slideDown("slow");
        });
        $("#text"+id_comment).focus();
    }
    else {
        $("#publication"+id_comment).slideUp();
    }
}
//Función para cerrar una publicación
function cerrarPublication(id_comment){
    $("#publication"+id_comment).slideUp();
}
//Función para comentar una publicación
function commentPublication(pagina, id, action, comment){
    $("#publication"+id).hide("slow");
    $("#cargando"+id).show();
    $("#void").load(pagina, {
        id:id,
        action:action,
        message:comment.elements[0].value
    });
    $("#publication"+id).load('publication.php', {
        id:id
    }, function(){
        $("#cargando"+id).hide();
        $("#publication"+id).slideDown("slow");
    });
}
//Función para borrar una publicación
function borrarPublication(action,pagina,id){
    $("#slide_"+id).hide();
    $("#void").load(pagina, {
        action:action,
        id:id
    });
}
}
```

13.4.1.8 callback.php

```
<html>
<head>
<title>Flypper - Auth Facebook</title>
</head>
<body>
    <?
    session_start();
    include './lib_facebook/facebook.php';
    include 'dbconnection.php';
    //Creamos la instancia de la aplicacion
    $facebook = new Facebook(array(
        'appId' => '114470661977025',
        'secret' => '2bbalad4a08162b3fccc418625a8d409',
        'cookie' => true,
    ));
    //Obtenemos el token de autorización
    $access_token=$facebook->getAccessToken();
    //Guardamos en la BD
    mysql_query("UPDATE jos_users SET oauth_token_fb='".$access_token.'" WHERE
id=".$sid_user,$conexion);
    ?>
    <script>
        //Cierro la ventana de autorización
        window.close();
    </script>
</body>
</html>
```

```
</script>
</body>
</html>
```

13.4.1.9 functions.php

```
<?php
//Funcion para dar formato a las URLs
function parseUrl($text) {
    $text = preg_replace('/(https?:\\/\\/\\S+)/', '<a href="\1"
target="_blank">\1</a>', $text);
    return $text;
}
?>
```

13.4.2 Paquete Twitter

13.4.2.1 index.php

```
<?php
/*
    index.php es el encargado de comprobar si el usuario ha sincronizado twitter con
    flypper
    Tambien se encarga de mostrar las diferentes opciones de twitter.
*/
session_start();
require_once('twitteroauth/twitteroauth.php');
require_once('twitteroauth/OAuth.php');
require_once('config.php');
require_once('dbconnection.php');

//Obtengo de la bd el oauth_token y oauth_token_secret
$oauth_token = mysql_query("SELECT oauth_token FROM jos_users WHERE id=" . $id_user,
$conexion);
$oauth_token_secret = mysql_query("SELECT oauth_token_secret FROM jos_users WHERE id=" .
$id_user, $conexion);
$_SESSION['oauth_token'] = mysql_result($oauth_token, 0);
$_SESSION['oauth_token_secret'] = mysql_result($oauth_token_secret, 0);
?>
<!doctype html>
<html>
    <head>
        <title>Flypper - Twitter</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <link href="twitter.css" rel="stylesheet" type="text/css" />
        <script type="text/javascript"
src='http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js'></script>
        <script type="text/javascript" src='ajax.js'></script>
    </head>
    <body>
        <?
            //Creo el objeto TwitterOAuth con los tokens de autorizacion para poder consumir
            los servicios del API
            $connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET,
            $_SESSION['oauth_token'], $_SESSION['oauth_token_secret']);
            /* Obtener los datos del usuario */
            $user = $connection->get('account/verify_credentials');
            //Comprueba si están disponibles los tokens de autorizacion
            if ($_SESSION['oauth_token'] != NULL && $_SESSION['oauth_token_secret'] != NULL
            && isset($user->id) && $user->id != null) {
                ?>
                <div class="content">
                    <!-- Usuario -->
                    <div class="user">
```

```

        
        <p><? echo $user->name; ?></p>
        <p><? echo "@" . $user->screen_name; ?></p>
    </div>
    <!-- Formulario para publicar -->
    <form id="form" name="form"
action="javascript:twitear('tweet','contenido','actions.php',document.form);">
        <input type="text" id="buscar" maxlength="140" autocomplete="off"
name="buscar" placeholder="¿Qué está pasando?">
        <input type="submit" value="Tweet">
    </form>
    <!-- Menu -->
    <div class="menu">
        <a href="#"
onclick="cargarSinParametros('contenido','time_line.php');">Inicio</a>
        <a href="#"
onclick="cargarSinParametros('contenido','perfil.php');">Perfil</a>
        <a href="#"
onclick="cargarSinParametros('contenido','menciones.php');">Menciones</a>
        <a href='./logout.php'>Cerrar sesión</a>
    </div>
    <div class="cargando" id="cargando">
        
    </div>
    <div id="contenido"></div>
    <div id="void"></div>
</div>
<script type="text/javascript">
    cargarSinParametros("contenido","time_line.php");
</script>
<?
} else {
    /* Creamos el objeto Twitter */
    $connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET);
    /* Obtenemos las credenciales temporales */
    $request_token = $connection->getRequestToken(OAUTH_CALLBACK);

    //Guardamos en la base de datos las credenciales temporales
    $result = mysql_query("UPDATE jos_users SET oauth_token='\" .
$request_token['oauth_token'] . "\" ,oauth_token_secret='\" .
$request_token['oauth_token_secret'] . "\" WHERE id='\" . $id_user, $conexion);

    //Token necesario para redirigir a twitter
    $token = $request_token['oauth_token'];
    $url = $connection->getAuthorizeURL($token);
    ?>
<script>
    $(document).ready(function() {
        $('#connect').click(function() {
            $.oauthpopup({
                path: '<? echo $url; ?>',
                callback: function() {
                    window.location.reload();
                }
            });
        });
    });
</script>

<input type="button" id="connect" />
<?
}
?>
</body>
</html>

```

13.4.2.2 time_line.php

```

<?
session_start();
include('config.php');
include('functions.php');
require_once('twitteroauth/twitteroauth.php');

```

```

require_once('twitteroauth/OAuth.php');

$connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET, $_SESSION['oauth_token'],
$_SESSION['oauth_token_secret']);
$user = $connection->get('account/verify_credentials');
/* Obtener los tweets del timeline */
$home_tl = $connection->get('statuses/home_timeline', array('count' => 30, 'include_rts'
=> true));
/* Muestro los tweets del timeline */
for ($i = 0; $i < count($home_tl); $i++) {
    ?>
    <div class="tweet" id="tweet_<? echo $home_tl[$i]->id; ?>">
        <?
        if ($user->name == $home_tl[$i]->user->name) {
            ?>
            <div class="rt">
                <a onclick="javascript:borrarTweet('delete_tweet','actions.php','<? echo
$home_tl[$i]->id; ?>');">
                    
                </a>
            </div>
            <?
        }
        if ($user->name != $home_tl[$i]->user->name) {
            ?>
            <div class="rt">
                <a onclick="javascript:retweet('retweet','actions.php','<? echo
$home_tl[$i]->id; ?>');">
                    
                </a>
            </div>
            <?
        }
    ?>
    <img src='<? echo $home_tl[$i]->user->profile_image_url; ?>'>
    <p id="user">
        <a href='http://twitter.com/<? echo $home_tl[$i]->user->screen_name; ?>'
target='_blank'>@<? echo $home_tl[$i]->user->screen_name; ?></a>
    </p>
    <p id="name"><? echo $home_tl[$i]->user->name; ?></p>
    <div id="text"><? echo formatTweet($home_tl[$i]->text); ?></div>

    </div>
    <?
}
?>

```

13.4.2.3 perfil.php

```

<?
session_start();
include('config.php');
include('functions.php');
require_once('twitteroauth/twitteroauth.php');
require_once('twitteroauth/OAuth.php');

$connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET, $_SESSION['oauth_token'],
$_SESSION['oauth_token_secret']);
$user = $connection->get('account/verify_credentials');
/* Obtener los tweets escritos por el usuario */
$user_tl = $connection->get('statuses/user_timeline', array('count' => 30, 'include_rts'
=> true));
/* Muestro los tweets escritos por el usuario */
for ($i = 0; $i < count($user_tl); $i++) {
    ?>
    <div class="tweet" id="tweet_<? echo $user_tl[$i]->id; ?>">
        <?
        if ($user->name == $user_tl[$i]->user->name) {

```

```

        ?>
        <div class="rt">
            <a onclick="javascript:borrarTweet('delete_tweet','actions.php','<? echo
$user_tl[$i]->id; ?>');">
                
            </a>
        </div>
        <?
    }
    ?>
    <img src='<? echo $user_tl[$i]->user->profile_image_url; ?>'>
    <p id="user">
        <a href='http://twitter.com/<? echo $user_tl[$i]->user->screen_name; ?>'
target='_blank'>@<? echo $user_tl[$i]->user->screen_name; ?></a>
    </p>
    <p id="name"><? echo $user_tl[$i]->user->name; ?></p>
    <div id="text"><? echo formatTweet($user_tl[$i]->text); ?></div>

</div>
<?
}
?>

```

13.4.2.4 menciones.php

```

<?
session_start();
include('config.php');
include('functions.php');
require_once('twitteroauth/twitteroauth.php');
require_once('twitteroauth/OAuth.php');

$connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET, $_SESSION['oauth_token'],
$_SESSION['oauth_token_secret']);

/* Obtener las menciones del usuario */
$menciones = $connection->get('statuses/mentions', array('count' => 30, 'include_rts' =>
true));
/* Muestro las menciones del usuario */
for ($i = 0; $i < count($menciones); $i++) {
    ?>
    <div class="tweet">
        <img src='<? echo $menciones[$i]->user->profile_image_url; ?>'>
        <p id="user">
            <a href='http://twitter.com/<? echo $menciones[$i]->user->screen_name; ?>'
target='_blank'>@<? echo $menciones[$i]->user->screen_name; ?></a>
        </p>
        <p id="name"><? echo $menciones[$i]->user->name; ?></p>
        <div id="text"><? echo formatTweet($menciones[$i]->text); ?></div>
    </div>
    <?
}
?>

```

13.4.2.5 actions.php

```

<?php

//Acción a realizar
$action = $_POST['action'];
//ID del post
$id = $_POST['id'];

require_once('twitteroauth/twitteroauth.php');
require_once('twitteroauth/OAuth.php');
require_once('config.php');
session_start();
//Obtengo el objeto de Twitter
$connection = new TwitterOAuth(CONSUMER_KEY, CONSUMER_SECRET, $_SESSION['oauth_token'],
$_SESSION['oauth_token_secret']);

```

```

switch ($action) {
//Twtitear
case "tweet":
    $connection->post('statuses/update', array('status' => $_POST["tweet"]));
    echo "<div class='exito'>Mensaje retwiteado</div>";
    break;
//Eliminar tweet
case "delete_tweet":
    $connection->delete('statuses/destroy/' . $id);
    break;
//Retwitear
case "retweet":
    $connection->post("statuses/retweet/" . $id);
    break;
}
?>

```

13.4.2.6 ajax.js

```

//Pop up oauth
(function(jQuery) {
    jQuery.oauthpopup = function(options)
    {
        options.windowName = options.windowName || 'ConnectWithOAuth'; // should not
include space for IE
        options.windowOptions = options.windowOptions ||
'location=0,status=0,width=800,height=700';
        options.callback = options.callback || function(){
            window.location.reload();
        };
        var that = this;

        that._oauthWindow = window.open(options.path, options.windowName,
options.windowOptions);
        that._oauthInterval = window.setInterval(function(){
            if (that._oauthWindow.closed) {
                window.clearInterval(that._oauthInterval);
                options.callback();
            }
        }, 1000);
    };
})(jQuery);
//Funcion para cargar un php sin parametros
function cargarSinParametros(div,pagina) {
    $('#contenido').hide();
    $('#cargando').show();
    $('#'+div).load(pagina, function(){
        $('#cargando').hide();
        $('#contenido').fadeIn('slow');
    });
}
//Funcion para twitear
function twitear(action,div,pagina,comment) {
    if(comment.elements[0].value != '') {
        $('#contenido').hide();
        $('#cargando').show();
        $('#'+div).load(pagina, {
            tweet:comment.elements[0].value,
            action:action
        }, function(){
            $('#'+div).load('time_line.php', function(){
                $('#cargando').hide();
                $('#form')[0].reset();
                $('#contenido').fadeIn('slow');
            });
        });
    }
}
//Funcion para borrar un tweet
function borrarTweet(action,pagina,id) {
    $('#tweet_'+id).hide();
    $('#void').load(pagina, {

```

```

        action:action,
        id:id
    });
}
//Funcion para retwitear
function retweet(action,pagina,id){
    $("#void").load(pagina, {
        action:action,
        id:id
    });
}

```

13.4.2.7 functions.php

```

<?php

//Función para formatear las URL
function txt2link($txt) {

    $patrones = array("/(http:\/\/\/[^\s]+)/i");
    $reemplazo = array("<a href=\"\1\" target=\"_blank\">\1</a>");

    return preg_replace($patrones, $reemplazo, $txt);
}

//Función para parsear los tweets
function formatTweet($text) {
    // dar formato a URLs
    $text = preg_replace('/(https?:\/\//\S+)/', '<a href="\1" target="_blank">\1</a>',
    $text);

    // dar formato a las menciones
    $text = preg_replace('/(^|\s)@(\w+)/', '\1<a href="http://twitter.com/\2"
    target="_blank">\2</a>', $text);

    // dar formato a hastags
    $text = preg_replace('/(^|\s)#(\w+)/', '\1<a
    href="http://search.twitter.com/search?q=%23\2" target="_blank">\2</a>', $text);

    return $text;
}
?>

```

13.4.3 Paquete Youtube

13.4.3.1 index.php

```

<!doctype html>
<html>

    <head>
        <title>Flypper-Youtube</title>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
        <script
        src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>
        <link href="youtube.css" rel="stylesheet" type="text/css" />
        <script type='text/javascript' src='ajax.js'></script>
    </head>
    <body>
        <div class="content">
            <form name="form"
            action="javascript:cargarConParametros('contenido','search.php',document.form);">
                <input type="text" id="buscar" name="buscar" placeholder="¿Qué quieres
                ver?">
                <input type="submit" value="Buscar">
            </form>
            <div class="menu">

```

```

        <a href="#"
onclick="cargarSinParametros('contenido','featured.php');">Recomendados</a>
        <a href="#"
onclick="cargarSinParametros('contenido','popular.php');">Más vistos</a>
        <a href="#"
onclick="cargarSinParametros('contenido','trending.php');">Trending</a>
    </div>
    <div class="cargando" id="cargando">
        
    </div>
    <div id="contenido"></div>
</div>

```

13.4.3.2 featured.php

```

<?php
include_once 'functions.php';
include_once 'xml2array.php';

$curl =
curl_init('http://gdata.youtube.com/feeds/api/standardfeeds/ES/recently_featured');
curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
curl_setopt($curl, CURLOPT_HEADER, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
$xml = curl_exec($curl);
curl_close($curl);
//Paso el xml a array
$xml = xml2array($xml);
?>
<h2>Videos recomendados</h2>

<?php
for ($i = 0; $i < 15; $i++) {
    //Obtengo los datos en xml de cada uno de los videos
    $curl = curl_init($xml['feed']['entry'][$i]['id']);
    curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    $vxml = curl_exec($curl);
    curl_close($curl);
    //Paso el xml a array
    $vxml = xml2array($vxml);

    //Atributos
    $id = obtenerID($vxml['entry']['id']);
    $thumbnail = $vxml['entry']['media:group']['media:thumbnail']['1_attr']['url'];
    $title = $vxml['entry']['title'];
    $duration = $vxml['entry']['media:group']['yt:duration_attr']['seconds'];
    //Muestro los videos
    ?>
    <div class="video" onclick="cargarVideo('play.php','<? echo $id; ?>')">
        <h3><? echo $title; ?></h3>
        <p>
            " alt="<? echo
$title; ?>" />
            Duración: <? secs2time($duration); ?>
        </p>
    </div>
    <div class="cargando" id='cargando<? echo $id; ?>'>
        <img src='http://flypper.es/flypper/api/fb/img/cargando.gif'>
    </div>
    <div class='publication' id='publication<? echo $id; ?>'></div>
<? }
?>

</body>
</html>

```

13.4.3.3 popular.php

```

<?php
include_once 'functions.php';
include_once 'xml2array.php';
//Obtengo una lista de videos
$curl = curl_init('http://gdata.youtube.com/feeds/api/standardfeeds/ES/most_viewed');
curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
curl_setopt($curl, CURLOPT_HEADER, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
$xml = curl_exec($curl);
curl_close($curl);
//Paso el xml a array
$xml = xml2array($xml);
?>
<h2>Videos más vistos en España</h2>

<?php
for ($i = 0; $i < 15; $i++) {
    //Obtengo los datos de cada video
    $curl = curl_init($xml['feed']['entry'][$i]['id']);
    curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    $vxml = curl_exec($curl);
    curl_close($curl);
    //Paso el xml a array
    $vxml = xml2array($vxml);

    //Atributos
    $id = obtenerID($vxml['entry']['id']);
    $thumbnail = $vxml['entry']['media:group']['media:thumbnail']['_attr']['url'];
    $title = $vxml['entry']['title'];
    $duration = $vxml['entry']['media:group']['yt:duration_attr']['seconds'];
    //Muestro los videos
    ?>
    <div class="video" onclick="cargarVideo('play.php','<? echo $id; ?>')">
        <h3><? echo $title; ?></h3>
        <p>
            " alt="<? echo
$title; ?>" />
            Duración: <? segs2time($duration); ?>
        </p>
    </div>
    <div class="cargando" id='cargando<? echo $id; ?>'>
        <img src='http://flypper.es/flypper/api/fb/img/cargando.gif' alt='Cargando'>
    </div>
    <div class='publication' id='publication<? echo $id; ?>'></div>
<? }
?>

</body>
</html>

```

13.4.3.4 trending.php

```

<?php
include_once 'functions.php';
include_once 'xml2array.php';
//Obtengo una lista de videos
$curl = curl_init('http://gdata.youtube.com/feeds/api/standardfeeds/ES/on_the_web');
curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
curl_setopt($curl, CURLOPT_HEADER, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
$xml = curl_exec($curl);
curl_close($curl);
//Paso el xml a array
$xml = xml2array($xml);
?>
<h2>Trending Videos</h2>

```

```

<?php
for ($i = 0; $i < 15; $i++) {
    //Obtengo los datos para cada video
    $curl = curl_init($xml['feed']['entry'][$i]['id']);
    curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    $vxml = curl_exec($curl);
    curl_close($curl);
    //Paso el xml a array
    $vxml = xml2array($vxml);

    //Atributos
    $id = obtenerID($vxml['entry']['id']);
    $thumbnail = $vxml['entry']['media:group']['media:thumbnail']['1_attr']['url'];
    $title = $vxml['entry']['title'];
    $duration = $vxml['entry']['media:group']['yt:duration_attr']['seconds'];
    //Muestro el video
    ?>
    <div class="video" onclick="cargarVideo('play.php','<? echo $id; ?>')">
        <h3><? echo $title; ?></h3>
        <p>
            " alt="<? echo
            $title; ?>" />
            Duración: <? secs2time($duration); ?>
        </p>
    </div>
    <div class="cargando" id='cargando<? echo $id; ?>' alt="cargando">
        <img src='http://flypper.es/flypper/api/fb/img/cargando.gif'>
    </div>
    <div class='publication' id='publication<? echo $id; ?>'></div>
    <? }
    ?>
</body>
</html>

```

13.4.3.5 search.php

```

<?php
include_once 'functions.php';
include_once 'xml2array.php';
//Obtengo una lista de videos
$buscar = str_replace(" ", "+", $_POST['buscar']);
$curl = curl_init('http://gdata.youtube.com/feeds/api/videos?vq=' . $buscar .
'&orderby=relevance&start-index=1&max-results=15');
curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
curl_setopt($curl, CURLOPT_HEADER, false);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
$xml = curl_exec($curl);
curl_close($curl);
//Paso el xml a array
$xml = xml2array($xml);

for ($i = 0; $i < count($xml['feed']['entry']); $i++) {
    //Obtengo los datos de cada video
    $curl = curl_init($xml['feed']['entry'][$i]['id']);
    curl_setopt($curl, CURLOPT_HTTPHEADER, 'GET');
    curl_setopt($curl, CURLOPT_HEADER, false);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    $vxml = curl_exec($curl);
    curl_close($curl);
    //Paso el xml a array
    $vxml = xml2array($vxml);
    //Atributos
    $id = obtenerID($vxml['entry']['id']);
    $thumbnail = $vxml['entry']['media:group']['media:thumbnail']['1_attr']['url'];
    $title = $vxml['entry']['title'];
    $duration = $vxml['entry']['media:group']['yt:duration_attr']['seconds'];
}

```

```

//Muestro el video
?>
<div class="video" onclick="cargarVideo('play.php','<? echo $id; ?>')">
  <h3><? echo $title; ?></h3>
  <p>
    " alt="<? echo
$title; ?>" />
    Duración: <? segs2time($duration); ?>
  </p>
</div>
<div class="cargando" id='cargando<? echo $id; ?>' alt="cargando">
  <img src='http://flypper.es/flypper/api/fb/img/cargando.gif'>
</div>
<div class='publication' id='publication<? echo $id; ?>'></div>
<?
}
//Error
if (count($xml['feed']['entry']) == 0) {
  ?>
  <div class="error">
    La búsqueda no tiene resultados.
  </div>
  <?
}
?>

```

13.4.3.6 play.php

```

<?php
include 'functions.php';
$id = $_POST['id'];
$embed = "http://www.youtube.com/embed/" . $id;
?>
<div class="slide">
  <iframe name="Reproductor video" width="278" height="200" src=<?php echo $embed; ?>
  frameborder="0" allowfullscreen></iframe>
  <a class="cerrar" onclick="cerrarVideo('<? echo $id; ?>')">Cerrar</a>
</div>

```

13.4.3.7 ajax.js

```

//Función para cargar un php sin parámetros
function cargarSinParametros(div,pagina) {
  $('#contenido').hide();
  $('#cargando').show();
  $("#" + div).load(pagina, function() {
    $('#cargando').hide();
    $('#contenido').fadeIn('slow');
  });
}

//Función para cargar un php con parámetros
function cargarConParametros(div,pagina,comment) {
  if(comment.elements[0].value != ''){
    $('#contenido').hide();
    $('#cargando').show();
    $("#" + div).load(pagina, {
      buscar:comment.elements[0].value
    }, function() {
      $('#cargando').hide();
      $('#contenido').fadeIn('slow');
    });
  }
}

//Función para cargar un video
function cargarVideo(pagina, id_video){
  if ($("#publication"+id_video).is(":hidden")) {
    $('#cargando'+id_video).show();
    $("#" + id_video).load(pagina, {
      id:id_video
    }, function() {

```

```

        $("#cargando"+id_video).hide();
        $("#publication"+id_video).slideDown("slow");
    });
}
else{
    $("#publication"+id_video).slideUp();
}
}
//Función para cerrar un video
function cerrarVideo(id){
    $("#publication"+id).empty();
}

```

13.4.3.8 functions.php

```

<?php

//URLs del tipo http://gdata.youtube.com/feeds/api/videos/kffacxfA7G4
function obtenerID($url) {
    $pieces = explode("/", $url);
    return $pieces[6];
}

//Pasa el tiempo dado en segundos a minutos y segundos
function segs2time($tiempo) {
    $minutos = number_format($tiempo / 60, 0);
    $segundos = $tiempo % 60;
    echo $minutos . " " . $segundos . " ";
}

?>

```

13.4.4 Paquete Spotify

13.4.4.1 index.php

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
    <head>
        <title>Flypper - Spotify</title>
        <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
        <!-- Javascript botón tweet -->
        <script src="http://platform.twitter.com/widgets.js"
type="text/javascript"></script>
        <script
src="http://ajax.googleapis.com/ajax/libs/jquery/1.4.2/jquery.min.js"></script>
        <link href="spotify.css" rel="stylesheet" type="text/css" />
        <script type='text/javascript' src='ajax.js'></script>
    </head>
    <body style="background-color:#fff;">
        <div class="content">
            <form name="form"
action="javascript:cargarConParametros('contenido','search.php',document.form);">
                <input type="text" id="buscar" name="buscar" placeholder="¿Qué quieres
oir?">
                <input type="submit" value="Buscar">
            </form>
            <div class="cargando" id="cargando">
                
            </div>
            <div id="contenido"></div>
        </div>

```

13.4.4.2 search.php

```

<?php
//Sustituyo los espacios por + para realizar la búsqueda
$buscar = str_replace(" ", "+", $_POST['buscar']);
//Ejecuto la búsqueda mediante curl
$curl = curl_init('http://ws.spotify.com/search/1/track?q=' . $buscar);
curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
$xml = curl_exec($curl);
curl_close($curl);
//Paso el xml a array
$xml = simplexml_load_string($xml);

//Obtengo los datos recibidos
for ($i = 0; $i < count($xml->track); $i++) {
    //Nombre
    $song = $xml->track[$i]->name;
    //Url
    $song_id = $xml->track[$i]->attributes()->href;
    $id = str_replace("spotify:track:", "", $song_id);
    $song_uri = "http://open.spotify.com/" . str_replace("spotify:track:", "track/",
    $song_id);
    //Artista
    $artist = $xml->track[$i]->artist->name;
    $artist_uri = $xml->track[$i]->artist->attributes()->href;
    $artist_uri = "http://open.spotify.com/" . str_replace("spotify:artist:", "artist/",
    $artist_uri);
    //Album
    $album = $xml->track[$i]->album->name;
    $album_uri = $xml->track[$i]->album->attributes()->href;
    $album_uri = "http://open.spotify.com/" . str_replace("spotify:album:", "album/",
    $album_uri);
    //Muestro las canciones
    ?>
    <div class="song" onclick="cargarCancion('play.php','<? echo $song; ?>','<? echo
    $id; ?>','<? echo $song_uri; ?>','<? echo $artist; ?>','<? echo $song_id; ?>')">
        <a >
            <p id="artist"><? echo $artist; ?></p>
            <p id="song"><? echo $song; ?></p>
            <p id="album"><? echo $album; ?></p>
        </a>
    </div>
    <div class='cargando' id='cargando<? echo $id; ?>'>
        <img src='http://flypper.es/flypper/api/fb/img/cargando.gif'>
    </div>
    <div class='publication' id='publication<? echo $id; ?>'>

</div>
<?
}
if (count($xml['feed']['entry']) == 0) {
    ?>
    <div class="error">
        La búsqueda no tiene resultados.
    </div>
    <?
}
?>

```

13.4.4.3 play.php

```

<div class="slide">
    <!-- Incluyo el reproductor de Spotify-->
    <iframe src="https://embed.spotify.com/?uri=<? echo $_POST['song_id']; ?>"
    width="278" height="350" frameborder="0" allowtransparency="true"></iframe>
    <!-- Compartir en Twitter-->
    <script>!function(d,s,id){var
    js,fjs=d.getElementsByTagName(s)[0];if(!d.getElementById(id)){js=d.createElement(s);js.i
    d=id;js.src="//platform.twitter.com/widgets.js";fjs.parentNode.insertBefore(js,fjs);}(d
    oment,"script","twitter-wjs");</script>
    <div>
        <a href="http://twitter.com/share" class="twitter-share-button"

```

```

data-url="<? echo $_POST['song_uri']; ?>"
data-via="flypper_es"
data-text="Compartiendo la canción <? echo $_POST['song']; ?> de <? echo
$artist; ?> desde Flypper"
data-related="flypper_es"
data-lang="es"
data-size="large">Tweet</a>
</div>
<!-- Compartir en Facebook-->
<iframe src="https://www.facebook.com/plugins/like.php?href= <? echo
$_POST['song_uri']; ?>" scrolling="no" frameborder="0" style="border:none; width:260px;
height: 30px">
</iframe>
<div class="cerrar" onclick="cerrarCancion('<? echo $_POST['id']; ?>')">
    Cerrar
</div>
</div>

```

13.4.4.4 ajax.js

//Función para cargar un php sin parámetros

```

function cargarSinParametros(div,pagina){
    $('#contenido').hide();
    $('#cargando').show();
    $('#"+div).load(pagina, function(){
        $('#cargando').hide();
        $('#contenido').fadeIn('slow');
    });
}
//Función para cargar un php con parámetros
function cargarConParametros(div,pagina,comment){
    if(comment.elements[0].value != ''){
        $('#contenido').hide();
        $('#cargando').show();
        $('#"+div).load(pagina, {
            buscar:comment.elements[0].value
        }, function(){
            $('#cargando').hide();
            $('#contenido').fadeIn('slow');
        });
    }
}
//Función para cargar la canción
function cargarCancion(pagina,cancion,id,uri,artista,track){
    if ($('#publication"+id).is(":hidden")) {
        $('#cargando"+id).show();
        $('#publication"+id).load(pagina, {
            song:cancion,
            song_id:track,
            song_uri:uri,
            artist:artista,
            id:id
        }, function(){
            $('#cargando"+id).hide();
            $('#publication"+id).slideDown("slow");
        });
    }
    else{
        $('#publication"+id).slideUp();
    }
}
//Función para cerrar una canción
function cerrarCancion(id){
    $('#publication"+id).empty();
}

```