



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMÁTICA

INGENIERO TÉCNICO EN INFORMÁTICA DE GESTIÓN

GESTION DE LA API DE TWITTER.COM

Realizado por

Rafael P. Ruiz López

Dirigido por

José Ramón Portillo Fernández

Departamento

MATEMÁTICA APLICADA I

Sevilla, Junio 2008

Índice

1. INTRODUCCIÓN.....	1
1. Definición de objetivos.....	2
2. Historia.....	2
3. Tecnología.....	4
4. Twitter.....	5
2. ANÁLISIS DE ANTECEDENTES Y APORTACIÓN REALIZADA	11
3. ANÁLISIS TEMPORAL.....	15
4. HERRAMIENTAS DE IMPLEMENTACIÓN.....	17
5. ANÁLISIS DE REQUISITOS, DISEÑO E IMPLEMENTACIÓN.....	21
5.1 Análisis de requisitos.....	21
5.2 Diseño.....	23
5.2.1 Subsistema de Aplicación Principal.....	25
5.2.2 Subsistema de Ampliaciones.....	27
5.2.3 Subsistema de Presentación.....	28
5.3 Implementación.....	28
5.3.1 Subsistema de aplicación principal.....	28
5.3.1.1 twitter.class.php.....	28
5.3.1.2 Db_functions.php.....	29
5.3.1.3 Utilidades.php.....	30
5.3.1.4.Utiles_fechas.php.....	31
5.3.1.5 Validacion.php.....	31
5.3.1.6 Devdiv.php.....	32
5.3.2 Subsistema de Ampliaciones.....	32
5.3.2.1 eti_functions.php.....	32
5.3.2.2 Devdivtng.php.....	32
5.3.2.3 Posttag.php.....	32

5.3.2.4 Etimain.php.....	32
5.3.3 Subsistema de presentación.....	33
5.3.3.1 funciones.js.....	33
5.3.3.2 Scripts Generadores HTML.....	34
6. MANUAL DE USUARIO.....	37
7. PRUEBAS.....	40
8. COMPARACIÓN CON OTRAS ALTERNATIVAS.....	43
9. CONCLUSIONES.....	46
10. BIBLIOGRAFÍA.....	48

Capítulo 1: Introducción

Con este proyecto se pretende realizar una aplicación Web que mejore y amplíe los servicios proporcionados por la página Web www.twitter.com. Dicha página es un sistema muy simple de “microbloggin” que te permite compartir con otros usuarios mensajes de una longitud máxima de 140 caracteres de longitud, y que además pone a disposición de los usuarios una API (del inglés Application Programming Interface - Interfaz de Programación de Aplicaciones).

La aplicación consistirá en una interfaz que implemente de un modo mas atractivo para el usuario las funcionalidades de la página Web www.twitter.com, y además ofrezca otras funcionalidades que actualmente el servicio en cuestión no ofrece.

La primera parte del proyecto se centró en la investigación sobre como funcionaban las distintas tecnologías de las utilizadas en el proyecto como son PHP, curl, MySQL, AJAX, y la propia API provista por www.twitter.com. Y además, analizar que funciones de las que ya realizaba www.twitter.com, podría realizar la aplicación y que otras funciones o mejoras podrían implementarse sobre la API de dicha

página web.

La parte de implementación del proyecto comenzó realizando una base sobre lo que poder trabajar, como son un sistema de registro de usuarios, un sistema de login y sesiones. Más adelante se empezó a trabajar con los servicios que ofrecía www.twitter.com, estableciendo unos patrones y procesos en el lenguaje PHP para implementarlos. Se continuó trabajando con AJAX y Javascript, para darle forma a la presentación de los datos en la página web. Posteriormente se genero un motor para etiquetas, y se implemento sobre la aplicación. Y para terminar se ha realizado una amplia corrección de errores y mejoras en la interfaz.

1.1 Definición de objetivos

- Desarrollar un proyecto original, ya que no existen aplicaciones de años anteriores de este tipo.
- Profundizar en el conocimiento del entorno de programación PHP-MySQL, y en el lenguaje Javascript.
- Investigar y poner en práctica la tecnología Web AJAX.
- Desarrollar una aplicación Web en el lenguaje PHP.
- Estudiar el funcionamiento de la tecnología AJAX/Javascript.
- Conseguir que la aplicación Web desarrollada haga uso de la tecnología AJAX para lograr una interfaz atractiva para el usuario.
- Ampliar los servicios ofrecidos por www.twitter.com, con un sistema de etiquetas.

1.2 Historia

Desde sus inicios hasta la actualidad, la Web ha ido evolucionando. Hoy, nos encontramos en una etapa de esta evolución marcada por servicios y características particulares que requieren nuevas técnicas de desarrollo de aplicaciones Web y también nuevas tecnologías (o, en algunos casos, nuevos usos de tecnologías existentes).

En sus orígenes, la Web era utilizada por los navegantes para obtener información. La unidad de trabajo era la página Web y, de esta forma, los usuarios navegaban entre múltiples páginas a través de hipervínculos. En esta conceptualización, el elemento central es el servidor Web con su información. Generalmente, estos sitios consistían en sitios corporativos, sitios de noticias, sitios de información específica.

El siguiente paso en esta evolución consistió en utilizar la red como comunidades, en las que los propios navegantes (o usuarios de la Web) pudieran interactuar, ya sea entre ellos o

con otros sitios que ofrecían servicios. Durante esta fase aparecieron servicios tales como foros de discusión, servicios de e-banking, servicios de mensajería, compras electrónicas basadas en comunidades, etc.

El paso actual, es la visión de la Web como una plataforma, es decir, como el medio para la distribución de servicios y para la construcción de información, su selección, etc. También, detrás de esta nueva visión, surgen nuevos modelos de negocio basados en la web como plataforma, y además de modelos de negocio, también surgen nuevas formas de comunicarse sobre esa plataforma como son los

“blogs”, y posteriormente el “microbloggin”.

Un blog, o en español también una bitácora, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente. Habitualmente, en cada artículo de un blog, los lectores pueden escribir sus comentarios y el autor darles respuesta, de forma que es posible establecer un diálogo.

El micro-blogging, también conocido como nano-blogging, es un servicio que permite a sus usuarios enviar mensajes de sólo texto, con una longitud máxima de 140 caracteres, vía SMS, mensajería instantánea, sitios web de o aplicaciones ad hoc.

Estas actualizaciones se muestran en la página de perfil del usuario, y son también enviadas de forma inmediata a otros usuarios que han elegido la opción de recibirlas. El usuario origen puede restringir el envío de estos mensajes sólo a miembros de su círculo de amigos, o permitir su acceso a todos los usuarios, que es la opción por defecto.

1.3 Tecnología

Una de las principales tecnologías utilizadas en este proyecto es AJAX. AJAX es una combinación de cuatro tecnologías ya existentes:

- 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 asincrónicamente 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 comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.

Por otra parte también se han utilizado otros lenguajes de programación, de almacenamiento de datos, y de publicación como son PHP, MySQL, y Apache:

- PHP es un lenguaje de programación interpretado usado normalmente para la creación de páginas web dinámicas. PHP es un acrónimo recursivo que significa "PHP Hypertext Pre-processor" (inicialmente PHP Tools, o, Personal Home Page Tools).

- MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario.

- El servidor HTTP Apache es un software (libre) servidor HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio

virtual.

Además, se utilizan dos herramientas adicionales e imprescindibles como son el modulo para apache de PHP y la librería curl para PHP.

- El modulo para apache "mod_php" amplia las posibilidades del servidor web apache, y hace que muestre páginas dinámicas programas en PHP.

- 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. Libcurl es la librería/API correspondiente que los usuarios pueden incorporar en sus programas; cURL actúa como un wrapper aislado para la librería libcurl. La librería libcurl se puede usar desde más de 30 lenguajes distintos.

1.4 Twitter

Twitter es una red social libre y gratuita, y un servicio de micro-blogging que permite a los usuarios enviar mensajes de estado (o "tweets", mensajes de texto de una longitud máxima de 140 caracteres) a la página de Twitter, a través de su propia página web, un servicio de mensajería instantánea, o bien a partir de aplicaciones creadas por terceros como Twittrific.

Las actualizaciones se muestran en la página personal del usuario, y son enviadas instantáneamente a otros usuarios que han

decidido recibirlas. El usuario puede enviar mensajes a una persona en privado, solo enviar mensajes a sus amigos, o bien mandar los mensajes en general, que es la opción por defecto. Los usuarios pueden recibir las actualizaciones de los estados través de la página de twitter, mensajería instantánea, SMS, RSS, email, o a través de aplicaciones de terceros.

Twitter empezó siendo un proyecto de investigación y desarrollo dentro de una compañía start-up llamada Obvious en Marzo de 2006. Inicialmente fue utilizada internamente por la compañía, y fue oficialmente lanzado en Octubre de 2006.

En Abril de 2007, Obvious LLC separo el servicio, creando una entidad independiente llamada Twitter, Inc con Jack Dorsey como CEO.

Twitter ha sido utilizado por muchas organizaciones como bomberos o policía para coordinar eventos. Algunos de sus usuarios más importantes son los candidatos a presidentes de gobierno de los estados unidos Ron Paul, John Edwards, Barack Obama, y Hillary Clinton.

Por otra parte algunas universidades están utilizando en servicio para comunicarse con sus alumnos de manera instantánea.

Debido al éxito de twitter han surgido muchos servicios que imitan el concepto pero añaden características específicas del un país concreto, o combinan el micro-bloggin con otros servicios como el intercambio de archivos.

En Mayo de 2007 se contaban al menos 111 clones de twitter internacionales.

Twitter esta escrito en Ruby on Rails. Twitter alcanzo aproximadamente un 98% de tiempo online en 2007, es decir, alrededor de 6 días completos de caída. Los momentos en que Twitter no esta disponible son particularmente comunes durante eventos de la industria de la tecnología.

Capítulo 2: Análisis de antecedentes y aportación realizada

Como ya se ha descrito anteriormente, el objetivo de nuestra aplicación era una mejora de la interfaz y de las posibilidades de la aplicación servida por Twitter usando la API que tienen disponible.

Existen multitud de programas que hacen uso del API de Twitter para mejorar la interfaz y la forma de presentar los datos del propio Twitter. La gran mayoría de ellos están programados para la nueva plataforma lanzada por adobe, Adobe Air, aunque existen otros para otras plataformas como Flash. Estas aplicaciones son de código cerrado, además de que utilizan un lenguaje de programación propio de la plataforma.

Para la web, no existe ninguna alternativa actualmente que muestre los datos de Twitter de una forma atractiva. Una de las características principales de Web actual (conocida con el sobrenombre de Web 2.0) es el uso de “feeds” o archivos que contienen información actualizada. Estos archivos suelen tener extensión XML o RSS,

dependiendo del formato de orden interno que posean, además suelen carecer de estilos. Twitter hace uso de este tipo de archivos en su API y es a través de ellos de los que se extrae la información actualizada. No existe ninguna aplicación web que haga uso de este sistema para presentar los datos de Twitter, aunque si he encontrado otras aplicaciones que hacen uso de esta tecnología para presentar los datos de forma atractiva:

- [LiveMarks](#): LiveMarks es un servicio para mostrar del.icio.us y otros sistemas de marcadores online, en vivo.
- [Digg](#), y clones como [Meneame.net](#): Estos sistemas de promoción social de noticias poseen una aplicación llamada “sneak”, o en la versión traducida “fisgona” que muestra los eventos en la página web, como votos o envíos de noticias en vivo.

Sin embargo si hacemos un breve análisis a ambas opciones comprobaremos que ninguna de las dos opciones nos es valida. LiveMarks es de código cerrado, es decir, no se tiene acceso al código fuente. Además tras un pequeño estudio de su comportamiento, se puede comprobar que realmente no muestra información inmediata, es decir, descarga gran cantidad de información del servidor de del.icio.us, y luego la va mostrando poco a poco, imitando que se estuviera generando instantáneamente esa información. La “fisgona” de Meneame.net si es de código abierto, sin embargo tampoco me resulto útil, debido a la complejidad del código y a su falta de explicación. Además, la fisgona y nuestra aplicación tratan fuentes de datos muy diferentes, ya que la fisgona cubre diferentes acciones, mientras que nuestra aplicación solo debe cubrir la presentación de mensajes de texto.

Por otra parte, debido al éxito de Twitter, cada día surgen nuevas alternativas al original, con diferentes características y opciones. Jaiku y Pownce son las aplicaciones mejor situadas:

- Jaiku: Jaiku es la competencia más cercana a [twitter](#). Además de enviar mensajes jaiku te permite compartir otro tipo de información sobre ti, como si acabas de subir fotos a flickr o guardar un favorito en del.icio.us. [Google](#) adquirió este producto, para ofrecerlo a sus usuarios como parte de su plataforma de servicios lo que supone una muy fuerte amenaza a [twitter](#).
- Pownce: es un servicio similar a Twitter, que permite a los usuarios crear redes de amigos y compartir con ellos, algunos de ellos o a cualquiera, mensajes de texto, archivos, enlaces e invitaciones a eventos.
- Recientemente se ha lanzado una aplicación de software libre en España que esta creciendo muy rápidamente. Jisko permite grupos, posee un sistema de relevancia según karma y otras muchas opciones.

Aunque también existen otras alternativas dignas de mención por diferentes motivos:

[Sweetter](#): es un servicio de microbloggin muy similar a twitter con la diferencia de que los usuarios poseen un propiedad llamada "karma", que mide la popularidad del mismo. Este proyecto esta siendo desarrollado por SUGUS (grupo de usuarios de GNU de la Universidad de Sevilla). Es Open Source.

Aunque estas aplicaciones pueden parecer técnicamente

superiores, de momento Twitter es la aplicación con más tráfico de todas ellas, y además la que mas interés despierta en Google.

Gráfico de volumen de búsquedas en Google:

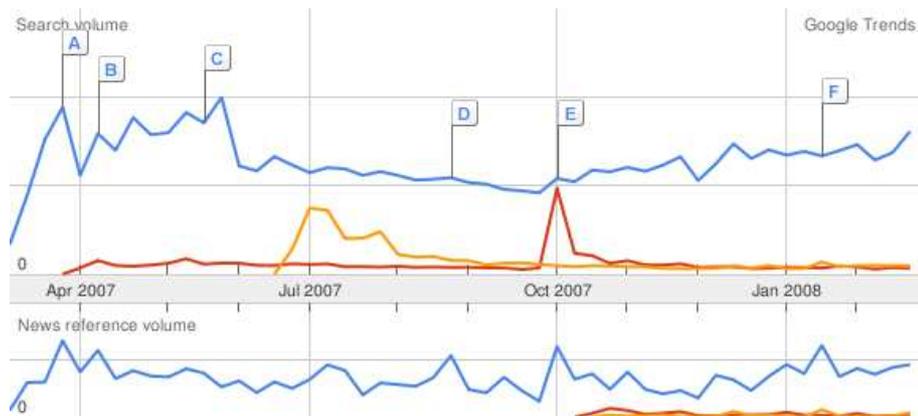
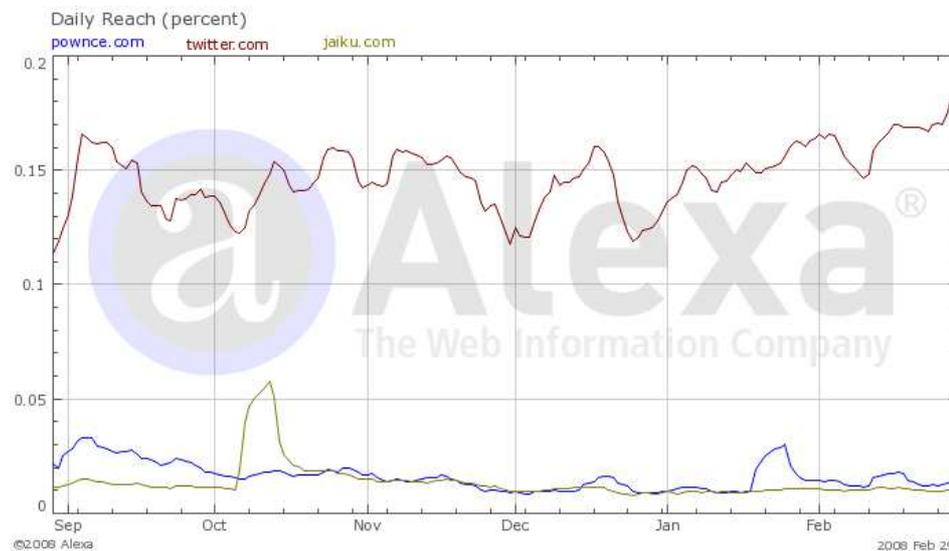


Gráfico de volumen de tráfico de Alexa:



Capítulo 3: Análisis Temporal

En este capítulo voy a mostrar el tiempo que he dedicado a la realización del proyecto. En realidad estos tiempos deben ser tomados como una aproximación ya que nunca se dedica el 100% del tiempo a una sola tarea, es decir, en algunos momentos, durante la programación de la aplicación, necesitas consultar más información, o cuando estas informándote, recurres a cambiar algo en el código.

El desarrollo de este proyecto ha sido realizado durante todo el curso, dedicándole más tiempo en aquellos momentos en los que no tenía exámenes.

En un principio estude cuales eran las mejores opciones para realizar este proyecto, debido a que tenia unos ligeros conocimientos en PHP, me incline por esta opción. Gran parte del tiempo, se empleo en familiarizarme con el lenguaje de programación "javascript" ya que era la primera vez que entraba en contacto con el. También era la primera vez que entraba en contacto con el desarrollo de aplicaciones que hicieran uso de la tecnología AJAX y también estude cual era el mejor "Framework" para esta aplicación el particular entre los muchos

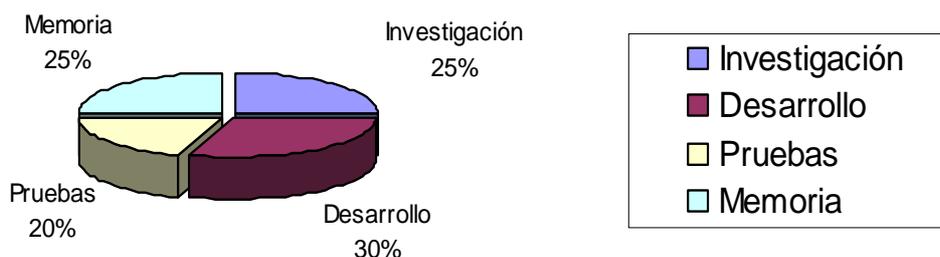
disponibles: Prototype, JQuery, MooTools, etc.

Otro de los puntos que también requirió bastante tiempo fue el estudio del funcionamiento del API de twitter, ya que nunca había trabajado con un API. Por otra parte, también desconocía que era curl y tuve que familiarizarme con el funcionamiento del conjunto php/curl y el API de twitter.

La redacción de la documentación del proyecto también ha requerido gran cantidad de tiempo, y recordar muchas cosas que ya tenía casi olvidadas del proyecto, como la estructura de los directorios o el funcionamiento de algunas funciones. Esto se produjo sobre todo por que se comenzó la redacción justo después de los exámenes de febrero, y por tanto el periodo en el que le dediqué algo menos de tiempo.

Ahora veremos los tiempos dedicados en la realización del proyecto:

Tiempo de investigación	Tiempo de desarrollo	Tiempo de pruebas	Tiempo de memoria	Tiempo total
75h	90h	60h	75h	300h



Capítulo 4: Herramientas de implementación

Para el desarrollo e implementación de esta aplicación hemos usado en su mayoría software gratuito, libre y de libre distribución. Como lenguaje principal de programación he elegido PHP ya que es uno de los lenguajes de programación web dinámica más populares actualmente. El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

Con las primeras 2 versiones de PHP, PHP 3 y PHP 4, se había conseguido una plataforma potente y estable para la programación de páginas del lado del servidor. Sin embargo, todavía existían puntos negros en el desarrollo PHP que se han tratado de solucionar con la versión 5, aspectos que se echaron en falta en la versión 4, casi desde el

día de su lanzamiento. Nos referimos principalmente a la programación orientada a objetos (POO) que, a pesar de que estaba soportada a partir de PHP3, sólo implementaba una parte muy pequeña de las características de este tipo de programación.

Estos motivos son los que nos han hecho inclinarnos por la versión 5 de PHP para nuestro proyecto.

Por otra parte, hemos necesitado habilitar el modulo para PHP curl. Curl es una aplicación para la línea de comandos para la transferencia de archivos con sintaxis URL. El modulo para PHP homónimo, permite gestionar las transferencias de archivos directamente desde PHP. Curl permite gestionar gran cantidad de protocolos y será el encargado de obtener los feeds XML del servidor de twitter.

Para el almacenamiento de datos, hemos elegido el sistema de gestión de bases de datos MySQL Server. El propósito general de los sistemas de gestión de base de datos es el de manejar de manera clara, sencilla y ordenada un conjunto de datos que posteriormente se convertirán en información relevante. Existen diferentes alternativas como sistema de gestión de bases de datos, algunas libres como MySQL, PostgreSQL, o SQLite. Otras no libres, aunque si gratuitas como Microsoft SQL Server Compact Edition, y también existen opciones de pago como Oracle, o IBM DB2.

Descartadas las opciones de pago, MySQL es el servidor de bases de datos más popular y que más aceptación tiene ahora mismo. Por otra parte, gracias a las facilidades del lenguaje SQL y a la abstracción de la capa de datos realizada durante el proyecto seria factible cambiar el servidor de bases de datos por uno que se ajuste más a las necesidades del servidor donde se aloje la aplicación.

El IDE (Entorno de desarrollo integrado) elegido para el desarrollo del proyecto fue una “distribución” del famoso entorno de desarrollo Eclipse. Esta “distribución”, llamada EasyEclipse for PHP, es libre y gratuita, y además contiene todo lo necesario para la programación en el entorno de HTML, PHP, MySQL, y Javascript. Los plugins ya incluidos que nos resultaron más útiles fueron:

PHP Eclipse: que permite editar, ejecutar y comprobar, aplicaciones en PHP.

Eclipse Web Tools Editors: que permite editar y validar gran cantidad de formatos de archivos, entre ellos HTML, CSS, y Javascript.

Para la presentación final de los datos en la web, teníamos múltiples opciones también.

HTML y CSS, conseguiríamos una página estática que solo se actualizaría bajo petición del usuario o el navegador, bajo algún evento concreto.

HTML, CSS, y Flash, con ayuda del actionscript podríamos generar nuestros propios eventos, ya sean temporales, o generados por el usuario o el navegador. El navegador del usuario necesita un plugin, y además habría k utilizar software comercial y de pago para obtener unos buenos resultados.

HTML, CSS y AJAX/Javascript, la solución elegida, ya que nos permite generar eventos basados en el tiempo y en acciones del usuario y el navegador. Además viene integrado por defecto en los principales navegadores, y es por esto que no necesita de ninguna herramienta específica para su desarrollo.

Aunque no son necesarias herramientas específicas para el desarrollo de aplicaciones con AJAX/Javascript existen diferentes librerías o frameworks que agilizan el desarrollo. Las mas populares son JQuery, MooTools, Prototype y Scriptaculous, aunque existen otros también muy utilizados como Mochikit, Rico, o Xajax, cada uno orientado a unos objetivos concretos, y totalmente compatibles entre si.

Para nuestra aplicación hemos usado Prototype y Scriptaculous, y sobre estos dos frameworks hemos escrito la mayoría de las funciones relacionadas con la presentación de la información de la aplicación. Prototype es un Framework basado en JavaScript orientado a proporcionar al desarrollador de técnicas AJAX listas para ser usadas. Prototype nos simplifica gran parte del trabajo cuando se pretende desarrollar páginas altamente interactivas. Script.aculo.us es una librería JavaScript que permite el uso de controles AJAX, drag'n'drop, y otros efectos visuales en una página web. Script.aculo.us se distribuye mediante descargas en varios formatos de archivo, y también está incluido en Ruby on Rails y otros frameworks de desarrollo web.

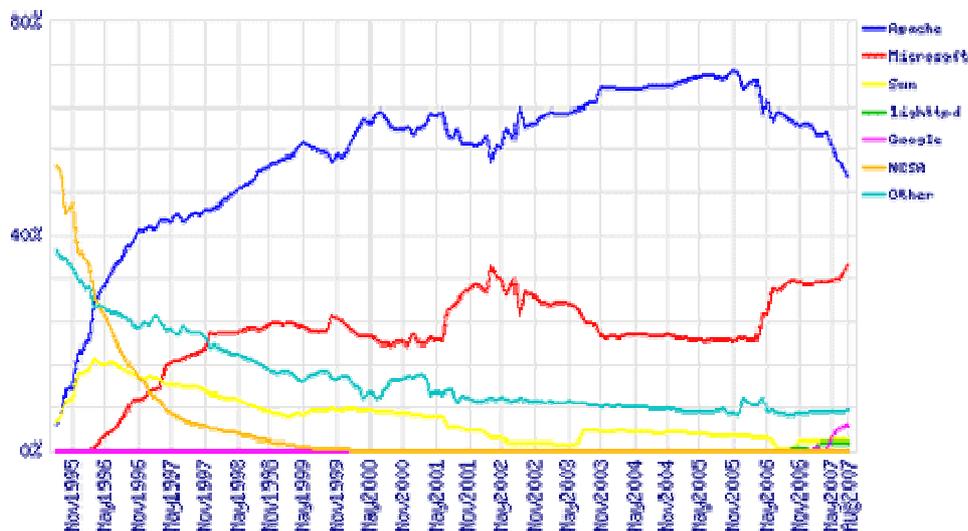
Capítulo 5: Análisis de requisitos, diseño e implementación.

5.1 Análisis de requisitos

Ya que esta aplicación esta diseñada para ser ejecutada en un entorno web, este entorno debe cumplir varios requisitos tanto del lado del cliente, como del lado del servidor. Comenzaremos del lado del servidor, que es la parte con la que tendrá mas contacto el desarrollador.

El servidor web debe ser compatible con PHP y con un servidor de bases de datos, es decir, debe poder interpretar páginas escritas en lenguaje PHP y además debe permitir que estas páginas se conecten a un servidor de bases de datos para obtener la información.

Actualmente solo dos programas de servidores web albergan más del 85% de las páginas que se sirven actualmente. Estos servidores son Apache, e IIS.



Ambos servidores son totalmente compatibles con PHP y con servidores de bases de datos, ya que PHP dispone de un modulo especifico para cada uno de ellos.

Del lado del cliente, solo será necesario que el navegador del que dispone el cliente sea compatible con Javascript, y la tecnología AJAX, además de que el propio cliente disponga de una cuenta en Twitter.

La mayor parte de los navegadores actuales implementa dichas tecnologías como son:

- Navegadores basados en Gecko como Mozilla, Mozilla Firefox, SeaMonkey, Camino, K-Meleon, Flock, Epiphany, Galeon y Netscape versión 7.1 y superiores
- Microsoft Internet Explorer para Windows versión 5.0 y superiores, y los navegadores basados en él
- Navegadores con el API KHTML versión 3.2 y superiores implementado, incluyendo Konqueror versión 3.2 y superiores,

Apple Safari versión 1.2 y superiores, y el Web Browser for S60 de Nokia tercera generación y posteriores

- Opera versión 8.0 y superiores, incluyendo Opera Mobile Browser versión 8.0 y superiores.

Por otra parte los navegadores que no implementan estas tecnologías son:

- Opera 7 y anteriores
- Microsoft Internet Explorer para Windows versión 4.0 y anteriores
- Microsoft Internet Explorer para Macintosh, todas las versiones
- Dillo
- Navegadores basados en texto como Lynx y Links
- Navegadores para incapacitados visuales (braille)

5.2 Diseño

El diseño de este proyecto se puede dividir en varias secciones claramente diferenciadas. La primera de ellas se corresponde con el funcionamiento principal de la aplicación y la comunicación con el API de twitter. La segunda parte se corresponde con la presentación de la información, programada en HTML, CSS, y Javascript. Y la última parte son las ampliaciones a la aplicación, como el sistema de etiquetas.

En la primera parte se incluyen las clases y funciones destinadas los accesos a la base de datos, y los accesos al API de twitter.

Los accesos a la base de datos fundamentalmente cubren dos de las funciones de la página. El registro, logueado, y salida de usuarios para el uso de sesiones y por otra parte, las posibles ampliaciones de funcionalidad, en este caso, el uso de etiquetas para los usuarios.

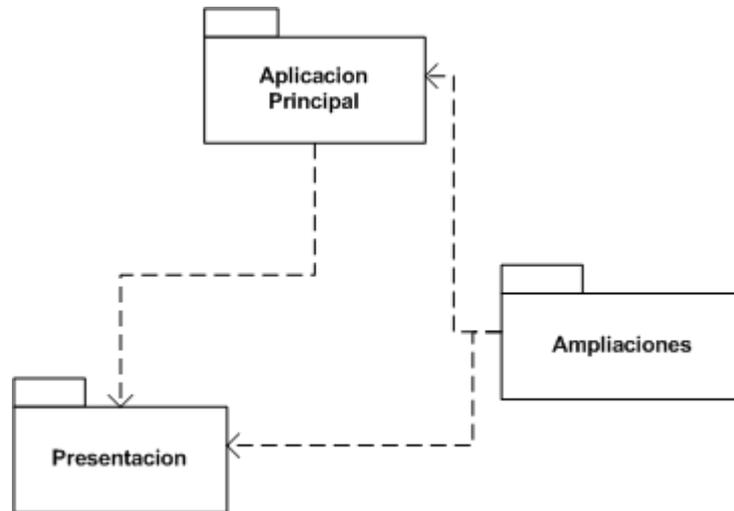
Los accesos al API de twitter, nos sirven para obtener los mensajes mas recientes, de las personas a las que sigues (friend_timeline), los mensajes de la gente en general (public_timeline), o los mensajes que van dirigidos a ti (replies). Estos accesos se ejecutan mediante funciones, que son llamadas por un script php. Obtienen el feed en formato XML mas reciente, y lo convierten al tipo array de PHP. Una vez en este formato, se obtienen los datos que se necesitan presentar con los accesos al array, y se almacenan en una variable en formato HTML.

En la segunda parte, se hace uso de Javascript y en particular de la tecnología AJAX, para ejecutar el script anterior en lenguaje PHP, mandándole los parámetros necesarios para su ejecución, y obteniendo como resultado de dicha ejecución una cadena, formateada en HTML. Esa información es directamente añadida a la página actual, aunque de forma no visible. Posteriormente, se realiza una comparación entre la información visible, y la información que esta oculta. Si difieren, se actualiza la información visible.

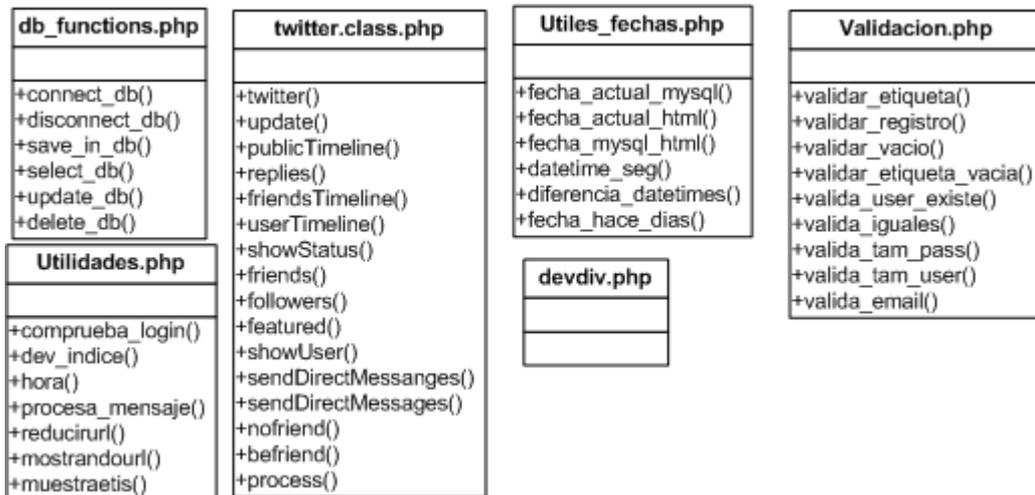
Además, se han añadido algunos efectos de la librería “scriptaculus”, para hacer la carga y la recarga automática algo más atractiva para el usuario.

En la tercera parte, se reutilizan gran parte de las clases y scripts programados anteriormente para implementar nuevas funciones. En este caso un sistema de etiquetas que filtre los mensajes de un determinado grupo de usuarios. Obviamente esta nueva función hará uso tanto del bloque primero, ya que hace accesos a la base de datos y accesos al API de twitter, además también reutilizaremos los scripts de la parte segunda para presentar los datos filtrados por las etiquetas.

Después de todas estas consideraciones, el esquema de la aplicación queda de la siguiente manera:



5.2.1 Subsistema de Aplicación Principal



El subsistema de Aplicación Principal comprende varias

Clases:

Db_functions.php contiene todas las funciones necesarias para conectarse a la base de datos, almacenar, actualizar, obtener, y eliminar los datos de la base de datos.

Utilidades.php es un conjunto de métodos específicos para solventar algunas necesidades, como el login, y el procesado de la información antes de mostrarla.

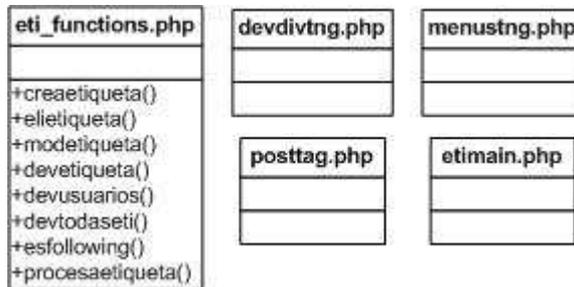
Twitter.class.php es el comprende todos los métodos que ofrece el API de Twitter para interactuar con sus servidores. Nos permite actualizar el estado, y obtener todos los feeds de información que ofrece Twitter.

Utiles_fechas.php conjunto de métodos para el tratamiento y formateado de fechas. Para pasar de formato MySQL a HTML y viceversa.

Dev_div.php es un script que recibe mediante los datos de sesión, cual es el feed de datos procedente de Twitter que se desea. Este será obtenido del servidor periódicamente. Una vez obtenido el feed analiza si hay alguna actualización con respecto al actual, y en ese caso, lo actualiza.

Validacion.php es el conjunto de métodos, para la comprobación del los formularios de la página.

5.2.2 Subsistema de Ampliaciones

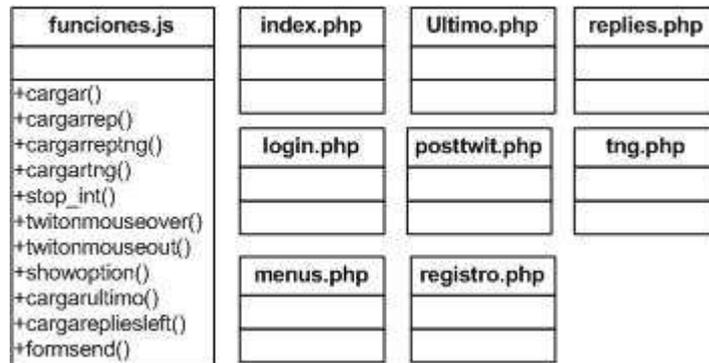


Eti_functions.php es el conjunto de métodos que comprende la gestión de etiquetas ya sea para la creación modificación o eliminación de las misma.

Devdivtng.php, menustng.php, posttag.php, etimain.php son el conjunto de scripts que basándose en los métodos de eti_functions.php generan el contenido a mostrar por el sistema de presentación. Así por ejemplo devdivtng.php devuelve la “friend_timeline” del usuario filtrada por etiquetas. Menustng.php y posttag.php generan los menús y el formulario para la sección de etiquetas. Y etimain.php devuelve una lista con los usuarios correspondientes, en formato HTML para directamente, ser utilizados por el subsistema de presentación.

Este subsistema esta orientado a incluir otras clases y scripts, aumentando la funcionalidad de la aplicación de una manera sencilla.

5.2.3 Subsistema de Presentación



Este subsistema presenta la información de una forma atractiva para el usuario. Funciones.js contiene el conjunto de métodos, que haciendo uso de ajax, nos permite presentar la información a los usuarios. Hace uso de dos frameworks para ajax como son prototype.js, y scriptaculus.js

El resto, son scripts que generan los contenidos de la página de forma dinámica, utilizando las funciones contenidas en funciones.js, y estableciendo periodicidades, para mantener constantemente la página actualizada, sin la necesidad de que el usuario recargue la página, ahorrando así un tiempo considerable.

5.3 Implementación

En este apartado se hará una definición de las clases y los métodos más relevantes para el funcionamiento de la aplicación.

5.3.1 Subsistema de aplicación principal

5.3.1.1 twitter.class.php

Esta clase, es un “wrapper” para el uso de la interfaz de twitter, es decir, twitter ofrece un conjunto de métodos en forma de “url” que al ser llamados, por ejemplo desde un navegador, producen como resultado un feed, o archivo de texto con formato XML, con la finalidad de que sea fácilmente procesable.

twitter.class.php esta escrita en utilizando el formato de orientación a objetos que nos permite PHP, y por tanto posee unos atributos, independientes para entidad, y unos métodos que actúan sobre esos atributos. En este caso, los atributos son el nombre de usuario y contraseña y el formato en que se procesaran los feeds.

Los métodos más importantes de esta clase son:

Twitter() – constructor del objeto sobre el que se ejecutaran los métodos posteriormente. Una vez creado el objeto, introducimos en sus campos el nombre de usuario y contraseña, de forma que este objeto representa únicamente a un usuario de twitter.

Process() – Esta función recibe por parámetros una url y unos parámetros. Esa url es alguna de las funciones que ofrece mediante su API. Este método, establece una conexión mediante curl con el servidor de twitter, y utiliza los parámetros para realizar la operación correctamente.

El resto de los métodos son simplemente un “wrapper” para

cada método del API de twitter. Es decir, cada método selecciona cual es la url correcta y cuales son los parámetros correctos para llamar a Process(), y posteriormente ejecutan Process().

5.3.1.2 Db_functions.php

Esta clase contiene todos los métodos de los que hace uso la aplicación para interactuar con la base de datos. Los métodos implementados son los de mas corriente uso en cualquier tipo de base de datos relacionales.

Db_connect() y db_disconnect() que establecen y destruyen la conexión con el servidor de bases de datos. Contienen los parámetros de la conexión a la base de datos como son el nombre de usuario de la base de datos, la contraseña, la dirección del servidor etc.

El resto de los métodos son las consultas básicas, tales como save_in_db(), que recibe por parámetro una tabla, un campo, y un valor y lo almacena en la base de datos.

Select_db() que ejecuta una consulta del tipo "select" sobre la base de datos a la que estemos conectados pudiéndole pasar por parámetro las condiciones para la consulta.

Update_db() y delete_db() actualizan o borran un dato concreto de la base de datos en función de los parámetros recibidos.

5.3.1.3 Utilidades.php

Esta clase contiene un conjunto de métodos necesarios para la ejecución de la aplicación pero independientes del resto de los métodos e independientes de entre si. Como son:

Comprueba_login() sirve para el control de sesiones php, comprueba en la base de datos que el nombre de usuario y contraseña recibidos por parámetros existen y son correctos. En ese caso crea una sesión única para ese usuario.

Hora() recibe un texto por parámetro sin formato y lo transforma en la hora de forma correcta.

Procesamensaje() reducirurl() mostrandourl() muestraetis() son un conjunto de métodos para el procesado de los textos que se imprimirán en la aplicación. De forma que se reconozcan las url, si son demasiado largas se recorten, se reconozcan los nombres de los usuarios de twitter y se introduzca un enlace a su perfil en twitter, y se por último que muestre las etiquetas correspondientes a un usuario.

5.3.1.4 Utiles_fechas.php

Es un conjunto de métodos para la conversión de fechas entre los formatos de de MySQL y el formato español de fechas. Además provee de otras utilidades para calcular el tiempo transcurrido entre diferentes fechas.

5.3.1.5 Validacion.php

Es otro conjunto de métodos pero en este caso orientado a validar y comprobar la información introducida por cada formulario. Con el propósito de evitar posibles ataques XSS y errores e incongruencias en la base de datos.

El método mas importante en este caso es el método que valida el formulario de registro `validar_registro()` que comprueba si el usuario ya existe, si el email es valido, que la contraseña tiene una longitud mínima y esta correctamente introducida en los dos campos etc.

5.3.1.6 Devdiv.php

Es un script que obtiene alguna de las posibles “time_lines” es decir, lo que acaban de decir tus amigos, lo que acaba de decir todo el mundo, o lo que acaba de decir el usuario, y lo procesa, devolviendo un array con cada uno de los tweets en formato HTML.

5.3.2 Subsistema de Ampliaciones

5.3.2.1 eti_functions.php

Este archivo contiene todos los métodos necesarios para la gestión de las etiquetas, es decir, añadir, eliminar, modificar, mostrar una lista con los usuarios correspondientes a una determinada etiqueta, una lista con lo usuarios etiquetados con cualquier etiqueta, devolver todas las etiquetas etc.

5.3.2.2 Devdivtng.php

Este archivo contiene un script similar al del subsistema principal llamado devdiv.php que consistía en obtener los feed de twitter y convertirlos a un formato HTML, para poder ser mostrados directamente. En este caso se han introducido todas las modificaciones necesarias para implementar el sistema de etiquetas, es decir, devdivtng.php filtrará si un tweet se debe mostrar o no, en función de la etiqueta que tenga y de las restricciones establecidas en ese momento.

5.3.2.3 Posttag.php

Contiene un formulario y todo lo necesario para la introducción de las etiquetas en el sistema de etiquetas.

5.3.2.4 Etimain.php

Es un script que devuelve una lista completa de todas las etiquetas en la base de datos para un usuario concreto. Lo devuelve en formato HTML y listo para ser insertado en cualquier zona de la web.

5.3.3 Subsistema de presentación

5.3.3.1 funciones.js

Es un conjunto de funciones escritas en javascript que hacen que la aplicación ejecute los scripts de los sistemas principal y ampliación, y además son las encargadas de que mejorar la experiencia de usuario.

Cargar() y cargartng() son las funciones principales de este conjunto. Reciben como parámetro la página a cargar, ejecutan el script

correspondiente del sistema principal o ampliación y lo almacenan en unos divs ocultos en la página web. Ya que el resultado de los scripts es simplemente HTML, solamente es copiarlos dentro del div correspondiente.

A continuación se compara lo que se está mostrando con lo contenido en el div oculto, y si es diferente, se actualizan los tweets que se están mostrando.

Todo esto utilizando las librerías Prototype y Scrip.taculo.us que a su vez hacen uso de AJAX. Se han utilizado para cargar los tweets en los divs correspondientes sin tener que recargar la página y sin la interacción del usuario con lo que se consigue que la página sea más atractiva para el usuario.

Cargarrep() y cargarreptng() debido a las restricciones que Twitter nos impone con respecto al número de peticiones a algunos de sus feeds, era conveniente tener estos dos métodos que básicamente establecen cada cuánto tiempo se debe recargar una página determinada.

Básicamente ejecutan cargar() y cargartng() periódicamente, con un número de segundos determinado por las restricciones de twitter y por la página que deseamos cargar.

Stop_int() este método se encarga de parar la ejecución de los métodos anteriores. Es utilizado para hacer un efecto de pause sobre la "time_line" para que el usuario pueda interactuar con los tweets que están en ese momento mostrándose.

Twitonmouseover() y twitonmouseout(), establecen los estilos que se muestran en el div cuando el ratón pasa por encima o el ratón

deja de estar encima del tweet.

Showoption() muestra el conjunto de opciones disponibles sobre las que el usuario puede actuar en el tweet actual.

Formsend() es el método encargado de enviar los formularios sin tener que recargar la página. Lo realiza mediante AJAX, y consigue una mejor experiencia de usuario, ya que el usuario puede realizar otras acciones en la misma página, mientras el formulario se esta mandando. Los métodos tradicionales de envío de formularios impiden esto, ya que necesitan la recarga de la página para dicho envío, y el usuario debe esperar a que se realice el envío del formulario y a que se recargue la página para seguir usando la aplicación.

5.3.4 Scripts Generadores HTML

El resto de los scripts de este subsistema se encargan de generar el código HTML para que sea fácilmente modificable y personalizable, para un usuario concreto, y de manera ordenada. Existen scripts para generar la columna principal de la página, las columnas laterales, y a su vez, como ya hemos visto en los otros subsistemas existen también generadores para los contenidos de dichas secciones.

Por otra parte, los menús de navegación han sido también extraídos, colocados en otro script aparte, de forma que no es necesario buscar entre las páginas de la aplicación donde están los menús, y son fácilmente y altamente personalizables.

Además también incluimos aquí los scripts necesarios para generar los formularios tanto de registro como de login que se comunican con el subsistema principal para añadir esta funcionalidad a la aplicación.

Capítulo 6: Manual de Usuario

En este capítulo se hará una descripción del funcionamiento de la aplicación.



PFC: Twitter Basado en el api de Twitter Search

Inicio Tags'n'Groups

Usuario: proyecto

Contraseña: *****

Login

Registrate
¿Olvidaste la Contraseña?

PFC:Twitter

Debes iniciar sesion para comenzar a utilizar la aplicacion

La primera página a la que tendrá acceso el usuario al iniciar la aplicación, es con formulario de login o acceso, y con las opciones de registrarse o recuperar la contraseña, en el caso de que la hubiera perdido.

Una vez que el usuario ha introducido el nombre de usuario y contraseña correctamente, accederá a su panel de usuario, donde podrá ver por defecto su “friend_timeline” en la columna central, su estado actual a la derecha, y sus “replies” o respuestas en el panel izq.



Desde esta sección, puedes acceder a ver cada uno de tus listas de tweets, como por ejemplo, tu “histórico” que muestra tus últimos tweets, “replies” que muestra en la columna central las diez ultimas respuestas, “public_timeline” que muestra los mensajes de todo el mundo. También puedes acceder al sistema de etiquetas, o salir de la aplicación.

Histórico:

The screenshot shows a web interface for 'PFC: Twitter', which is based on the Twitter API. The page has a search bar at the top right and a navigation menu with options like 'Inicio', 'Tags'n'Groups', 'Historico', 'Amigos', 'Replies', 'Public Timeline', 'Mensajes Directos', and 'Salir'. The main content area is divided into three columns. The left column contains a status update form with an 'Enviar' button and a 'Restablecer' button. The middle column displays a list of tweets from the user 'proyecto', including timestamps and the text of each tweet. The right column shows a large, stylized quote of the most recent tweet: 'proyecto: haciendo unas capturas para la documentacion del proyecto'.

PFC: Twitter Basado en el api de Twitter

Inicio **Tags'n'Groups**

Historico Amigos Replies Public Timeline Mensajes Directos Salir

Actualiza tu estado

Enviar Restablecer

Replies

Hora 8:25:03

 **vonmacklein**
@proyecto
<http://tinyurl.com/...> ahi va el Shared Items

Hora 8:18:22

 **vonmacklein**
@proyecto yo creo que comparto eso creo xD

Hora 8:36:30

 **proyecto** haciendo unas capturas para la documentacion del proyecto

Hora 8:24:29

 **proyecto** @penyaskito anda... pensaba que dabas la direccion de tus compartidos en greader... xD

Hora 8:23:30

 **proyecto** @penyaskito following por ahi tambien xD

Hora 8:20:50

 **proyecto** @vonmacklein si me pasas esa direccion.. pues leo lo k compartes

Hora 8:20:29



proyecto:

haciendo unas capturas para la documentacion del proyecto

Hora 8:36:30

Histórico muestra los últimos tweets que ha realizado el usuario.

Replies:

The screenshot shows a web application titled "PFC: Twitter" with the subtitle "Basado en el api de Twitter". It features a search bar and navigation tabs for "Inicio", "Tags'nGroups", "Historico", "Amigos", "Replies", "Public Timeline", "Mensajes Directos", and "Salir". The main content area displays a list of replies to a tweet. Each reply includes a timestamp, a user profile picture, the user's name and handle, and the text of the reply. The replies are as follows:

- Hora 8:25:03: **yonmacklein** @proyecto <http://tinyurl...> ahi va el Shared Items
- Hora 8:18:22: **yonmacklein** @proyecto yo creo que comparto... eso creo xD
- Hora 7:33:20: **SmasSive** @proyecto <http://www.goog...>
- Hora 7:32:07: **momiator** @proyecto gracias sabia que se me olvidaba algo. Suerte que tenemos los pastelitos del Dia de Internet que lo solucionaran todo xDDD
- Hora 7:31:28: (partially visible)

On the left side, there is a section for "Actualiza tu estado" with a text input field and "Enviar" and "Restablecer" buttons. Below it, a "Replies" section shows a tweet from **yonmacklein** @proyecto with the text: <http://tinyurl...> ahi va el Shared Items.

On the right side, there is a sidebar with the text: "proyecto: haciendo unas capturas para la documentacion del proyecto" and a timestamp "Hora 8:36:30".

Muestra los últimos mensajes dirigidos a ti, identificándolos por la arroba (“@”) que precede al nombre de usuario.

Etiquetas y Grupos:

The screenshot displays the PFC: Twitter web application. At the top, it says "PFC: Twitter" and "Basado en el api de Twitter". There is a search bar on the right. Below the header, there are navigation tabs for "Inicio" and "Tags'n'Groups".

The main content area is divided into three columns:

- Left Column:** "Actualiza tu estado" (Update your status) with a text input field and "Enviar" (Send) and "Restablecer" (Reset) buttons. Below it is a "Replies" section showing a tweet from "SmasSive @proyecto" at 6:02:23 with the text "claro!".
- Center Column:** A tweet stream. The top tweet is from "danisy" at 6:49:20 with the text "m0la, nos dan alojamiento, 'observación astronómica', visita a Atapuerca, un paseo en tren y una visita al castillo de Burgos ^^". Below it is a tweet from "Fotomaf @victoriano1" at 6:47:19 with the text "no hombre! que la 20D viene con el carisisisísimo 18-55 :) Seguro que lo cuelan, nolo dudes :D".
- Right Column:** A section titled "proyecto:" showing the text "haciendo captura". Below it is a section titled "Etiquetas" (Tags) listing: "twitbarna (2)", "twitche (1)", "twitmad (7)", and "twittsev (2)".

A la derecha se muestran las etiquetas del usuario, mientras que en la columna central se muestran por defecto los tweets de los usuarios con alguna etiqueta. Si seleccionas una etiqueta en concreto solo se mostrarán los tweets de los usuarios con esa etiqueta.

Capítulo 7: Pruebas

Esta aplicación ha sido diseñada para ser utilizada por un usuario medio, y en los navegadores actuales como son Firefox e Internet Explorer. Además ha sido probado para otros sistemas minoritarios como son el conjunto de usuarios de Apple, con su sistema operativo Mac Os X y su navegador Safari, o los usuarios de Linux, con sus dos navegadores principales Firefox, para el entorno Gnome y Konqueror para el entorno KDE.

Para esto hemos utilizado un hosting estaba a disposición del alumno, y hemos instalado ahí nuestra aplicación web, de forma que esa accesible a todo el mundo, aunque no esta publicitada y por tanto solo podrán acceder a la aplicación aquellos usuarios a los que sea comunicada la dirección.

El proceso de instalación como ya hemos comentado anteriormente requiere:

- Servidor web, capaz de interpretar PHP.
- Servidor de bases de datos MySQL.
- Plugin para Curl disponible o ya instalado.

Una vez comprobado que el hosting satisface todos los requisitos para la instalación, procedimos de la siguiente forma:

- Generando un script de creación de las bases de datos necesarias. Para esto utilizamos la herramienta PHPMyAdmin que permite exportar dicho script.
- Instalamos ese script en el servidor, utilizando la herramienta, también disponible en el hosting, PHPMyAdmin, y su opción para importar scripts.
- Por ultimo copiamos los archivos de la aplicación al servidor, a un directorio apropiado.

Una vez instalada correctamente la aplicación la ejecutamos dirigiendo nuestro navegador a la dirección de nuestro hosting y al directorio que le hemos asignado anteriormente.

Esta aplicación ha sido probada en todos los navegadores actuales con más repercusión:

Firefox: Comprobado su funcionamiento en la versión actual Firefox 2.0.0.12 tanto en sistemas operativos Windows, como en Linux y Mac Os X.

Internet Explorer: Comprobado su correcto

funcionamiento en la última versión (IE7) de su navegador, y probado también en IE6, aunque en ambas requiere algunas modificaciones en la presentación (HTML/CSS).

Opera: Probado en la versión actual tanto bajo Linux, como bajo Windows, también requiere algunas modificaciones en la presentación (HTML/CSS). Aunque es totalmente funcional.

Safari: Ha sido probado bajo su sistema operativo nativo Mac Os X, y su funcionamiento es totalmente correcto.

Konkeror: Probado en una distribución Linux, Ubuntu con el gestor de ventanas KDE, funciona correctamente aunque podría necesitar algunos ajustes de presentación (HTML/CSS).

Estos ajustes que podría necesitar la aplicación en algunos navegadores se produce por la diferente forma de interpretar las etiquetas HTML y CSS de los navegadores actuales. Se esta trabajando para que todos los navegadores interpreten de la misma manera los mismos contenidos HTML/CSS pero hasta que ese momento llegue habrá que trabajar con “hacks” para cada navegador en concreto.

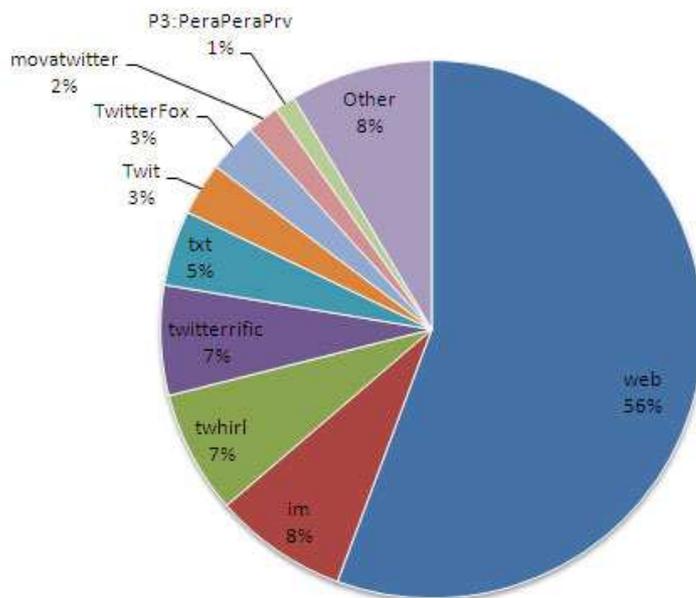
Por otra parte, este software ha sido probado por varios usuarios invitados por el alumno y la experiencia fue satisfactoria en todos los casos, aunque en diferente grado.

Gracias a estas colaboraciones se corrigieron muchos de los errores en la aplicación que no se manifestaban, en el uso realizado por el alumno.

Capítulo 8: Comparación con otras alternativas

Debido a que twitter es una aplicación muy reciente y con unas tecnologías muy actuales y populares, existen gran cantidad de clientes de twitter en diferentes lenguajes de programación y con diferentes características.

Todas estas aplicaciones, son aplicaciones de escritorio, es decir, aplicaciones para que un único usuario las ejecute en su ordenador. Las plataformas que están mas de moda y en las que los clientes para twitter mas éxito están teniendo son Adobe Air, y Microsoft Silverlight.



Tanto Air como Silverlight son la evolución de la plataforma Flash, que permitía embeber RIAs (Rich Internet Applications) en la web fácilmente. Estas dos nuevas plataformas consiguen que puedas usar RIAs directamente en tu escritorio. Otra aplicación similar son los widgets de los sistemas Mac Os X, y Linux, que también permiten embeber RIAs, directamente en tu escritorio.

Este gráfico ha sido obtenido del blog “Read Write Web” que hace un estudio de cuales son los clientes mas usados.

Se puede observar en el gráfico que las formas mas habituales (64%) de interactuar con twitter es mediante web o mediante

mensajería instantánea, evitando el uso de clientes twitter externos.

El otro 36% se lo reparten los diferentes clientes para twitter como son twhirl(basado en adobe Air), twitterific(la aplicación de escritorio para Mac), y el resto son mensajes de texto desde el móvil, plugin para firefox, y luego una gran cantidad de clientes que se reparte el 8%.

De estos resultados se pueden extraer algunas conclusiones como, que existen bastantes aplicaciones que no tienen establecido el nombre del cliente, esto hace que muchos tweets se contabilicen con “desde web” cuando el realidad no lo son.

Otra conclusión es que un gran número de usuarios lo usa desde mensajería instantánea, porque ofrece más dinamismo que desde la web.

Estos dos grupos de usuarios son los potenciales usuarios de esta aplicación web, ya que esta web ofrece mejor experiencia de usuario que la web original y a la vez es tan dinámica con la mensajería instantánea.

Capítulo 9: Conclusiones

La realización de este proyecto me ha supuesto enfrentarme por primera vez al desarrollo de una aplicación completa, desde sus inicios con la planificación correspondiente.

En un principio el proyecto era muy abierto y podría haberse tomado de muchas formas, una de las cosas que mas tiempo me llevo fue idear y saber exactamente que iba a programar, con que objetivos y funcionalidades.

Mi proyecto tenía el objetivo principal de ser un cliente para twitter que añadiera alguna funcionalidad, y que además mejorara la experiencia de usuario que posee la propia página de twitter. En base a eso, gran parte de la investigación se centró en establecer cual era la ampliación mas factible, que finalmente fue la implementación de un sistema de etiquetas, y por otro lado, como mejorar la interfaz de twitter, y con ello la experiencia de usuario. La alternativa en este caso fue JavaScript y Ajax. Una vez establecido que iba a desarrollar las

mejoras en JavaScript y Ajax, fue difícil saber que mejoras serian las necesarias y programarlas en JavaScript correctamente.

Uno de los puntos críticos del proyecto estuvo en desarrollar el script que muestra las posibles opciones sobre un tweet. Ese script es de gran complejidad ya que modifica los atributos de múltiples divs que están en continuo cambio, y además, necesita mantener información para todos en general. Otro punto crítico fue un pequeño cambio en la API de twitter que hizo que tuviera que modificar cada uno de los métodos de mi clase en PHP.

He de destacar también que es la primera vez que en este departamento se trabaja sobre el API de twitter y que una vez establecidas estas primeras clases sobre este API será mucho más sencillo desarrollar aplicaciones para futuros alumnos y futuros proyectos en este departamento.

Capítulo 10: Bibliografía

En general:

- <http://www.wikipedia.org/>

Grupo en Google para el desarrollo de aplicaciones para twitter:

- <http://groups.google.com/group/twitter-development-talk/web/api-documentation>

Página Oficial del lenguaje de programación PHP:

- <http://www.php.net/>
- <http://es.php.net/curl>

En particular esta es imprescindible ya que es necesario un amplio conocimiento en el funcionamiento de la librería curl para php y que funciones están disponibles.

Prototype:

- <http://www.prototypejs.org/api>
- <http://globalmoxie.com/projects/prototype-pdf/index.shtml>

Scriptaculous:

- <http://script.aculo.us/docs>

Sistema de etiquetas:

- <http://www.pui.ch/phred/archives/2005/06/tagssystem-performance-tests.html>

Esta es la bibliografía básica que usó durante todo el proyecto, aunque también he consultado algunos otros manuales en casos puntuales, como en dudas con MySQL.

