



UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR

INGENIERÍA TÉCNICA DE TELECOMUNICACIÓN
TELEMÁTICA

PROYECTO FIN DE CARRERA

Apuntweets.

Herramienta de colaboración sobre materiales educativos integrada con Twitter

Autor: Carlos Azaustre Rodríguez

Director: Israel Gutiérrez Rojas

Tutora: Patricia Arias Cabarcos

14 de septiembre de 2012

Agradecimientos

A mis padres Montse y David, por criarme y educarme en el esfuerzo continuo y en luchar para conseguir mis objetivos y sueños. Por enseñarme a pensar y por ayudarme sin pedirlo.

A mi hermana Amaya y mi abuela Julia por estar siempre ahí cuando se les necesita y demostrar su cariño día a día. Y en general a toda mi familia por ayudarme y creer en mi.

A mis compañeros de la Universidad, Adolfo, Ana, Bea, Marcela, Marta, Javi, Raquel, Sergio y Juan Alberto. Algunos os fuisteis antes de la Universidad, otros ya acabasteis, pero todos me habéis aportado algo como persona y formáis una parte importante de mi vida.

En especial a Paola, por su paciencia, su cariño, sus ganas de vivir y de aprender continuamente que hacen que se te contagie esa misma vitalidad y perseverancia que ella tiene. Gracias por estar siempre ahí, por animarme en los momentos de bajón y animarme en los momentos de euforia. Por compartir todo conmigo durante todo este tiempo y el que nos queda.

A mis director y tutora de proyecto, a Israel por ser un buen amigo y ayudarme a llevar este proyecto adelante desde cero y a Patricia por tutelarme en la recta final para poder presentarlo. Gracias por vuestro apoyo, implicación y amistad.

A todos vosotros, que siempre habéis estado a mi lado, muchas gracias.

Resumen

Este proyecto fin de carrera trata del diseño y desarrollo de una herramienta de colaboración sobre materiales educativos integrada con Twitter.

Apuntweets (Apuntes + Tweets), así es como lo hemos llamado, es un proyecto de e-learning que permite a alumnos y profesores colaborar en la anotación sobre un material educativo (apuntes de clase, enunciados de prácticas, etc.) desde su creación y puesta a punto hasta su uso en clase y en casa.

Se ha diseñado e implementado una herramienta basada en tecnologías web (HTML5, Javascript, PHP) que permite la participación tanto de personal docente como de alumnos en la discusión sobre un material educativo, por ejemplo los apuntes de una clase.

Aprovechando el auge de las redes sociales y el microblogging, constará de una capa social basada en la red Twitter, a cuyos servicios se accederá a través de su API (interfaz de programación de aplicaciones).

Apuntweets permite que los profesores puedan colgar apuntes de sus asignaturas en este formato y los alumnos, a través de sus cuentas de Twitter o desde la misma aplicación, comenten en ellos, pregunten dudas, las resuelvan entre ellos y con el profesor, ofrezcan alternativas, etc.

Abstract

This final degree project is about the design and development of a collaboration tool to annotate educational resources integrated with Twitter.

Apuntweets (“Apuntes”, Spanish for notes + tweets) is a e-learning project that allows teachers and students to annotate educational resources (class notes, lab session wordings, etc.) from creation and up-to-dating, to interactions in the class or at home.

A tool was designed and developed, based on web technologies (HTML5, Javascript, PHP), that enables the participation of teaching staff and learners in discussions about educational resources, for instance the teacher notes of a session. Taking advantage of the social networks and microblogging hype, it will have a social layer based on Twitter, accessing its services via the Twitter API (Application Programming Interface).

Apuntweets allows teachers to upload their notes to the tool as a web page, and students can comment on them, ask questions, solve peers problems, offer alternatives, etc. using their Twitter clients or directly from the tool.

Índice general

Índice general	X
Índice de figuras	XVI
Índice de tablas	XVIII
1. Introducción	1
1.1. Motivación del proyecto	1
1.2. Objetivos	2
1.3. Contenido de la memoria	3
2. Estado del arte	5
2.1. Introducción	5
2.2. HTML5	5
2.2.1. Introducción	5

2.2.2. Estructura	7
2.2.2.1. Nuevos elementos (semánticos).....	7
2.2.2.2. Nuevos atributos.....	10
2.2.2.3. Nuevos atributos globales.....	11
2.2.2.4. Nuevas APIs.....	12
2.2.2.5. Conclusión.....	15
2.2.3. Implementación de HTML5 en nuestra aplicación	16
2.3. Twitter	16
2.3.1. Introducción	16
2.3.2. Historia	17
2.3.3. Tecnología	18
2.3.4. Mensajes	18
2.3.5. Contenido	18
2.3.6. Autenticación	19
2.4. Javascript	20
2.4.1. Introducción	20
2.4.2. Historia	20
2.4.3. jQuery	21

2.4.3.1 Ventajas de jQuery.....	22
2.5. OAuth	23
2.5.1. Introducción	23
2.5.2. Flujo de autenticación.....	23
2.6. El aprendizaje electrónico: <i>e-learning</i>	25
2.6.1. Introducción	25
2.6.2. Experiencias del uso de Twitter en clases docentes.....	25
2.6.3. Plataformas de e-learning.....	28
2.6.4. Ventajas del e-learning.....	35
3. Desarrollo de la aplicación <i>Apuntweets</i>	39
3.1. Introducción.....	39
3.2. Arquitectura.....	39
3.2.1. Arquitectura Tecnológica	39
3.2.2. Arquitectura por módulos de la aplicación	41
3.2.3. Arquitectura de la Base de datos	43
3.3. API de Twitter	45
3.3.1. Librería TwitterOAuth en PHP.....	45

3.4. Interfaz de usuario	47
3.4.1. Diseño orientado al profesor.....	49
3.4.2. Diseño orientado al estudiante.....	49
3.5. Desarrollo de la aplicación	51
3.5.1. Registro de la aplicación en Twitter.....	51
3.5.2. Diseño de la interfaz de usuario.....	54
3.5.3. Desarrollo de módulos funcionales	59
3.5.3.1. Gestión de Usuarios	59
3.5.3.2. Vista principal	60
3.5.3.3. Vista de archivos	61
3.5.3.4. Módulo de comentarios.....	61
4. Pruebas	63
4.1. Escenarios de pruebas	63
4.2. Pruebas en local.....	68
4.3. Pruebas en servidor.....	70
5. Historia del proyecto.....	73
5.1. Distribución temporal.....	73

5.1.1. Documentación (1 mes)	74
5.1.2. Implementación del código (3 meses)	74
5.1.3. Diseño de la interfaz (1 mes)	75
5.1.4. Pruebas de funcionamiento en local (2 meses).....	75
5.1.5. Pruebas de funcionamiento en Servidor (1 mes).....	75
5.1.6. Redacción de la memoria (7 meses)	75
5.2. Presupuesto del proyecto	76
5.2.1 Costes de personal.....	76
5.2.3. Presupuesto total.....	78
6. Conclusiones y trabajos futuros	79
6.1. Conclusiones	79
6.2. Líneas futuras	80
A. Manual de usuario.....	85
A.1. Requisitos	85
A.2. Manejo por personal docente	85
A.3. Manejo para todos los usuarios.....	86

B. Glosario de Términos.....	87
Bibliografía.....	89

Índice de figuras

Figura 2.1: Logo de HTML5.....	6
Figura 2.2: Compatibilidad de HTML5 en navegadores (extraída de [3])	8
Figura 2.3: Nuevos elementos en HTML5 (extraída de [4]).....	9
Figura 2.4: Logo de Twitter.....	17
Figura 2.3: Diagrama de flujo de autenticación OAuth (extraída de [5])	24
Figura 2.4: Plataforma Moodle (Extraída de [18])	29
Figura 2.5: Plataforma Claroline (Extraída de [18])	30
Figura 2.6: Plataforma Udemy (Extraída de [18]).....	31
Figura 2.9: Plataforma eDhii (Extraída de [18]).....	34
Figura 2.10: Plataforma Peer2Peer University (Extraída de [18]).....	35
Figura 3.1: Arquitectura tecnológica de la aplicación	40
Figura 3.2: Dispositivos funcionando en España, en millones de unidades	41
Figura 3.3: Arquitectura por módulos de la aplicación	42
Figura 3.4: Arquitectura de la base de datos	44

Figura 3.5: Autorización de Twitter en la aplicación (Extraída de [14]).....	47
Figura 3.6: Formulario de carga de material educativo en la aplicación.	49
Figura 3.7. Visor de apuntes y columna de comentarios (1)	50
Figura 3.8. Visor de apuntes y columna de comentarios (2)	50
Figura 3.9. Párrafo comentado	51
Figura 3.10: Creación de una aplicación en Twitter.....	52
Figura 3.11: Configuración OAuth.....	54
Figura 3.12: Pantalla de Inicio	55
Figura 3.13: Petición de autorización a Twitter.....	56
Figura 3.14: Confirmando la autenticación	57
Figura 3.15: Vista Principal de la aplicación	58
Figura 3.16: Vista principal de la aplicación con rol de administrador.....	59
Figura 3.17 Vista de archivo con comentarios.	62
Figura 5.1 Diagrama de Gantt de la realización del proyecto.....	73
Figura 5.2 Fases de desarrollo en el Diagrama de Gantt	74
Figura 6.1: Principales sistemas operativos móviles en España (Extraída de [19]).....	82
Figura 6.2: Descargas de Apps en España (Extraída de [19]).....	83

Índice de tablas

Tabla 4.1. Pruebas de funcionamiento en local.....	69
Tabla 4.2. Pruebas de funcionamiento en el servidor.....	71
Tabla 5.1: Costes de personal	76
Tabla 5.2: Costes de material	77
Tabla 5.3: Coste total.....	78

Capítulo 1

Introducción

1.1. Motivación del proyecto

HTML5 es la nueva revisión del lenguaje de programación de páginas web. En el momento de la realización de este proyecto fin de carrera se encontraba en una etapa adelantada de borrador y poco a poco los navegadores web la van implementando y soportando para que se convierta en el lenguaje del futuro de la web.

Esta es una de las motivaciones de este proyecto, el estudio de las nuevas tecnologías como son HTML5 y CSS3 para desarrollar una interfaz amigable y acorde a los futuros estándares de la web y el acceso a APIs de terceros para conexión a redes sociales, como por ejemplo la de la red Twitter. Las ventajas de la utilización de aplicaciones web son que no requieren instalar ningún software en el equipo del usuario, salvo el navegador web, que ya viene instalado en la mayoría de los sistemas operativos. Al no tener que instalar nada, el usuario no tiene que preocuparse si dispone de la última versión de la aplicación, ya que esta es actualizada por los administradores y el usuario no tiene que hacer nada.

Usamos Twitter por el auge de las redes sociales, en especial el microblogging. Twitter ha conseguido cambiar la manera en la que nos llegan las noticias, podemos enterarnos de un hecho que sucede en cualquier parte del mundo con solo mirar nuestro *timeline* (*línea de tiempo*

con los comentarios de los usuarios que seguimos), los *retweets* (compartir de nuevo una publicación) que hacen nuestros contactos, mirar los *trending topics* (temas del momento)... es esta experiencia la que me ha hecho decantarme por su uso, lograr unir el uso que se hace de Twitter con una plataforma de *e-learning* que fomente la colaboración y el aprendizaje activo entre alumnado y personal docente. Twitter nos encaja muy bien en este proyecto porque a fin de cuentas una clase docente es en realidad un conjunto de personas y por tanto forman una red de usuarios con necesidades específicas de comunicación.

1.2. Objetivos

Como se ha introducido en las motivaciones del proyecto, el objetivo de este proyecto es el estudio de las nuevas tecnologías que nos ofrece HTML5 como otras tecnologías web, para desarrollar una herramienta que permita colaborar sobre material educativo para compartirlo, trabajar en él, aprender... (apuntes de clase, prácticas de laboratorio) y explorar el funcionamiento social que se puede dar a la aplicación mediante el API de Twitter para conseguir una herramienta *e-learning* útil.

A continuación se listan los objetivos principales a cumplir por este proyecto:

- Estudio de las nuevas formas de autoaprendizaje, aprendizaje activo y componente social en la educación.
- Estudio de la tecnología HTML5.
 - Estudio de las tecnologías web del futuro: HTML5, CSS3, JS,...
- Estudio de tecnologías para la interacción social de usuarios.
 - Desarrollo de una capa social que permite que los usuarios (alumnos/profesor) interaccionen entre sí.
- Aprender a documentar las conclusiones obtenidas sobre el desarrollo de la herramienta, aquellas limitaciones que se han

encontrado y cómo ha influido en la forma final de la aplicación el día de su presentación

La finalidad de esta memoria es que todos los aspectos que se han plasmado en estos objetivos, y otros que vienen a colación, queden correctamente descritos en las distintas secciones que la componen. En la siguiente sección se detalla cómo se han estructurado los distintos capítulos para ello.

1.3. Contenido de la memoria

Una vez hecha la introducción, en el Capítulo 1, la memoria se estructura como sigue:

- En el capítulo 2 hablamos sobre el estado del arte del proyecto. Destacando las tecnologías empleadas en la realización de esta herramienta como son principalmente HTML5 y la red social Twitter. También dedicamos un apartado a comentar el término e-learning.
- En el capítulo 3 se explica el desarrollo de la aplicación. Describiendo su arquitectura, tecnologías empleadas, pasos dados durante el desarrollo y configuración. Así como el uso que hacemos del API de Twitter, el registro de la aplicación y los métodos de autenticación.
- En el capítulo 4 se documentan las pruebas realizadas para la depuración de la aplicación así como los problemas encontrados y las soluciones implementadas.
- En el Capítulo 5 se detalla la historia del proyecto, el tiempo dedicado a cada fase de desarrollo y su presupuesto final.

- En el Capítulo 6 se plantean algunos trabajos futuros que complementarían y/o ampliarían la funcionalidad de la versión actual.

Como parte de la memoria se ha incluido un apéndice para el Glosario de Términos que aclara toda la terminología tratada en siglas durante el desarrollo del proyecto. Y otro anexo a modo de manual de manejo de la aplicación tanto para usuarios administradores como para el resto de usuarios.

Finalmente se incluyen las referencias consultadas en el desarrollo del proyecto, que incluyen las especificaciones a las que ha sido necesario recurrir para implementar aspectos concretos de las aplicaciones desarrolladas, libros y recursos web.

Capítulo 2

Estado del arte

2.1. Introducción

Este capítulo tiene como objetivo presentar la idea de la aplicación web así como las distintas tecnologías que constituyen la base para el desarrollo de este proyecto. Aunque en esta introducción no se comprendan a priori muchos de los términos utilizados, es objetivo de este capítulo que a la finalización del mismo se comprendan perfectamente.

En primer lugar, se presenta la tecnología HTML5 que es una de las bases de este proyecto. Se comentan las novedades de esta versión del lenguaje estándar de internet y sus aplicaciones. Posteriormente contamos la historia de Twitter, otro de los pilares de este proyecto. Por último hablaremos del término *e-learning* y los cambios que supone en los métodos de enseñanza.

2.2. HTML5

2.2.1. Introducción

HTML5 (HyperText Markup Language version 5) está empezando a ser el reemplazo del actual (X)HTML, una de las patas de la web desde su

nacimiento. Precisamente en un momento en el que la web está lo suficientemente madura, este estándar aprende de los errores cometidos e intenta solucionar la mayoría de problemas con los que un desarrollador web se encuentra. Como muchas de sus novedades son interesantes y afectan directamente a la futura web, a continuación comentaremos los cambios más importantes. [1]



Figura 2.1: Logo de HTML5

Antes de seguir habría que aclarar que HTML5 sigue en borrador y lo seguirá estando durante algunos años más. El enfoque general ha cambiado bastante respecto a versiones anteriores de HTML, añadiendo semántica y accesibilidad implícitas, especificando cada detalle y borrando cualquier ambigüedad. También se tiene en cuenta que muchas páginas web actuales son dinámicas, pareciéndose más a aplicaciones que a documentos. Algo básico es que HTML5 está definido en base al DOM (la representación interna de una web con la que trabaja un navegador). El paradigma de la web 2.0 nos muestra el cambio de simples documentos HTML estáticos, a la evolución en aplicaciones web donde la gente colabora, estableciéndose relaciones sociales y formando una red de colaborativa de usuarios.

2.2.2. Estructura

2.2.2.1. Nuevos elementos (semánticos)

Hoy en día se abusa bastante del elemento *div*, que nos permite estructurar una web en bloques. En HTML5 hay varios elementos que sirven para estructurar mejor una página web, estableciendo qué es cada sección, y reemplazando en muchas ocasiones a *div*. Con este extra de semántica, será mucho más coherente y fácil de entender por otras personas. Y lo que es más importante, será trivial de entender para una máquina, dándole más importancia a unas secciones y pudiendo jugar con esos datos automáticamente. Concretamente, la tarea de un buscador será mucho más fácil, pero cualquier aplicación que “lea” páginas web se beneficiará. Estos son los nuevos elementos semánticos [2]:

- **section** representa una sección “general” dentro de un documento o aplicación, como un capítulo de un libro. Puede contener subsecciones y si lo acompañamos de las etiquetas de títulos de cabecera (h1-h6) y de grupos de títulos (hgroup) podemos estructurar mejor toda la página.
- **article** representa un contenido independiente en un documento, el caso más claro son las entradas de un blog o las noticias de un periódico online. Así, dentro de la portada podremos tener varios artículos demarcados semánticamente, por lo que una herramienta puede extraerlos fácilmente.
- **aside** representa un contenido que está muy poco relacionado con el resto de la página, como una barra lateral. Esencial para delimitar el contenido “importante” del contenido “de apoyo”, haciendo más caso al primero que al segundo.
- **header** representa la cabecera de una sección, y es de suponer que se le dé más importancia que al resto, sobre todo si la sección es un artículo.
- **footer** representa el pie de una sección, con información acerca de la página/sección que poco tiene que ver con el contenido de la página, como el autor, el copyright o el año.

- **nav** representa una sección dedicada a la navegación entre el sitio, como la típica barra superior de los periódicos.

La compatibilidad de los navegadores con este nuevo lenguaje ha ido evolucionando y prácticamente en la mayoría de las versiones actuales se admite un alto porcentaje de elementos HTML5. En la figura 2.2. de a continuación, podemos ver la compatibilidad de los navegadores actuales con los nuevos elementos semánticos descritos de HTML5 y los que se enumeran en el resto de este punto.

Summary
Calculation of support for currently selected criteria

	IE	Firefox	Chrome	Safari	Opera	iOS Safari	Opera Mini	Opera Mobile	Android Browser
		3.6: 50%						10.0: 38%	2.1: 40%
	6.0: 9%	9.0: 75%				3.2: 44%		11.0: 55%	2.2: 45%
	7.0: 12%	10.0: 77%	17.0: 85%			4.0-4.1: 50%		11.1: 61%	2.3: 46%
	8.0: 21%	11.0: 77%	18.0: 86%	5.0: 64%		4.2-4.3: 56%		11.5: 61%	3.0: 60%
Current	9.0: 44%	12.0: 77%	19.0: 86%	5.1: 72%	11.6: 65%	5.0: 71%	5.0-6.0: 24%	12.0: 68%	4.0: 66%
Near future	10.0: 70%	13.0: 78%	20.0: 88%	5.2: 74%	12.0: 70%				
Farther future		14.0: 78%	21.0: 88%						

Figura 2.2: Compatibilidad de HTML5 en navegadores (extraída de [3])

A continuación, en la figura 2.3, podemos ver un ejemplo de cómo cambiaría un documento escrito en HTML normal o XHTML, a un HTML5 con los nuevos elementos:

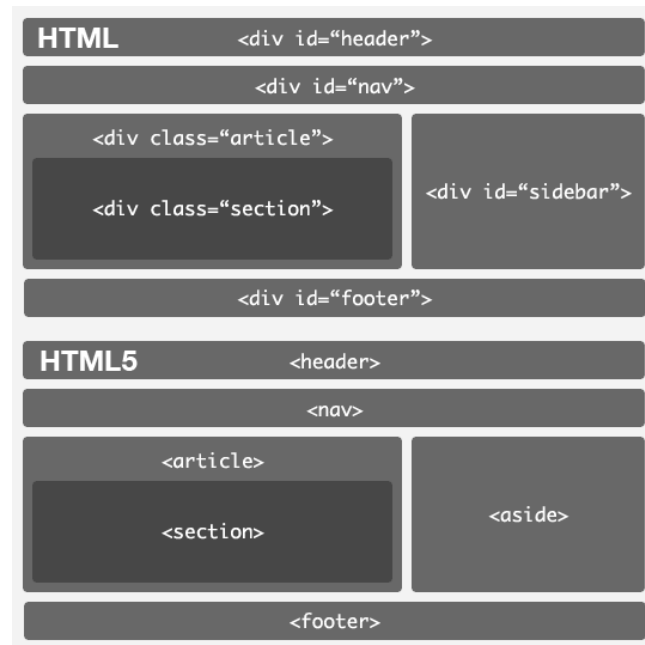


Figura 2.3: Nuevos elementos en HTML5 (extraída de [4])

- **audio** y **video** sirven para incrustar un contenido multimedia de sonido o de vídeo, respectivamente. Sin duda uno de los añadidos más interesantes, ya que permite reproducir/controlar vídeos y audios sin necesidad de *plugins* (complementos) como el de Flash. Se tratan de manera totalmente nativa como cualquier otro elemento, por ejemplo se pueden incluir enlaces o imágenes dentro de un vídeo.
- **embed** sirve para contenido incrustado pero no nativo, sino ejecutado por *plugins* como el de Flash. Aunque `embed` está soportado por casi todos los navegadores desde hace tiempo, es ahora cuando entra a formar parte del estándar y evita los problemas entre `object` y `embed`.
- **canvas** es un elemento complejo que permite generar gráficos. Es un lienzo en blanco para pintar, borrar, repintar, etc. Es la base de los juegos creados en HTML5.
- **figure** se plantea para asociar un contenido multimedia (una foto, un vídeo, etc) a un título o leyenda.

- **mark** representa un texto resaltado, por ejemplo para resaltar una búsqueda.
- **meter** representa una medida, como el número de KB. Tiene más sentido si lo unimos con el siguiente elemento, *progress*.
- **progress** representa el estado de una tarea, y se puede usar por ejemplo al subir un documento o al realizar varias tareas pesadas. Esto permitirá barras de tareas personalizadas y potentes.

2.2.2.2. Nuevos atributos

Los elementos ya existentes añaden nuevos atributos que permiten hacer cosas muy interesantes. Los más importantes o novedosos son:

- Atributo **ping** para el elemento **a**. Este atributo contiene una lista de URLs, las cuáles serán *llamadas* cuando un usuario haga click en ese enlace. Por ejemplo, un uso práctico sería para estadísticas.
- Atributo **autofocus** para los elementos de un formulario. Indica qué elemento de un formulario debe ganar el foco al cargar una página. Por ejemplo, en la página principal de Google la caja de texto gana el foco automáticamente para ayudarnos a escribir sin usar el ratón.
- Atributo **form** para los elementos de un formulario. Indica a qué formulario pertenece este elemento, y permite poner un elemento de un formulario en cualquier parte de una página. Tal y como está ahora, todos los elementos deben ir dentro de `<form>`.
- Atributo **required** para los elementos de un formulario. Indica que es obligatorio rellenar un valor para enviar ese formulario, algo que hoy en día se comprueba con Javascript o en el servidor.
- Atributos **autocomplete**, **min**, **max**, **multiple**, **pattern** y **step** para los elementos **input**. Estos atributos indican que el valor del input debe cumplir ciertos requisitos, por ejemplo seguir cierto patrón.

- Atributo **novalidate** para el elemento **form**. Esto deshabilitará la validación de sus elementos, algo útil si se usa dinámicamente (es decir, con Javascript, activando y desactivándolo).
- Atributos **formaction**, **formenctype**, **formmethod**, **formnovalidate**, y **formtarget** para los elementos de un formulario. Estos atributos modifican los atributos del formulario si el elemento es usado (por ejemplo un botón es pulsado o un input es rellenado). Esto permite que dependiendo qué botón usemos podemos tratar el formulario en diferentes páginas usando solo un formulario, algo complicado en HTML4.
- Atributo **scoped** en el elemento **style**. Esto permite aplicar estilos solo a cierto subárbol del documento. Por ejemplo, imagina que tenemos un elemento con id igual a “cabecera”; si añadimos el atributo **scoped** a un **style**, los estilos contenidos en él solamente se aplicarán a ese elemento y a sus hijos.
- Atributo **async** en el elemento **script**. Con este atributo especificamos que el código interno se puede ejecutar en cualquier momento de la página, mejorando la velocidad de carga.
- Atributo **manifest** en el elemento **html**. Esto permite indicar nuevos elementos, sobre todo será útil en aplicaciones web.
- Atributo **reversed** en el elemento **ol**. De esta forma la lista será numerada en orden descendente (3, 2, 1...).

2.2.2.3. Nuevos atributos globales

Además de los anteriores tenemos otros atributos que pueden ser aplicados a todos los elementos de un documento. Esto lo hacen especialmente conveniente si vamos a usar Javascript para modificarlos dinámicamente, ya que no tenemos que comprobar el tipo de elemento para usar los atributos comunes.

- Atributos **id**, **class**, **style**, **title**, **lang** y **tabindex**, ya existentes pero ahora permitidos en todos los elementos. Los atributos más usados se vuelven universales.
- Atributo **contenteditable**, para indicar que el elemento es editable por el usuario. Muy interesante, ya que ahora podremos editar cualquier cosa de manera trivial para el desarrollador, no solamente los elementos de un formulario.
- Atributo **contextmenu**, para indicar un menú contextual sobre un elemento. Como veis, está todo muy pensado para las aplicaciones web.
- Atributos **data-***, donde el asterisco puede ser cualquier nombre. Esto permite atributos personalizados, que luego podremos obtener con Javascript. Como por ejemplo el “drag&drop”, arrastrar un fichero al escritorio con el atributo “data-downloadurl”.
- Atributo **draggable**, para indicar que un elemento se puede arrastrar.
- Atributo **hidden**, para ocultar un elemento que ya no interesa.
- Atributo **spellcheck**, para indicar si el contenido de un elemento debe ser pasado por el corrector ortográfico.

2.2.2.4. Nuevas APIs

No está claro que todas las APIs siguientes se vayan a incluir en el estándar HTML5 propiamente, de hecho seguro que alguna de ellas se separará creando un estándar propio dedicado. De cualquier forma, estas son las nuevas APIs que nacen o se desarrollan en HTML5:

- Una API para **dibujar en 2D**, que se podrá usar junto al nuevo elemento canvas. Básicamente, se pueden pintar elementos sobre un lienzo, de manera similar a lenguajes como Java.

- Una API para **controlar los nuevos elementos multimedia**, video y audio. Así podremos controlar la reproducción con código Javascript, algo interesante pero que puede dar más de sí.
- Una API para **guardar datos localmente**, utilísimo para que las aplicaciones web puedan trabajar sin necesitar conexión a Internet. Ese DOM *storage* (almacenamiento) está ya implementado en las últimas versiones de los grandes navegadores, así que dentro de poco podremos disfrutar de esta clase de aplicaciones sin necesidad de otros tipos de almacenamiento local.
- Una API para que las **aplicaciones web puedan enlazarse a protocolos o tipos de archivos MIME**, otro añadido extremadamente útil. Esto permitiría abrir las fotos de tu disco duro directamente en una aplicación de retoque online, o un archivo mp3 en una biblioteca online, etc...
- Una API para **editar los campos que sean editables**. Esto permite controlar los elementos HTML que son editables por el usuario, tipo Word. Por ejemplo, Google Docs o editar HTML en emails será muy sencillo, y con esta API es trivial cambiar ese contenido con Javascript.
- Una API para **controlar las acciones Drag & Drop** sobre los elementos que se puedan arrastrar. Esto se puede conseguir actualmente con algunas librerías, pero con esta API el navegador lo permite de manera nativa y más poderosa.
- Una API para **controlar el historial** desde una aplicación web. Esto permitirá a las aplicaciones web que se muevan con Javascript añadir páginas al historial para que los botones Atrás-Adelante funcionen siempre.
- Una API para **habilitar la comunicación entre varias páginas web**. Es decir, si tenemos varios iframes externos en una web, podemos comunicarnos con ellos y compartir información de manera segura, por ejemplo con gadgets de Facebook o similares. Estos son los websockets y webRTC.

- **WebSocket** es una tecnología que proporciona un canal de comunicación bidireccional y full-duplex sobre un único socket TCP. Está diseñada para ser implementada en navegadores y servidores web, pero puede utilizarse por cualquier aplicación cliente/servidor. La API de WebSocket está siendo normalizada por el W3C, y el protocolo WebSocket, a su vez, está siendo normalizado por el IETF. Como las conexiones TCP ordinarias sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios WebSocket sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).
- El estándar **WebRTC**, permite comunicación de video y audio de alta calidad a través de la web sin plug-ins y de desarrollo abierto

Se han añadido a los elementos del DOM nuevas funciones y atributos que facilitan su uso y permiten realizar acciones muy usadas. Aquí comentaré los más interesantes, que trabajan sobre el documento (HTMLDocument) o sobre cualquier elemento (HTMLElement).

- La función **getElementsByClassName()** se añade a todos los elementos y al documento. Su funcionamiento es similar a `getElementById()`, pero en este caso selecciona todos los elementos del documento o de cierto subárbol del documento que contengan esa clase. Su definición es tan amplia que incluso elementos que contenga SVG o MathML pueden ser encontrados.
- El atributo **classList** está disponible para cualquier elemento, y contiene una lista con todas las clases que tiene ese elemento. Además contiene varios métodos que facilitan trabajar con esa lista: buscar, modificar, borrar, etc. Muy útil para trabajar con elementos que puedan tener más de una clase, es muy sencillo y muy conveniente de usar. Los enlaces también tienen un elemento

similar adicional llamado `relList` que permite lo mismo pero con el atributo `rel` (como el famoso `rel="nofollow"`).

- El atributo **`innerHTML`** se añade, por fin, al estándar. Prácticamente usado por todas las aplicaciones web y soportado por todos los navegadores, creo que todos los desarrolladores web lo conocen de sobra. Para aquellos que no lo conozcan, este atributo permite trabajar con el contenido de un elemento en texto plano, incluso cambiando elementos HTML que pueda haber. Igualmente, se añade a todos los elementos y al propio documento, pudiendo cambiar todo el contenido de una web.
- Los atributos **`activeElement`** y **`hasFocus`** están disponibles sobre un documento, y permiten conocer qué elemento está activo y si el documento tiene el foco, respectivamente.
- La función **`getSelection()`** se aplica también al documento y devuelve un objeto con el texto y elementos que están seleccionados.
- El atributo **`designMode`** es otra novedad sobre el documento e indica/modifica que el documento pueda o no ser editado.
- La función **`execCommand()`** se aplica sobre el documento y permite ejecutar acciones o “comandos” típicos de edición de documentos. Por ejemplo, con este método se llamaría a los útiles copiar/pegar, pero también a otros típicos como crear un enlace o cambiar el color de un elemento. Como el anterior, la mayoría de estos comandos trabajan sobre elementos editables.

2.2.2.5. Conclusión

Como hemos podido comprobar, las novedades de HTML5 se centran en facilitar la implementación de aplicaciones web, avanzar hacia la web semántica y limpiar el código redundante y obsoleto heredado de las anteriores versiones de HTML. De momento HTML5 no es un estándar implementado, si no un borrador, pero muchos navegadores ya

implementan parte de HTML5 y páginas experimentales juegan con los nuevos elementos.

2.2.3. Implementación de HTML5 en nuestra aplicación

En nuestra herramienta, el HTML5 está presente en toda la aplicación. El material educativo está maquetado siguiendo los estándares de la nueva versión, lo que me permitió desarrollar la estructura de la web más fácilmente y darle valor semántico a cada sección de tal manera que el contenido está dividido en *sections* (secciones), con sus cabeceras y títulos independientes, etc. Ésto nos facilita a la hora de integrarlo con Twitter de tal manera que las conversaciones desarrolladas a partir de los comentarios en Twitter estén en determinadas partes de la web y aporten contenido semántico. Dentro de estos elementos nos encontramos con los campos *header* (cabecera) y *footer* (pie de página) que definirán y darán valor a las cabeceras y pie de página.

Para darle un valor añadido a la aplicación, hemos añadido a los campos *input* de actualización de estado, la opción de que se puedan rellenar a través de nuestra voz gracias al nuevo método *x-webkit-speech* (sólo válido en Google Chrome) que recoge a través del micrófono el texto hablado y lo traduce a escrito.

2.3. Twitter

2.3.1. Introducción

Twitter es una red social basada en el microblogging, con sede en San Francisco (California). Creada por Jack Dorsey en marzo de 2006 y lanzada en julio del mismo año, la red ha ganado popularidad mundialmente y se estima que tiene más de 200 millones de usuarios, generando 65 millones de *tweets* (publicaciones o actualizaciones del estado de un usuario mediante mensajes de texto plano con un máximo de

140 caracteres) al día y maneja más de 800.000 peticiones de búsqueda diarias [5].



Figura 2.4: Logo de Twitter.

Los tweets, que se muestran en la página principal del usuario. Los usuarios pueden suscribirse a los tweets de otros usuarios (a esto se le llama *seguir* y a los suscriptores se les llama *seguidores*). Por defecto los mensajes son públicos, pudiendo difundirse privadamente mostrándolos únicamente a seguidores. Los usuarios pueden *twittear* desde la web del servicio, desde aplicaciones oficiales externas, aplicaciones en dispositivos móviles (como los *smartphones*) o mediante SMS, disponible en ciertos países.

2.3.2. Historia

Twitter comenzó como un proyecto de I+D dentro de *Obvious, LLC*. Una pequeña *start-up* de San Francisco durante marzo de 2006. El nombre original del producto era *twtr*, inspirado por *Flickr*. Al principio fue usado internamente por la compañía desarrolladora hasta que lo lanzó oficialmente al público en octubre del mismo año. El servicio rápidamente ganó adeptos, y en marzo de 2007 ganó el premio *South by Southwest Web Award* en la categoría de blog.

Jack Dorsey es el creador de esta aplicación web y actual presidente del consejo de administración de *Twitter Inc.*, empresa que surgió a raíz de *Obvious, LLC* y el éxito cosechado por Twitter. A principios de 2008, el equipo de Twitter estaba compuesto por 18 personas, durante 2009 su plantilla se multiplicó por cuatro y sigue creciendo hasta los 450 que tiene en 2011.

2.3.3. Tecnología

La interfaz web de Twitter está escrita en Ruby on Rails, y los mensajes se mantienen en un servidor que funciona con software programado en Scala y además dispone de una API abierta para todo tipo de desarrolladores, lo cual supone una gran ventaja para nosotros que queremos integrar Twitter en nuestra aplicación como un servicio. Según Twitter, más del 50% del tráfico que reciben llega a través del API.

Actualmente han liberado la plataforma *Bootstrap*, que define una maquetación basada en CSS y Javascript para la creación de webs y aplicaciones y permitir de esta manera ahorrar tiempo a la hora de diseñar la web y centrarnos en la programación.

2.3.4. Mensajes

Los usuarios pueden agrupar mensajes sobre un mismo tema mediante el uso de *hashtags* (palabras o frases iniciadas mediante el uso de una “#” almohadilla). De forma similar, la “@” arroba seguida de un nombre de usuario se usa para mencionar o contestar a otros usuarios. Para volver a postear un mensaje de otro usuario y compartirlos con los propios seguidores, la función de *retweet* se marca con un “RT” en el mensaje.

A finales de 2009 se añadió la opción de listas, haciendo posible el seguir (así como el mencionar y contestar) listas de usuarios en vez de usuarios individuales.

Los mensajes fueron fijados a 140 caracteres máximo para añadir compatibilidad con los mensajes SMS, ya que a través de este sistema también es posible actualizar el estado.

2.3.5. Contenido

Aunque se duda de su finalidad y de la utilidad de Twitter, su creciente número de seguidores ha demostrado que Twitter es uno de los líderes en el

sector del microblogging. Microblogging es un servicio que permite a los usuarios enviar y publicar mensajes breves (140 caracteres en el caso de Twitter). Los usos más conocidos son: el seguimiento de eventos en directo, la retransmisión de charlas y ponencias a las que poca gente tiene acceso, el intercambio de opiniones durante un evento en el que la gente asiste como público o incluso comentarios sobre películas o debates retransmitidos en televisión.

La empresa de investigación de mercado Pear Analytics, analizó 2000 tweets (procedentes de los Estados Unidos y en inglés) durante un periodo de dos semanas de 11:00h a 5:00h en Agosto de 2009 y los separó en seis categorías:

- Noticias, 40%
- Conversaciones, 38%
- Retweets, 9%
- Spam, 4%
- Palabras sin sentido, 4%

Con estos datos podemos observar que la gran mayoría del contenido que se genera en Twitter es de tipo social, los usuarios conversan entre ellos comentando impresiones y generando debate a través de las noticias que se publican en la misma red. Esto hace que sea una buena herramienta para nuestra aplicación.

2.3.6. Autenticación

A partir del 31 de agosto de 2010, las aplicaciones de Twitter, desarrolladas por terceros deben usar OAuth, un método de identificación que no requiere que el usuario dé su contraseña a la aplicación. Anteriormente la identificación por OAuth era opcional, ahora es obligatoria y el método de autenticación nombre de usuario/contraseña ha quedado obsoleto y ya no es funcional. En la sección 5 de este capítulo se describe de una manera más detallada el funcionamiento de este protocolo.

2.4. Javascript

2.4.1. Introducción

Javascript es un lenguaje de programación que se utiliza principalmente para crear páginas web dinámicas.

Una página web dinámica es aquella que incorpora efectos como texto que aparece y desaparece, animaciones, acciones que se activan al pulsar botones y ventanas con mensajes de aviso al usuario.

Técnicamente, Javascript es un lenguaje de programación interpretado, por lo que no es necesario compilar los programas para ejecutarlos. En otras palabras, los programas escritos con Javascript se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios.

A pesar de su nombre, Javascript no guarda ninguna relación directa con el lenguaje de programación Java.

2.4.2. Historia

A principios de los años 90, la mayoría de usuarios que se conectaban a Internet lo hacían con módems a una velocidad máxima de 28.8 kbps. En esa época, empezaban a desarrollarse las primeras aplicaciones web y por tanto, las páginas web comenzaban a incluir formularios complejos. [6]

Con unas aplicaciones web cada vez más complejas y una velocidad de navegación tan lenta, surgió la necesidad de un lenguaje de programación que se ejecutara en el navegador del usuario. De esta forma, si el usuario no rellenaba correctamente un formulario, no se le hacía esperar mucho tiempo hasta que el servidor volviera a mostrar el formulario indicando los errores existentes.

Brendan Eich, un programador que trabajaba en Netscape, pensó que podría solucionar este problema adaptando otras tecnologías existentes (como ScriptEase) al navegador Netscape Navigator 2.0, que iba a lanzarse en 1995. Inicialmente, Eich denominó a su lenguaje LiveScript.

Posteriormente, Netscape firmó una alianza con Sun Microsystems para el desarrollo del nuevo lenguaje de programación. Además, justo antes

del lanzamiento Netscape decidió cambiar el nombre por el de JavaScript. La razón del cambio de nombre fue exclusivamente por marketing, ya que Java era la palabra de moda en el mundo informático y de Internet de la época.

La primera versión de Javascript fue un completo éxito y Netscape Navigator 3.0 ya incorporaba la siguiente versión del lenguaje, la versión 1.1. Al mismo tiempo, Microsoft lanzó JScript con su navegador Internet Explorer 3. JScript era una copia de Javascript al que le cambiaron el nombre para evitar problemas legales.

Para evitar una guerra de tecnologías, Netscape decidió que lo mejor sería estandarizar el lenguaje JavaScript. De esta forma, en 1997 se envió la especificación Javascript 1.1 al organismo ECMA (European Computer Manufacturers Association).

ECMA creó el comité TC39 con el objetivo de "estandarizar de un lenguaje de script multiplataforma e independiente de cualquier empresa". El primer estándar que creó el comité TC39 se denominó ECMA-262, en el que se definió por primera vez el lenguaje ECMAScript.

Por este motivo, algunos programadores prefieren la denominación ECMAScript para referirse al lenguaje Javascript. De hecho, Javascript no es más que la implementación que realizó la empresa Netscape del estándar ECMAScript.

La organización internacional para la estandarización (ISO) adoptó el estándar ECMA-262 a través de su comisión IEC, dando lugar al estándar ISO/IEC-16262.

2.4.3. jQuery

jQuery es una librería de Javascript que permite simplificar la manera de interactuar con documentos HTML, manipular el DOM, manejar eventos, agregar interacción mediante AJAX, etc.

Consiste en un único fichero Javascript que contiene las funcionalidades comunes anteriormente citadas. La característica principal de la librería es que permite cambiar el contenido de una página web sin necesidad de recargarla, mediante la manipulación del DOM y con

peticiones AJAX. La forma de interactuar con la página es mediante la función `$()`, un alias de la función `jQuery()`, que recibe como parámetro una expresión CSS o el nombre de una etiqueta HTML y devuelve todos los elementos nodo que concuerden con la expresión

2.4.3.1 Ventajas de jQuery

jQuery no es la única librería o framework de Javascript que existe en el mercado. Existen varias soluciones similares que funcionan igualmente bien pero jQuery es un producto con una aceptación por parte de los programadores muy buena y con un grado de penetración en el mercado muy amplio, lo que hace suponer que es una de las mejores opciones. Es un producto serio, estable, bien documentado y con un gran equipo de desarrolladores a cargo de la mejora y actualización del framework.

Otras alternativas a jQuery son MooTools y Prototype. MooTools tiene el inconveniente de que la sintaxis de su API y su sintaxis son más pesada que jQuery y no es tan fácil de manipular el DOM que con ella. Por otra parte, Prototype además de las desventajas que presenta MooTools se le añade que es el más lento de las 3 librerías en el rendimiento del selector, tiene demasiados métodos triviales en el núcleo y como añadido, necesita la librería `script.aculo.us` para crear efectos. [7]

También es destacable la gran comunidad de creadores de plugins o componentes, lo que hace fácil encontrar soluciones ya creadas para implementar directamente.

2.5. OAuth

2.5.1. Introducción

OAuth (Open Authorization) es un protocolo abierto, propuesto por Blaine Cook y Chris Messina, que permite autorización segura de un API de modo estándar y simple para aplicaciones de escritorio, móviles y web.

Para los desarrolladores de clientes, OAuth es un método para interactuar con ellos y publicar datos protegidos. Para los proveedores de servicio, OAuth proporciona a los usuarios el acceso a sus datos al mismo tiempo que protege sus cuentas. En otras palabras, OAuth permite a un usuario del sitio A compartir la información del sitio A (Proveedor de servicio) con el sitio B (llamado cliente) sin compartir toda su identidad

2.5.2. Flujo de autenticación

En el siguiente diagrama de flujo podemos observar cómo se lleva a cabo el procedimiento para autenticarse siguiendo el protocolo OAuth.

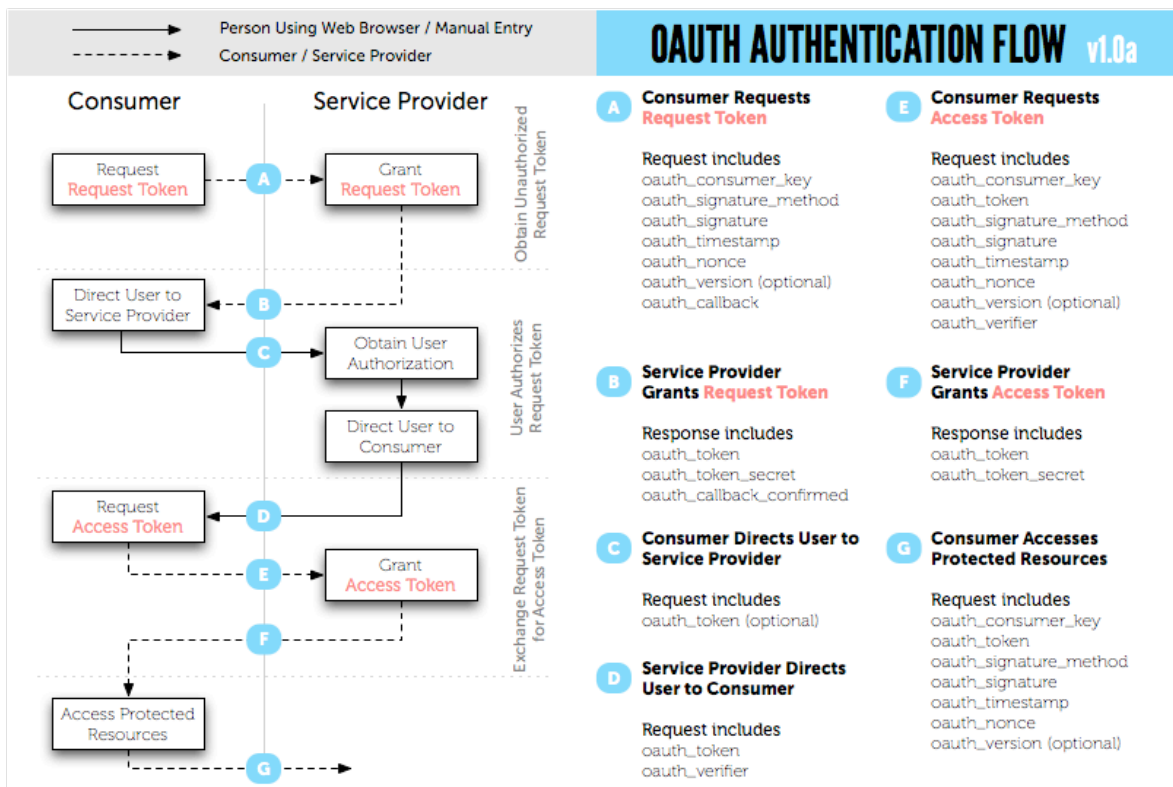


Figura 2.3: Diagrama de flujo de autenticación OAuth (extraída de [8])

En el protocolo intervienen dos entidades, la aplicación cliente (*Consumer*) y el proveedor de servicio (*Service Provider*). Desde que el usuario intenta acceder desde la aplicación móvil hasta que el proveedor de servicio (en nuestro caso, Twitter) le concede el acceso se suceden los siguientes pasos [9][10]:

- a) El usuario accede a la aplicación, nuestra aplicación sale a Twitter realizando una petición con un *Request Token*. Un *Request Token* es una solicitud de un valor usado por la aplicación web para obtener una autorización del usuario.
- b) Twitter responde al *Request Token* con una *Request Token Key* y una *Request Token Secret Key* que se usan para crear el enlace de autorización que permitirá al usuario acceder a Twitter a través de la aplicación cliente e identificarlo unívocamente.

- c) El usuario accede a través del enlace de autorización. El enlace aprueba el *Request Token* de vuelta a Twitter y una llamada del API acepta el *Request Token Secret*.
- d) Twitter envía al usuario/cliente un *Request Access Token* para poder identificarse con su usuario y contraseña de Twitter y otorgar permisos a Twitter para acceder a los datos a través de su API.
- e) Twitter concede el acceso aprobando el *Access Token*.
- f) El usuario queda identificado en la aplicación con su cuenta de Twitter y ya puede acceder a los servicios de la misma.

2.6. El aprendizaje electrónico: *e-learning*

2.6.1. Introducción

Se denomina aprendizaje electrónico (en inglés: e-learning) a la educación a distancia completamente virtualizada a través de los nuevos canales electrónicos (las redes de comunicación en especial internet) utilizando para ellos herramientas o aplicaciones web (Correo electrónico, páginas web, foros, mensajería instantánea, plataformas de formación, etc.) como soporte de los procesos de enseñanza/aprendizaje. En este caso concreto de “Apuntweets” nos referimos a e-learning como educación mejorada con tecnología y no a educación a distancia. Nos centramos en un curso mixto (“blended”) con parte de educación presencia y parte a distancia.[11]

2.6.2. Experiencias del uso de Twitter en clases docentes.

En este apartado vamos a enumerar algunas utilidades que se pueden dar a Twitter en el ámbito docente [12]

- **Twittear acerca de las fechas de exámenes y entregas de trabajos.**

Es uno de los usos más sencillos de Twitter en el aula para los profesores. De esta manera se puede configurar un *feed* (fuente de noticias a través de un servicio web) dedicado exclusivamente a fechas de entrega de trabajos, exámenes, etc.

- **Proporcionar un servicio de noticias actualizado para los alumnos.**

Subscribiéndose a diferentes canales de noticias para ofrecer distintas perspectivas y así poder comparar y contrastar las diferentes interpretaciones sobre acontecimientos y cuestiones.

- **Crear listas.**

Es interesante que los alumnos creen listas en Twitter con cuentas que tengan feeds relevantes a los objetivos de su titulación y llevar un diario de cualquier tendencia que vaya surgiendo. Por ejemplo en una titulación como la de Ingeniería Informática si se quiere estar al tanto de noticias/curiosidades sobre lenguajes de programación, es interesante tener una lista con cuentas de Twitter sobre ello. Si nos interesan las aplicaciones móviles, es útil tener en listas cuentas sobre Android, iOS, Windows Phone, etc... de esa manera estarán al tanto de las novedades más recientes

- **Seguimiento de *Trendings Topics*.**

De esta manera el profesor está al tanto de cómo las modas y las ideas se extienden entre los diversos medios de comunicaciones y puede aprender cómo utilizarlas en el aula si sus clases están relacionadas con la comunicación y la sociología.

- **Coordinar tareas.**

En lugar de usar el e-mail como forma de comunicación, los alumnos pueden usar Twitter para colaborar en diferentes proyectos y tener una referencia rápida y accesible para cualquier cambio.

- **Seguir un hashtag.**

Los profesores pueden incorporar etiquetas a sus lecciones y observar como las tendencias van creciendo y los usuarios usan las redes sociales para comunicar ideas.

- **Seguimiento de temas.**

El profesor puede hacer un seguimiento de debates preguntando a los alumnos que temas les gustaría tratar en el aula. Se puede suscribir a las correspondientes hashtags y llevar un seguimiento de los diferentes puntos de vista que se traten.

- **Hacer preguntas.**

Monica Rankin, profesora de Historia en la Universidad de Texas-Dallas comenta que utiliza Twitter con sus estudiantes para mantener un flujo de preguntas durante las clases. Para ello dispone de una pantalla donde se van mostrando los *tweets* durante el transcurso de las clases. [13]

- **Tomar y compartir notas.**

Un aula con recursos suficientes puede permitir a los estudiantes tuitear sus propios apuntes durante las clases y compartir con sus compañeros. Esta es la idea principal de la aplicación “Apuntweets” objeto de este proyecto. En lugar de que los alumnos compartan sus apuntes, la aplicación desarrollada está pensada de tal forma que el profesor facilite el material y los alumnos comenten sobre él y expresen sus dudas.

- **Sincronizar con un blog.**

Wordpress y otros sitios gratuitos de “blogging” se pueden sincronizar con Twitter publicando avisos de nuevas entradas. El profesor puede disponer de un blog sobre la asignatura que imparte y sincronizarla con su cuenta de Twitter. De esta manera sus alumnos pueden estar al tanto de las actualizaciones de la asignatura cada vez que escriba una nueva entrada. De igual manera, si los alumnos mantienen cada uno su propio blog, el profesor puede estar al tanto de las actualizaciones de sus alumnos si sincronizan sus blogs con Twitter y así no tendría que ir uno a uno visitándolos.

- **Publicación de materiales suplementarios.**

El profesor puede retweetear artículos, noticias, opiniones y otros datos interesantes que sean relevantes para que los alumnos sigan sus clases. Es un complemento excelente y cómodo para las clases.

- **Hacer encuestas.**

Los profesores pueden dar la opción a sus estudiantes de votar que actividades realizar o conocer sus opiniones sobre temas de actualidad.

- **Publicar videos.**

Los profesores que tengan acceso a una videocámara digital pueden colgar a través de un servicio de video conectado a Twitter (Youtube, Twiddeo, Ustream, etc) sus clases grabadas y así el estudiante puede acceder a ellas en cualquier momento.

2.6.3. Plataformas de e-learning.

Un sistema de gestión de contenidos es un programa que permite crear una estructura de soporte para la creación y administración de contenidos por parte de los participantes. El entorno de hardware y software diseñado para automatizar y gestionar el desarrollo de actividades formativas se conoce como plataforma de tele-formación o sistema de gestión de aprendizaje.

Un LMS (Learning Manegement System: sistema de gestión de aprendizaje) registra usuarios, organiza catálogos de cursos, almacena datos de los usuarios y provee informes para la gestión. Suelen incluir también herramientas de comunicación al servicio de los participantes en los cursos. Las mejoras en usabilidad (navegación fácil e intuitiva) y accesibilidad (posibilidad de acceso por personas con discapacidad) permiten salvar la brecha digital y extender las posibilidades de formación a mayor número de personas.

A continuación enumeramos una colección de plataformas gratuitas de aprendizaje electrónico existentes en el mercado, en concreto se detallan las 7 más usadas y conocidas, que nos permiten elaborar y compartir cursos [14].

- Moodle.

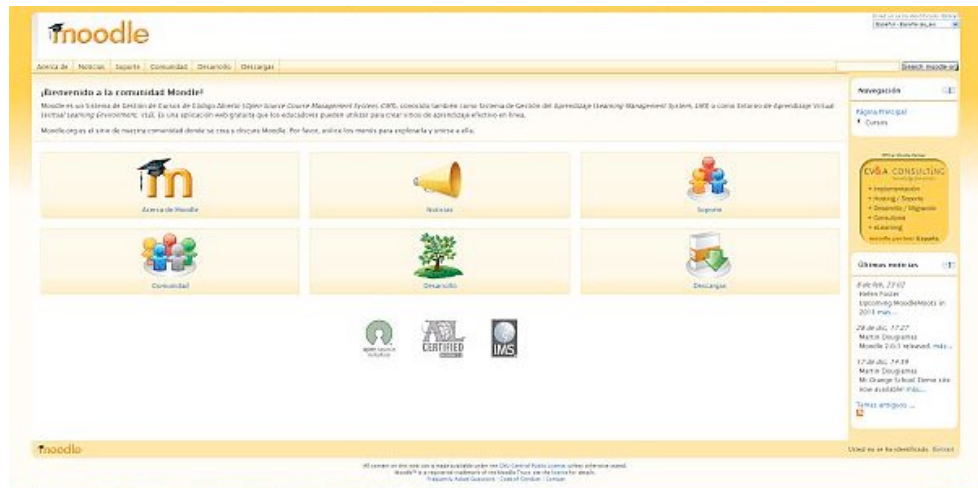


Figura 2.4: Plataforma Moodle (Extraída de [14])

Es una de las más conocidas y populares. Se trata de una plataforma completamente flexible que, mediante sus extensiones, módulos y temas, nos permite configurar el portal a medida, tanto para pequeños como grandes grupos de estudiantes. Posee una gran comunidad que le da soporte.

- **Claroline.**

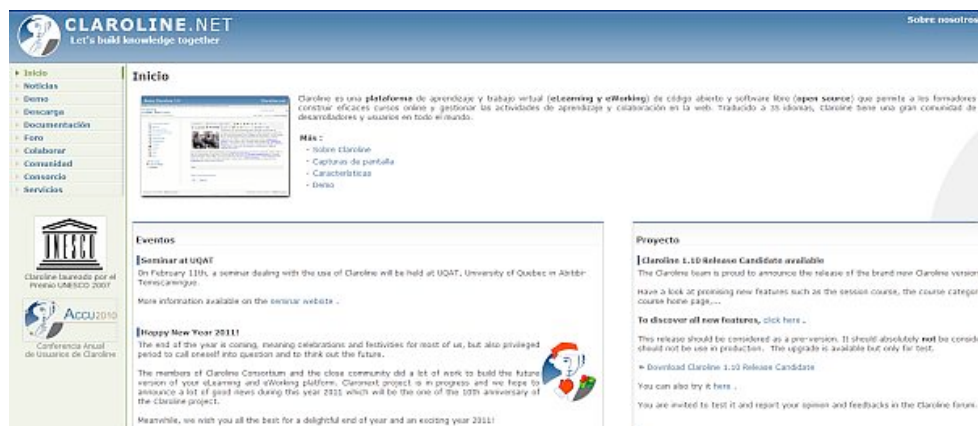


Figura 2.5: Plataforma Claroline (Extraída de [14])

Otra plataforma para diseñar y gestionar cursos online, con el añadido de poder crear espacios colaborativos. El docente tiene con esta plataforma una serie de herramientas que permiten escribir características del curso, participar en wikis, administrar foros, crear grupos de estudiantes, proponer tareas, etc.

- **Udemy.**

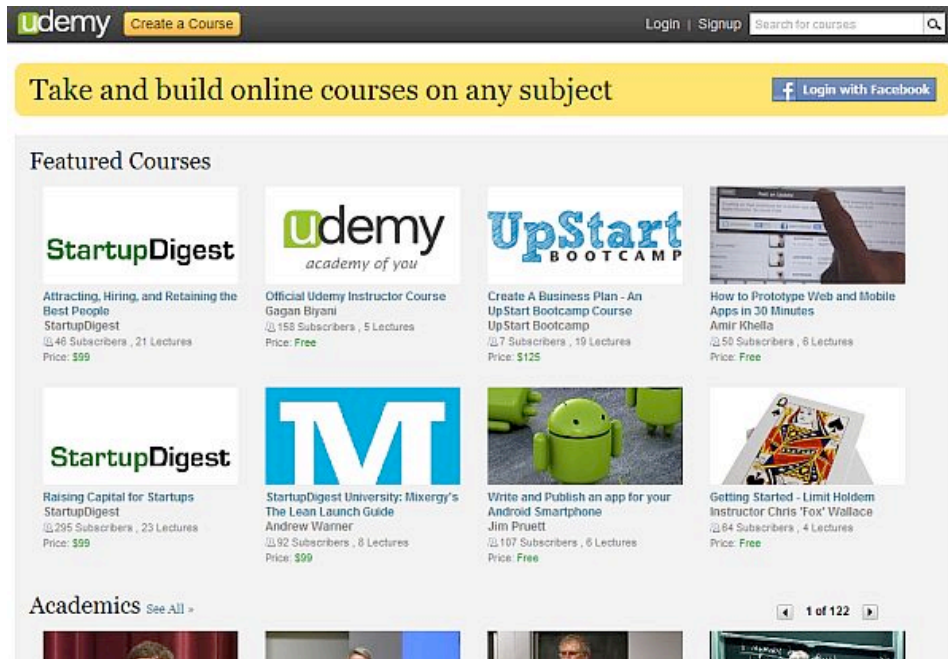


Figura 2.6: Plataforma Udemy (Extraída de [14])

Es un servicio online en el que simplemente hay que registrarse o acceder a través de Facebook Connect para poder crear cursos en pocos minutos, seleccionando categorías, subcategorías, e indicando el nivel de acceso, pudiendo ser público o limitado a los usuarios que invitemos.

Se puede subir material de todo tipo, presentaciones en PowerPoint, videos, audios, crear artículos, importar documentos de otros sitios, etc. También se pueden crear sesiones en vivo, utilizando una serie de herramientas para la interacción con los demás usuarios.

- **Rcampus.**

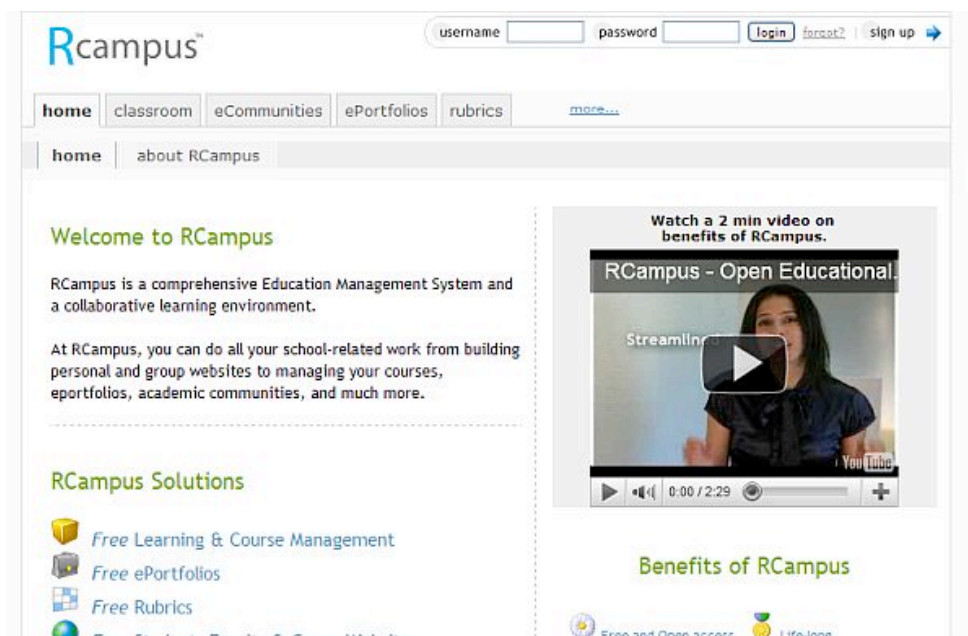


Figura 2.7: Plataforma Rcampus (Extraída de [14])

Otro recurso online, al igual que Udemy, destinado a la gestión de las herramientas de aprendizaje y entorno colaborativo en el que tan sólo hay que registrarse para crear cursos, recibir las tareas que se asignen a los alumnos y disponer de un acta de calificaciones.

- **Learnopia.**

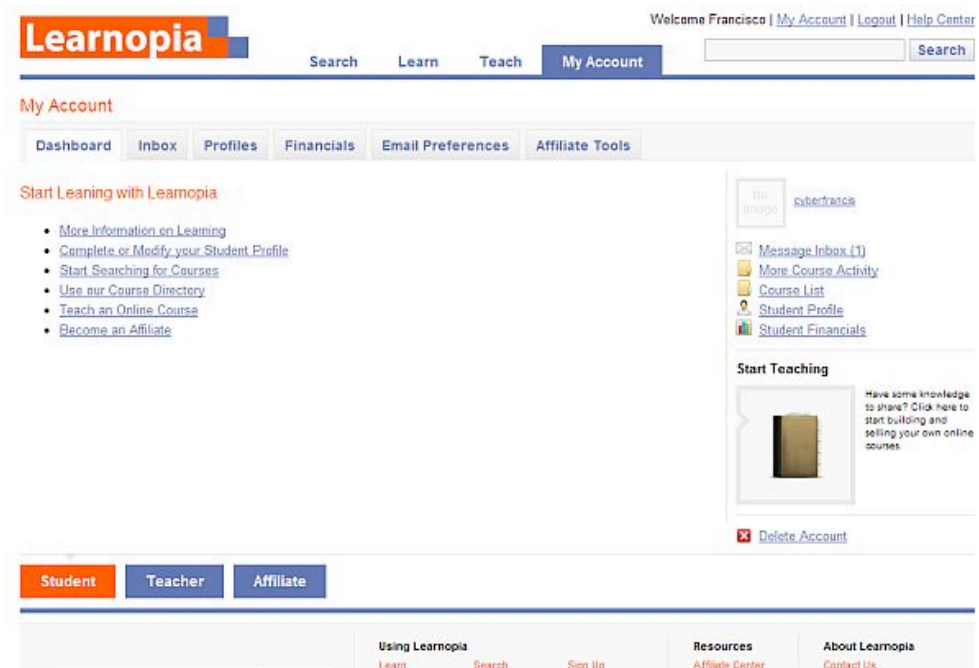


Figura 2.8: Plataforma Learnopia (Extraída de [14])

Otra plataforma que permite a los profesores crear, alojar e incluso vender (u ofrecer de manera gratuita) sus cursos y por otro lado a los alumnos se les permite acceder a dichos cursos.

- **eDhii**

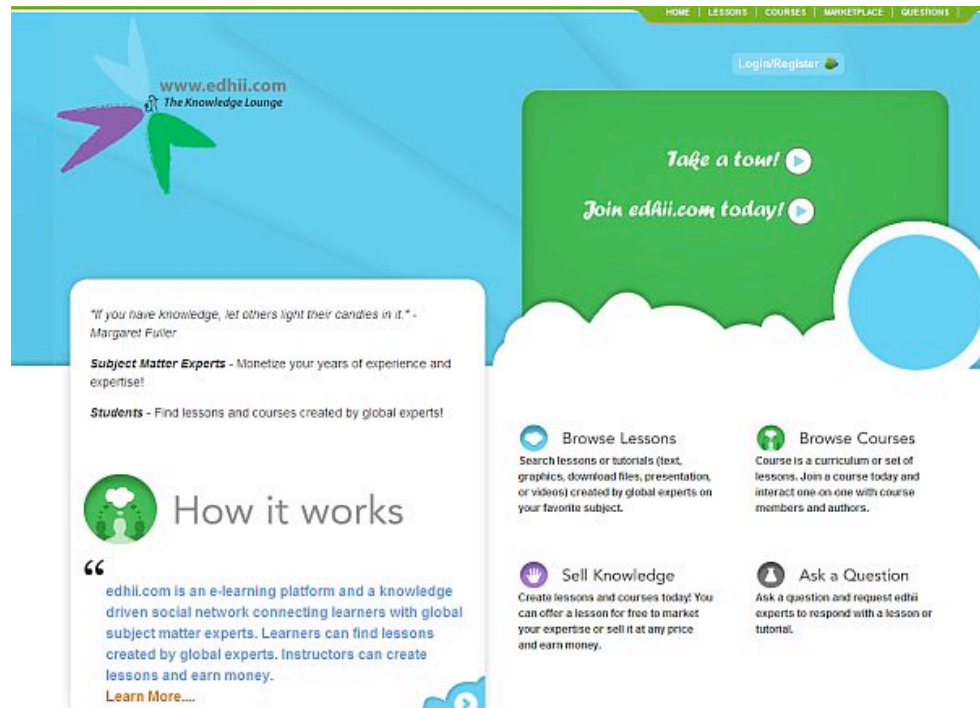


Figura 2.9: Plataforma eDhii (Extraída de [14])

Plataforma similar a Learnopia pero más amplia ya que se trata de un LMS con red social que permite a cualquier experto crear sus propios cursos y permite a los estudiantes buscar tanto cursos como lecciones.

- **Peer 2 Peer University**

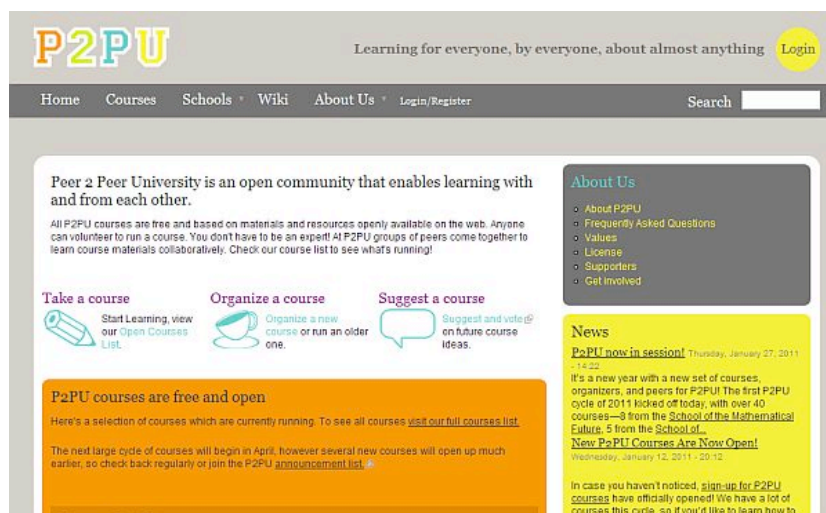


Figura 2.10: Plataforma Peer2Peer University (Extraída de [14])

Se trata de una comunidad abierta de usuarios en la que pueden compartir sus conocimientos a nivel universitario en sus respectivas áreas de manera gratuita. La limitación de esta plataforma es que hay que inscribirse en los períodos que establecen para poder participar en cada uno de los cursos.

2.6.4. Ventajas del e-learning.

El desarrollo del e-learning favorece un aprendizaje más flexible. Los y las estudiantes adoptan un papel activo en su proceso de aprendizaje, dejando de ser meros receptores pasivos de información o conocimiento. Para ello se les dota de espacios en los que es vital su participación, (chats, foros de discusión, correo electrónico...), además, aumenta la posibilidad de que sean ellos/as mismos/as quienes decidan qué, cómo y cuándo quieren estudiar.

Esta modalidad favorece también la interactividad, entendida ésta como la relación que se establece entre la persona que aprende o se forma y los demás componentes de la acción formativa; así este aspecto se desarrolla en diferentes ámbitos: con el contenido, con el/la tutora y con los demás

estudiantes. Para que se desarrolle esta interactividad son imprescindibles las herramientas de comunicación tanto sincrónicas como asincrónicas.

La teleformación nos ayuda a eliminar las barreras espacio-temporales que tradicionalmente habían supuesto una clara y evidente limitación a la información y a la educación. La modalidad e-learning no requiere un lugar concreto ni una coincidencia temporal. En definitiva, podemos decir que contribuye a democratizar la enseñanza, al hacer que ésta sea adaptable a cualquier colectivo y circunstancias.

El e-learning está permitiendo un cambio en la pedagogía y un cambio en el modelo clásico de enseñanza-aprendizaje. Del modelo presencial, donde el/la docente era la parte activa de una clase magistral, se ha pasado a un modelo a distancia en un aula virtual, donde el alumnado es el elemento más activo del proceso. Así el profesorado se transforma en facilitador del aprendizaje, ya que ofrece al estudiante herramientas para su (auto) aprendizaje, a la vez que resuelve sus dudas y necesidades. Podemos concluir pues que el e-learning se centra más en el aprendizaje que en la enseñanza. Para ello es muy importante cuidar la organización y disposición de los contenidos, así como la orientación del aprendizaje de los alumnos y alumnas, mediante tareas individuales y en grupo, con un seguimiento permanente por parte del tutor/a.

Las nuevas tecnologías de la comunicación nos están ofreciendo también distintas posibilidades y escenarios para pensar en la formación permanente. Parece ser que en un futuro muy cercano se tenderá a que los y las estudiantes y formandos tengan una muy buena formación básica y que la formación especializada se haga cuando se necesite (*just in time / just in case*, Estos términos se refieren al aprendizaje tradicional de “por si acaso se necesita”(just-in-case) frente al aprendizaje que se hace “en el momento que se necesita”(just-in-time)).

Otra ventaja puesta a disposición del alumnado e-learning es la posibilidad de acceder a numerosas fuentes de información. Además la información se recibe a través de distintos canales y se pueden así experimentar diferentes situaciones de aprendizaje.

Indudablemente el factor económico juega un papel decisivo; por un lado, los costes educativos han aumentado significativamente en las últimas

décadas, y por otro lado, las empresas se ven obligadas a pagar enormes sumas de dinero para capacitar adecuadamente a su personal: gastos de desplazamiento, de alquiler de aulas, de materiales didácticos, de personal especializado, etc. Sin lugar a dudas, la teleformación supone un desahogo económico porque requiere una fuerte inversión inicial, pero en ediciones sucesivas del curso los gastos son mínimos (alquiler de plataformas, actualización de contenidos, contratación de personal específico para las tutorías,...) y el gasto inicial se va amortizando.

En resumen, el e-learning supone:

- **Facilidad de acceso.**
La formación puede llegar a más personas, puesto que desaparecen las barreras espacio-temporales.
- **Elimina distancias físicas**
Se utilizan herramientas como el email, foros o chats para establecer la comunicación entre los usuarios (profesor/estudiantes).
- **Flexibilidad horaria**
El alumno accede en el momento que dispone de tiempo.
- **Aumento del número de destinatarios**
Esta modalidad de formación se puede dirigir a una audiencia mucho más amplia.
- **Favorece la interacción.**
Los alumnos pueden comunicarse unos con otros, con el profesor y con los recursos on-line disponibles en internet.
- **Disposición de recursos on-line y multimedia.**
Internet proporciona acceso instantáneo e ilimitado a una gran cantidad de recursos, como textos, artículos, gráficos, videos, etc...

Capítulo 3

Desarrollo de la aplicación *Apuntweets*

3.1. Introducción

En este capítulo describimos detalladamente el desarrollo de la aplicación, paso a paso desde el concepto inicial y su arquitectura, pasando por el uso del API de Twitter, así como librerías empleadas para su programación.

La aplicación tiene una parte *frontend* (lado cliente) que desarrollamos con HTML5 y una parte *backend* (lado servidor) en la que utilizaremos PHP para las llamadas a la base de datos y al API de Twitter.

3.2. Arquitectura

3.2.1. Arquitectura Tecnológica

La arquitectura se basa en un servidor web Linux con Apache Web Server instalado. La versión que usa el *hosting* (alojamiento web) utilizado es la 2.2.17. El servidor tiene configurado PHP en su versión 5.2.16 y una

base de datos MySQL versión 5.0.91-community-log según información proporcionada por el proveedor.

El usuario puede acceder como cliente desde cualquier plataforma, ya sea un PC de sobremesa, un portátil, un tablet o un smartphone, ya que la interfaz de la aplicación es HTML5 lo que la hace accesible desde cualquier dispositivo que tenga un navegador que soporte el estándar.

En la figura 3.1 podemos ver un esquema de la arquitectura tecnológica que tiene la aplicación y como a través de internet podemos acceder a ella desde cualquier punto con acceso sin depender de ningún sistema operativo, hardware específico, etc.

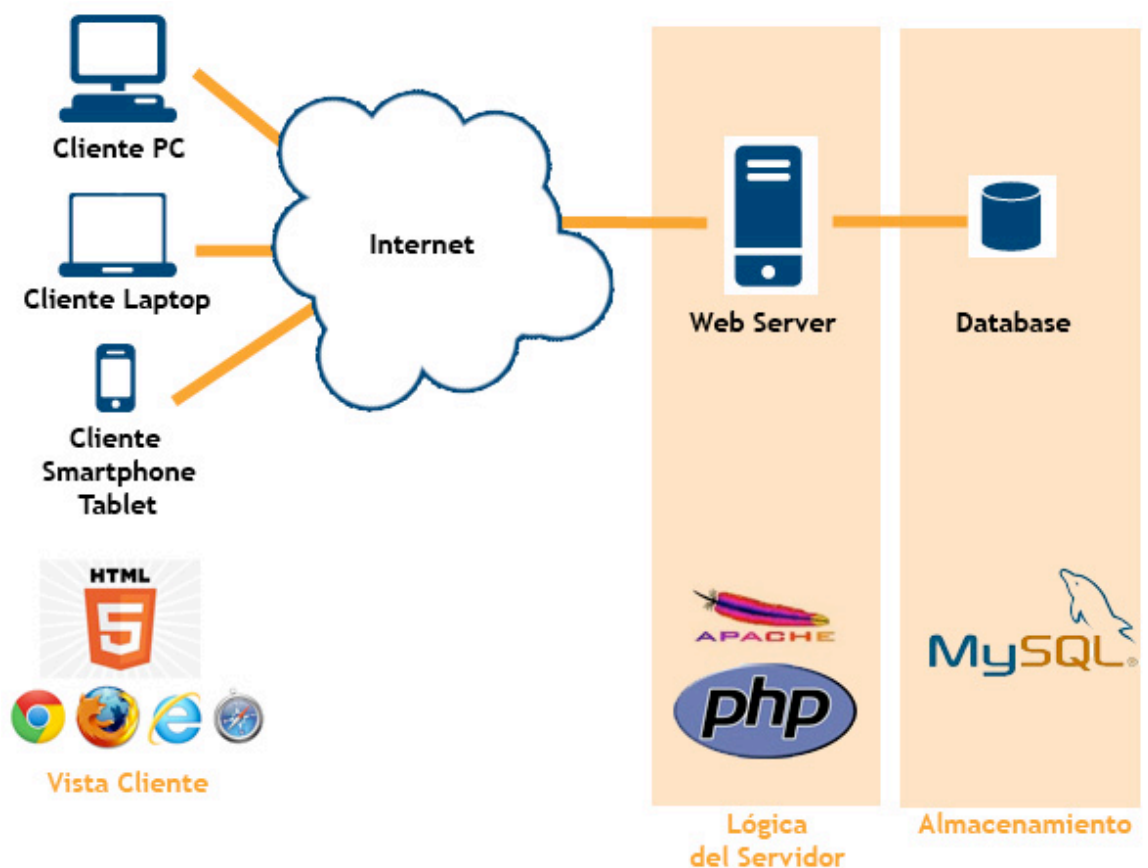


Figura 3.1: Arquitectura tecnológica de la aplicación

Nos hemos decantado por una aplicación web, dado que según las estadísticas de uso y centrándonos en España, un estudio reciente [15] muestra que el número de dispositivos en millones de unidades utilizados en nuestro país corresponden a 19 millones de ordenadores (sobremesa y portátiles) (53,07%), los smartphones se contabilizan en 14,8 millones de dispositivos (41,34%). Seguido tenemos los televisores que cuentan con una conexión a internet que son 1,1 millones de dispositivos (3,07%) y las tabletas que ocupan el último lugar con 900.000 unidades (2,51%). Todos estos números los podemos ver resumidos a continuación en la figura 3.2.

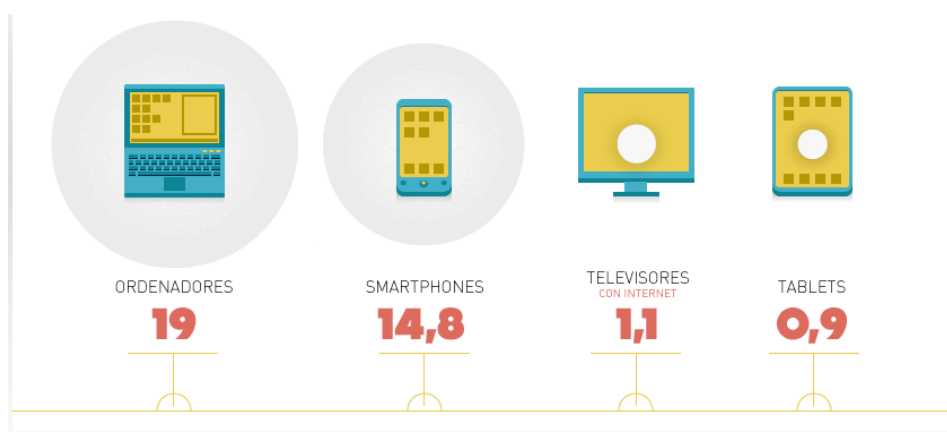


Figura 3.2: Dispositivos funcionando en España, en millones de unidades (Extraída de[15])

3.2.2. Arquitectura por módulos de la aplicación

Los módulos funcionales que tiene la aplicación y que permiten el flujo de navegación en ella, los podemos ver en la figura 3.3. A continuación explicaremos cada uno de ellos.

- **Index.**

Es la pantalla principal de la aplicación. A través de ella los usuarios entran en la aplicación autenticándose con su cuenta de twitter, ya sean usuarios estudiantes como usuarios docentes. Una vez autenticado el usuario, pasa a la pantalla principal

- **Principal.**

Es la pantalla principal de la aplicación una vez que el usuario se ha autenticado con su cuenta de Twitter. En ella se puede ver el listado de documentos (prácticas/apuntes) que hay subidos en la aplicación, ordenados por asignaturas. También hay un formulario de acceso por si se trata de un usuario docente y tiene cuenta de docente para poder autenticarse y subir archivos.

- **Principal (Rol Docente)**

Si el usuario es docente y tiene cuenta de acceso, puede autenticarse y subir archivos a la aplicación mediante un formulario. La pantalla es idéntica a la anterior salvo por el formulario de subida de archivos.

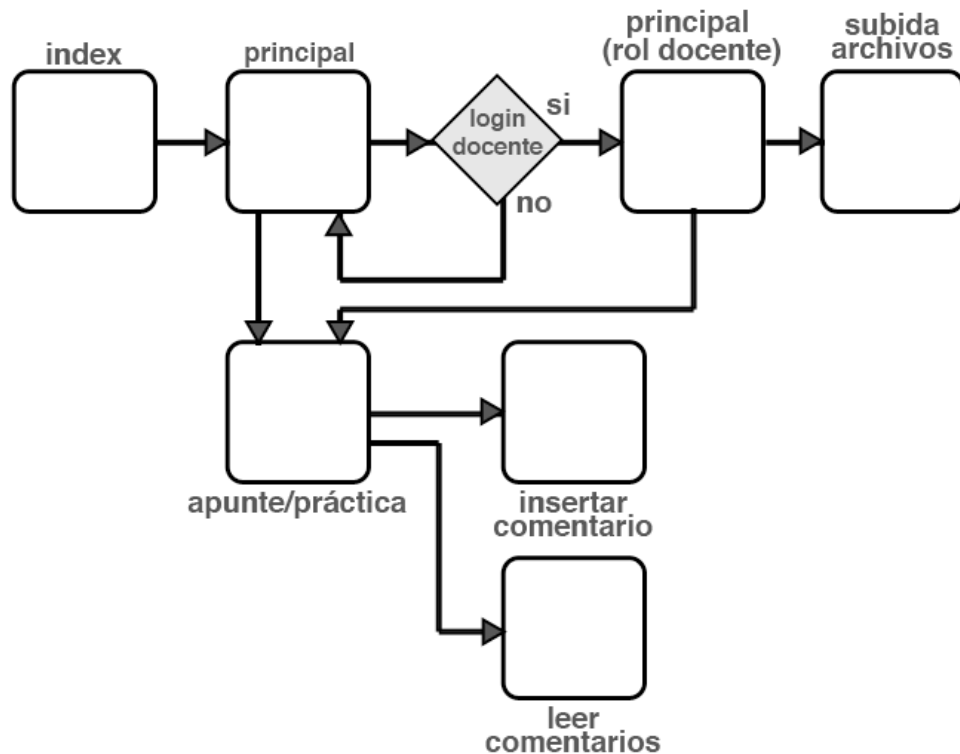


Figura 3.3: Arquitectura por módulos de la aplicación

- **Subida de Archivos**

Se trata de un formulario, de acceso exclusivo para docentes. A través de él se pueden subir prácticas/apuntes a la aplicación indicando su título, asignatura, titulación, curso, cuenta de twitter del docente y la URL donde tiene alojada el docente el documento. El sistema hace una copia al servidor del documento alojado en un servidor externo y lo presenta dentro de la aplicación.

- **Apunte/Práctica**

Los documentos cargados en la aplicación se muestran todos con el mismo formato, indicando los datos introducidos (Título, asignatura, curso, etc.) además del propio archivo incrustado. En el cual se pueden seleccionar los diferentes párrafos del documento para poder leer e insertar comentarios en cada uno.

- **Insertar comentario**

Se trata de un formulario en el cual introducimos un texto, comentando un apartado del documento que estamos visualizando. Este comentario quedará almacenado en la base de datos apuntando a un párrafo concreto.

- **Leer comentarios**

Seleccionando los diferentes párrafos del documento que estamos visualizando podemos leer los comentarios asociados, en la parte derecha de la pantalla.

En los subapartados siguientes se explica en más detalle el desarrollo de los diferentes módulos aquí presentados.

3.2.3. Arquitectura de la Base de datos

La base de datos del proyecto es muy sencilla, consistiendo en tres tablas, una tabla de usuarios administradores, otra tabla que almacena los archivos subidos a la aplicación y una última que guarda los comentarios (tweets) que los usuarios inserten en cada archivo.

Su diagrama sería el de la figura 3.4 que se muestra a continuación:

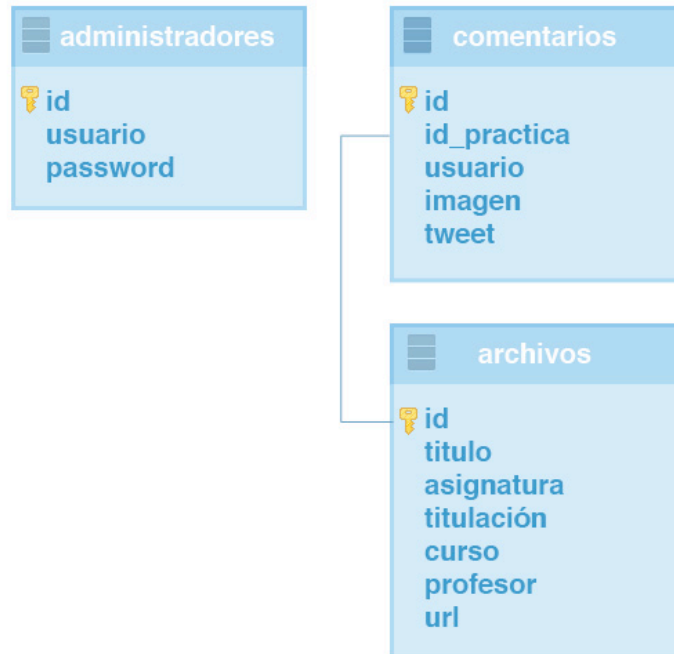


Figura 3.4: Arquitectura de la base de datos

La tabla de *administradores*, recoge una lista de nombres de usuario y contraseña de usuarios con rol de “administradores”. Este tipo de usuario será el único que pueda subir archivos a la aplicación, se trataría de personal docente, profesorado e investigadores que cargarían en el servidor el material de estudio en formato HTML.

Este material se recogerá en la tabla *archivos* que constará de los campos *profesor*, que indica el nombre del personal docente propietario del documento, *curso*, *asignatura* y *titulación* para ubicar el contexto del archivo subido, y la *ruta* del servidor donde ha quedado alojado el archivo.

Por último tenemos la tabla *comentarios*, dónde se almacenan todos los comentarios de los usuarios de Twitter que utilicen la aplicación dejen en las páginas de las prácticas sus tweets de dudas, preguntas, aclaraciones con el fin de dar uso a la aplicación. Esta tabla contendrá los campos *id_practica* que identifica que el comentario pertenece a una

práctica/apunte determinada y está relacionada con el campo *id* de la tabla *archivos*, el *comentario* en sí con su texto, además de la imagen del usuario en su cuenta de Twitter para identificarle y el nombre del usuario.

3.3. API de Twitter

Twitter ofrece a los desarrolladores una completa API basada en REST con la que se pueden implementar prácticamente todas las funcionalidades que ofrece su aplicación web. Por ejemplo enviar actualizaciones de estado, ver nuestro *timeline*, visualizar los *trending topics*, etc.[16]

El Timeline es nuestra página principal de Twitter, donde están los últimos tweets de la gente que seguimos. Un Trending Topic es una manera de saber cuál es la palabra o el tema más relevante del momento en la red. Si muchos usuarios están hablando sobre un determinado tema de actualidad, aparecerá como uno de los temas del momento, normalmente están clasificados en Global, por países y en ocasiones por ciudades. Un Retweet o RT es cuando un usuario comparte un tweet de otro usuario, lo repite para que lo puedan leer más usuarios y se extienda por la red.

3.3.1. Librería TwitterOAuth en PHP

Para el desarrollo de este proyecto, necesitamos una parte *backend* que gestione las consultas al API de Twitter, a la base de datos, etc.. Para esta parte he optado por PHP por ser un lenguaje muy utilizado en la red y por tanto muy probado, libre y con una gran documentación disponible.

Hemos optado por una librería PHP ya desarrollada por Abraham[17] que implementa todos las peticiones GET y POST del API REST de Twitter. Gracias a ella podemos obviar la programación que realice estas llamadas y recoja los datos (en XML o JSON, en Apuntweets utilizamos JSON), procesarlos, etc. Simplemente tendremos que indicar si es una petición de tipo GET o POST y a continuación, en texto, escribir la llamada REST. Esto nos devolverá un objeto que contendrá todos los datos que devolvería la llamada si se hiciera desde el navegador, pudiendo acceder a ellos como un array.

A continuación mostramos algunos ejemplos del funcionamiento de esta librería con métodos que hemos usado en nuestra aplicación. Uno de ellos es el mecanismo de login, para autenticarse en el sitio web o aplicación como usuario de Twitter y así poder acceder a la información que el API nos permite (como el avatar, nombre, últimos tweets, etc). En la figura 3.5 podemos ver un diagrama que explica los pasos de autenticación. Twitter utiliza OAuth para ello como se ha explicado en el capítulo 2 de esta memoria.

La aplicación envía una petición a <http://api.twitter.com/oauth/authenticate>. Si el usuario está autenticado pasa al siguiente paso, si no, redirige a la página de Twitter para que el usuario se identifique con su cuenta de Twitter y permita acceso a la información. Si el usuario no había permitido a Twitter anteriormente el acceso, Twitter le preguntará para concederlo.

Cuando se permita el acceso y el usuario esté autenticado, Twitter lo redireccionará de vuelta a la aplicación con un `oauth_token` y un `oauth_verifier` con los que la aplicación web podrá acceder a sus datos. De esta forma la aplicación completa el proceso de OAuth.

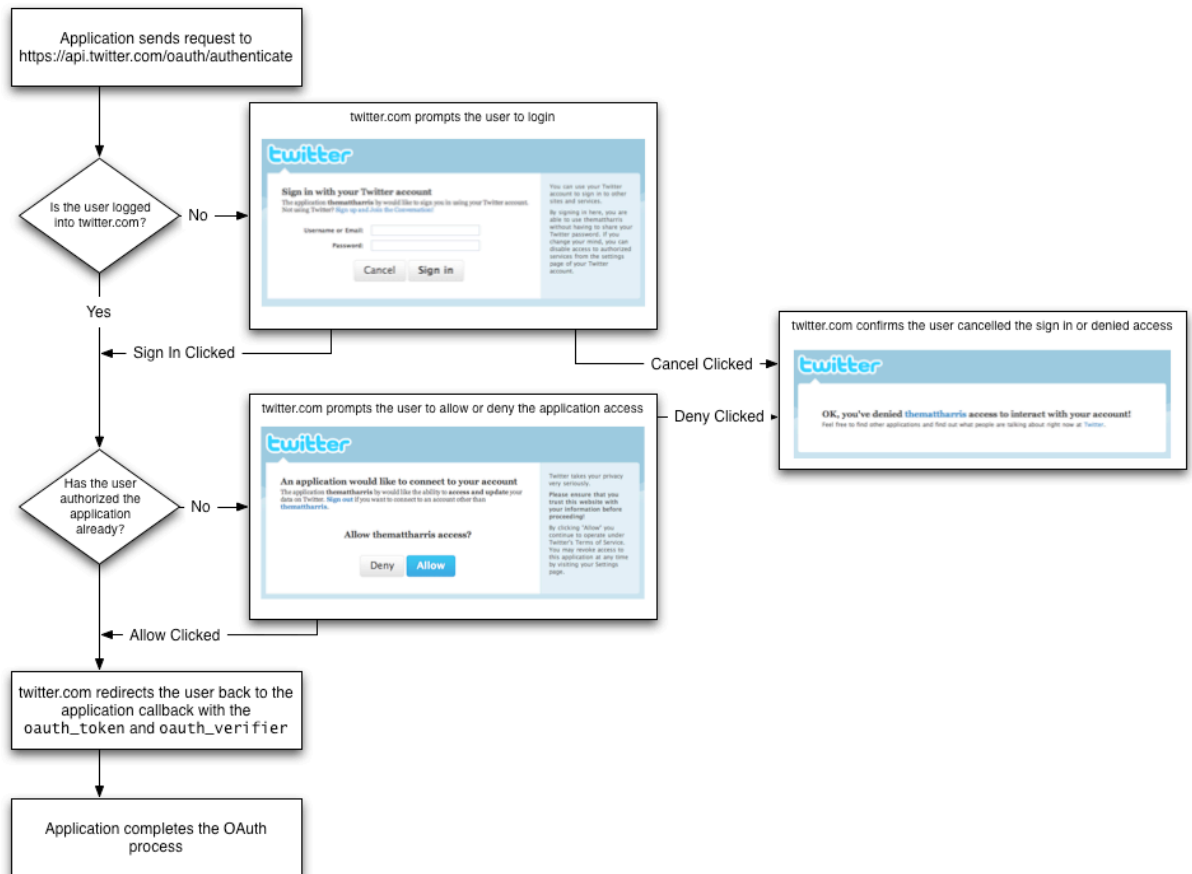


Figura 3.5: Autorización de Twitter en la aplicación (Extraída de [18])

3.4. Interfaz de usuario

La interfaz web son los elementos gráficos y estructurales que permiten al usuario acceder a los contenidos, navegar e interactuar. Para lograr que un usuario se sienta cómodo, permanezca en la página y vuelva más adelante, el diseño de la interfaz es importante.

La facilidad y la comodidad con que los usuarios acceden a los servicios que brinda una web está fundado en dos principios fundamentales para el desarrollo de una interfaz efectiva: la **simplicidad** y la **coherencia**.

- La **simplicidad** con que se desarrolle la interfaz es crucial para determinar que un usuario se sienta satisfecho y desee regresar a un sitio. El hecho de que una persona tenga que realizar una extensa navegación por el sitio para encontrar lo que busca es totalmente contraproducente. Por el contrario, si el sitio dispone de herramientas que permitan acceder rápidamente a aquello que necesita, seguramente volverá.

La simplicidad está dada por varios factores a tener en cuenta. El primer concepto importante es que los elementos gráficos o textuales que componen la interfaz deben ser claros y de fácil identificación. Por ejemplo, en “Apuntweets” se mantiene todo el diseño de la cabecera y pie de página en todas las páginas de la aplicación. De igual manera, la sección correspondiente al visor de apuntes es igual para todos, sea el material que sea el que se esté visualizando. Tan solo cambia el contenido pero la estructura visual y gráfica es la misma.

- También es importante que se guarde **coherencia** entre los diferentes elementos que componen la aplicación o sitio web, como los títulos, subtítulos, enlaces, menús, etc... Todos los elementos que permiten al usuario identificar y navegar deben ser coherentes con el cometido que desempeñan, de forma que la comprensión y búsqueda de los contenidos sean accesibles por el usuario sin que tenga la necesidad de realizar complejos razonamientos.

En “Apuntweets” hemos ordenado los apuntes por asignatura de tal manera que con pocos clicks se pueda acceder al contenido que resulta interesante para el usuario que es el material educativo sobre el que podrá comentar y preguntar dudas. De igual manera se guarda el mismo estilo para los elementos gráficos de forma que sea fácil reconocer su función.

A la hora de abordar el diseño de la interfaz de la aplicación se ha intentado que fuera una aplicación sencilla de fácil uso tanto para los estudiantes como para el profesorado, permitiendo que sin necesidad de tener conocimientos técnicos, su manejo y uso sean intuitivos y accesibles.

3.4.1. Diseño orientado al profesor.

Para que la aplicación sea cómoda y accesible al profesor hemos diseñado y programado la aplicación de tal manera que el material que ya posee el profesor en sus páginas web o de cualquier otra manera online, pueda ser utilizado en “Apuntweets” sin necesidad de tener que reescribirlos o adaptarlos. Además, no hace falta que tenga el archivo que quiere compartir, con introducir en el formulario la URL donde está alojado el material, la aplicación lo incrusta en el visor de apuntes de manera automática.

Sube tus apuntes/prácticas de clase

URL del apunte/prácti						
Título de la práctica/a	Asignatura	Titulación	Curso	Profesor	Twitter del Profesor	
<input type="button" value="Subir Archivo"/>						

Figura 3.6: Formulario de carga de material educativo en la aplicación.

3.4.2. Diseño orientado al estudiante.

Para hacer más cómoda la experiencia de usuario al alumno, el visor de apuntes/material educativo se ha diseñado de tal forma que la mayor parte de la pantalla esté ocupada por el material en sí para que su lectura sea fácil y cómoda. En la parte derecha de la pantalla se ha habilitado la sección de comentarios de los usuarios, estructurada como una pila y el formulario para dejar nuevos comentarios. Esto lo podemos ver en las figuras 3.7 y 3.8

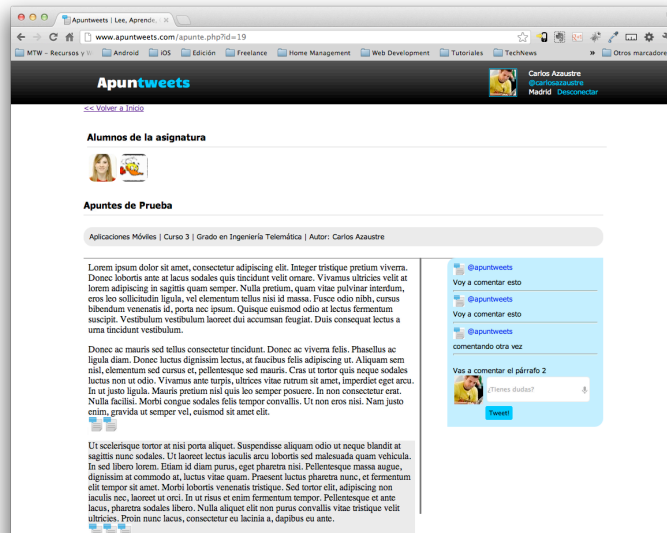


Figura 3.7. Visor de apuntes y columna de comentarios (1)

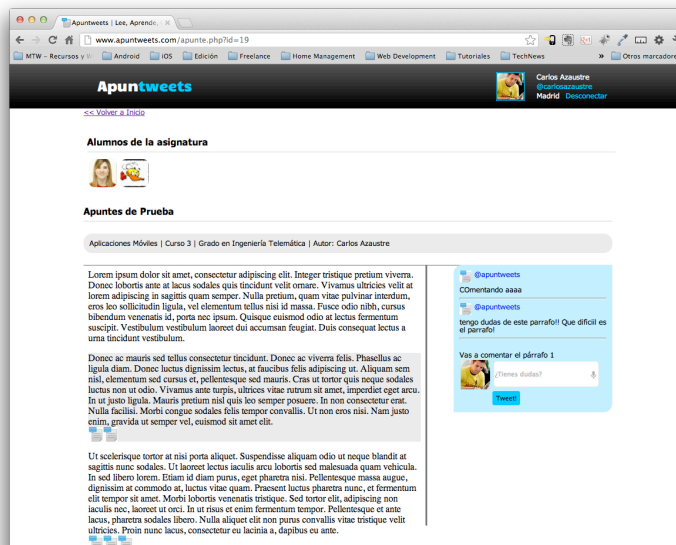


Figura 3.8. Visor de apuntes y columna de comentarios (2)

Como se ve en las anteriores figuras (3.7 y 3.8) esta sección se actualiza dinámicamente dependiendo del párrafo en el que situemos el cursor, de tal manera que aparecerán únicamente los comentarios del

párrafo en el que hagamos click, Así no se satura la página de información, es más limpia y clara para el usuario y todo es más ordenado.

Para saber si un párrafo tiene comentarios o no, se incluyen los avatares en miniatura de los usuarios que hayan comentado, incrustados a continuación del párrafo. Lo podemos ver en la figura 3.9

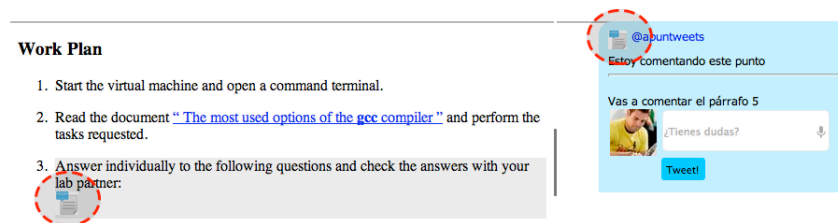


Figura 3.9. Párrafo comentado

3.5. Desarrollo de la aplicación

3.5.1. Registro de la aplicación en Twitter

Para poder utilizar el API de Twitter en nuestra aplicación debemos registrarla en Twitter, para ello debemos tener un hosting público donde estará subida y accesible nuestra aplicación, ya que si no algunas funcionalidades que nos ofrece Twitter (como la autorización via OAuth, explicada en el anterior apartado) no funcionarán (no se pueden probar en <http://localhost/>). En nuestro caso hemos usado un hosting que ya tenía contratado antes de la realización de este proyecto. De todas formas cualquier alojamiento web que tenga un servidor Apache corriendo, PHP y MySQL configurado nos serviría. Incluso si tenemos un servidor propio con una IP pública.

Una vez que cumplimos estos requisitos, tenemos que registrar nuestra aplicación web como aplicación de Twitter. Para ello debemos acceder desde nuestro navegador a la url <http://dev.twitter.com/apps> e ingresar con

nuestra cuenta de usuario de Twitter. Tras ingresar, se nos invita a crear una aplicación:

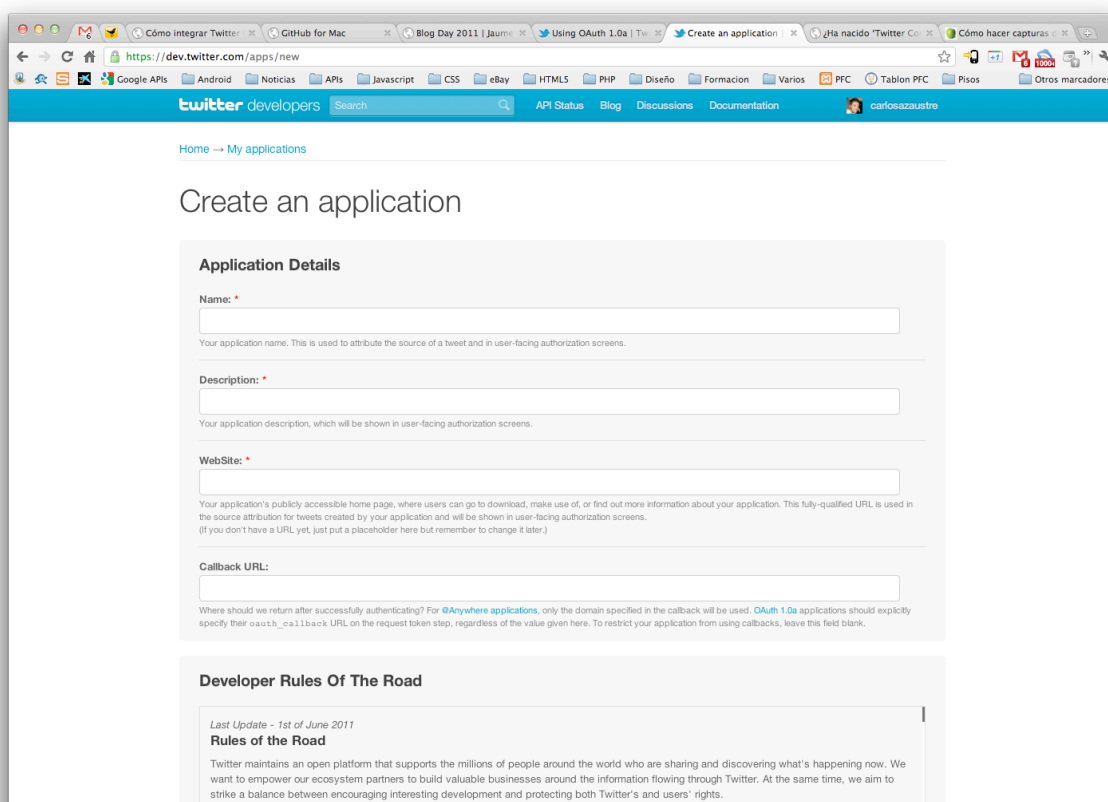


Figura 3.10: Creación de una aplicación en Twitter

Los datos a introducir en el formulario son los siguientes:

- **Name:** “Apuntweets” es el nombre que damos a nuestra aplicación. Cuando un usuario “Twitter” desde ella, el campo “Source” del Tweet llevará este nombre.
- **Description:** una pequeña descripción de la aplicación web. Aparecerá en la página de autenticación de OAuth cuando el usuario vaya a ingresar.

- **Website:** Una URL pública donde está alojada la aplicación y donde los usuarios pueden usar y encontrar más información acerca de ella.
- **Application Icon:** Un icono descriptivo de la aplicación que nos permita diferenciarla de otras que tengamos y que sea reconocible para el usuario.
- **Application-Type:** Tenemos 3 opciones: “Read Only” con la que solo podríamos leer tweets, timelines etc... pero no realizar peticiones tipo POST. “Read & Write” nos permite escribir tweets y leerlos, es la que usaremos para nuestra aplicación. “Read, Write and Access direct messages” nos permite los mismos permisos que la anterior opción, añadiendo el acceso a los mensajes privados, que por ahora no nos interesa para nuestra aplicación
- **Callback URL:** Este campo es importante, aquí debemos escribir la url de la página donde Twitter irá cuando el usuario se haya autenticado. Ahí es donde aparecerá el nombre de usuario, sus tweets, etc. todo lo que queramos mostrar del usuario haciendo uso del API de Twitter.

Tras el registro, Twitter nos devuelve la información que hemos registrado y nos proporciona dos pares de claves para identificar unívocamente la aplicación y aportarle seguridad. Estas claves son *Consumer Key*, *Consumer Secret* y las de autenticación vía OAuth que son *Access Token* y *Access Token Secret*. Estos datos podemos verlos en la figura 3.11 de a continuación.

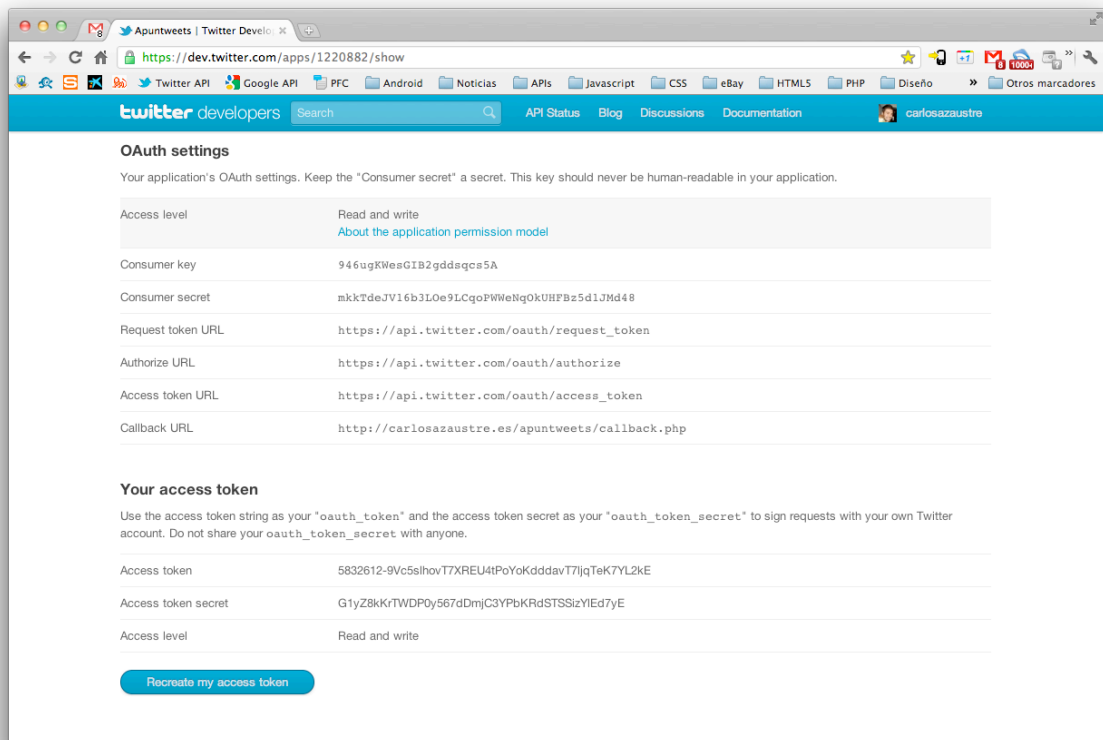


Figura 3.11: Configuración OAuth

Estas claves son importantes y nunca deben ir “en claro” en la aplicación, ya que pueden suplantar nuestra identidad y publicar Tweets bajo nuestro usuario y aplicación sin permiso.

Con estos pasos nuestra aplicación queda correctamente registrada en Twitter y podemos utilizar su API para acceder a los datos que necesitemos.

3.5.2. Diseño de la interfaz de usuario

En la figura 3.12 se muestra la interfaz de la aplicación cuando accedemos a ella. Consta del logo y un botón, proporcionado por el API de Twitter para autenticarse con una cuenta de la red social mediante el

mecanismo de OAuth. Se invita al usuario a que se cree una cuenta si no la posee aún, a través de una URL.

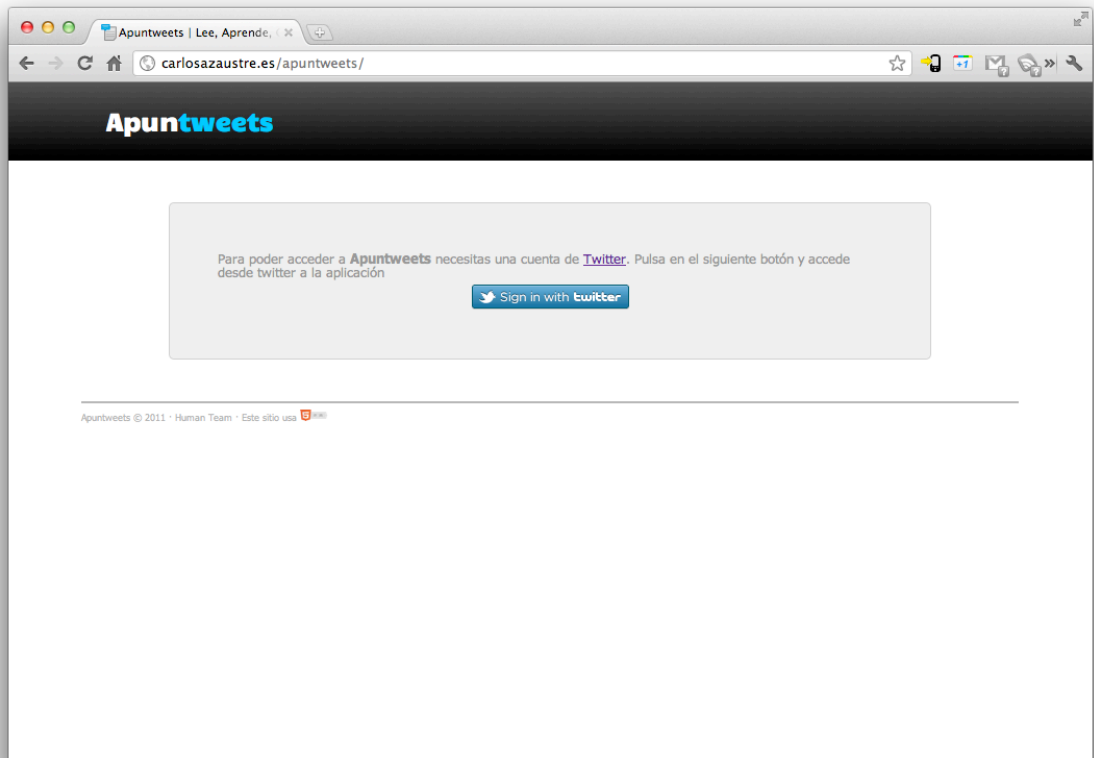


Figura 3.12: Pantalla de Inicio

Al pulsar en el botón de login, nos redirecciona a la página de autorización de Twitter para introducir la cuenta de usuario y su contraseña (Figura 3.13). De esta manera se autoriza a la aplicación web acceder a los datos de usuario de Twitter, tal y como refleja la figura 3.14. Los datos a los que tendrá acceso son:

- Poder leer tweets de la cronología del usuario
- Ver tus seguidores y poder seguir a nuevas personas
- Actualizar el perfil

- Publicar tweets por el usuario

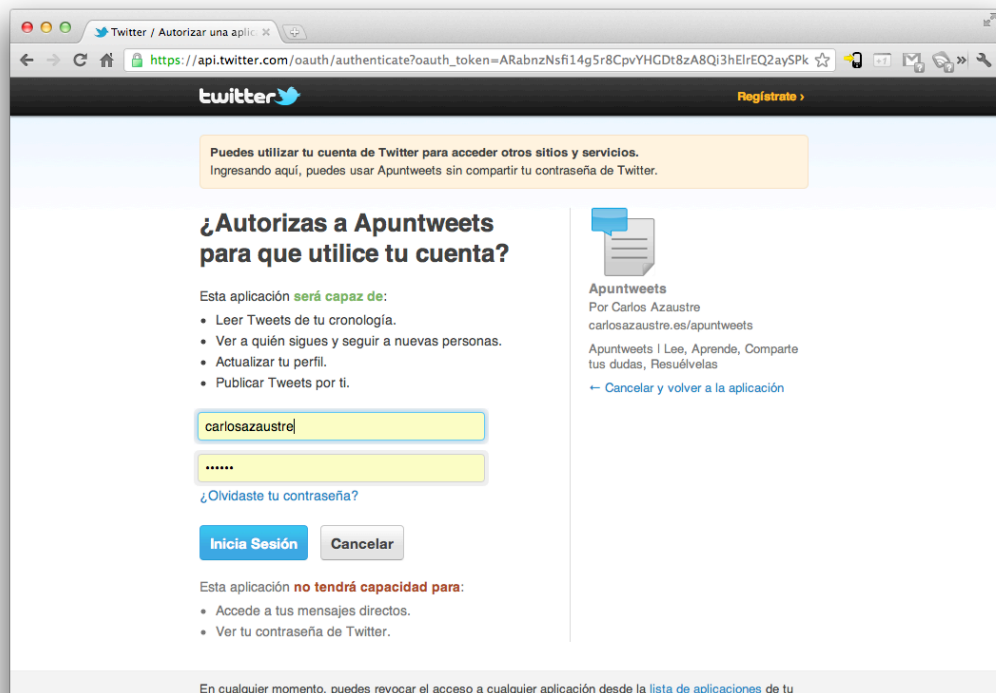


Figura 3.13: Petición de autorización a Twitter

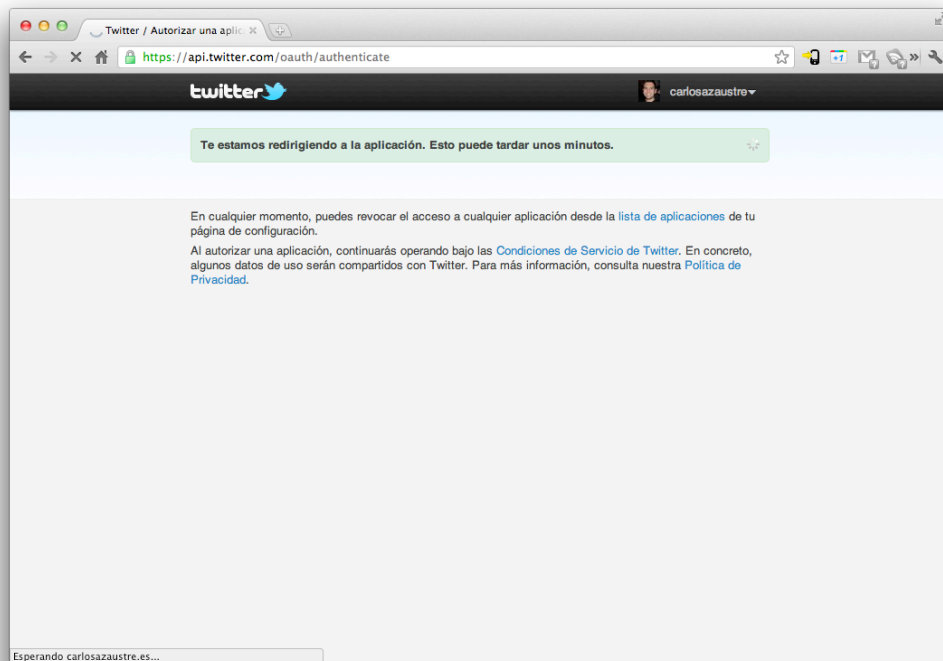
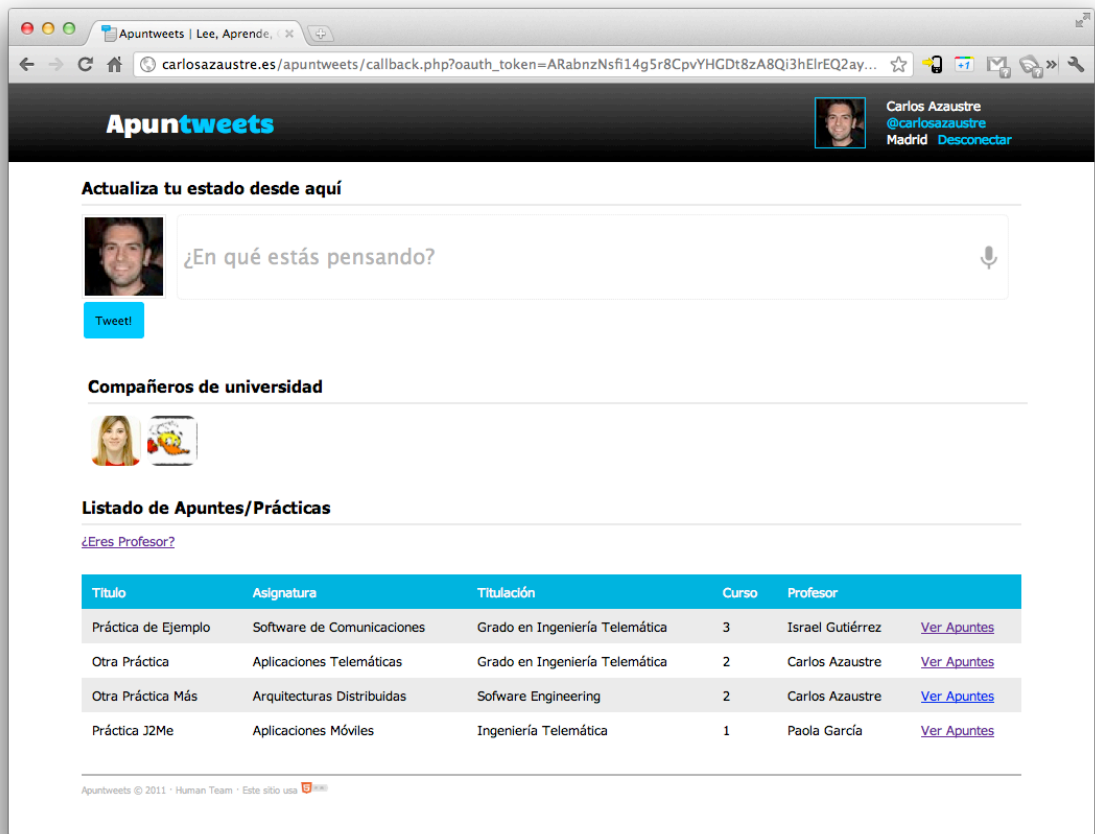


Figura 3.14: Confirmando la autenticación

Una vez autorizada la aplicación accedemos a la página de retorno que será la principal de toda la aplicación. Como se puede ver en la imagen de la figura 3.15, en la parte superior derecha se muestra la imagen del usuario junto a su nombre real, nombre de usuario y ubicación. Se ofrece también la posibilidad de desconectarse.

Desde la aplicación se ofrece al usuario que pueda actualizar su estado en Twitter, como complemento a la aplicación, esto se ubica en la parte superior. Seguidamente se muestra una lista de imágenes de usuario enlazadas a sus perfiles de Twitter, que corresponden a la lista “Universidad” que tenga el usuario configurada. En caso de no tenerla se mostrará un texto donde se invita a crear una para añadir usuarios cercanos con el fin de tenerlos más a mano.



En el caso de que el usuario tenga el rol de “administrador” existirá un módulo para la subida de archivos como podemos ver en la imagen la figura 3.16.

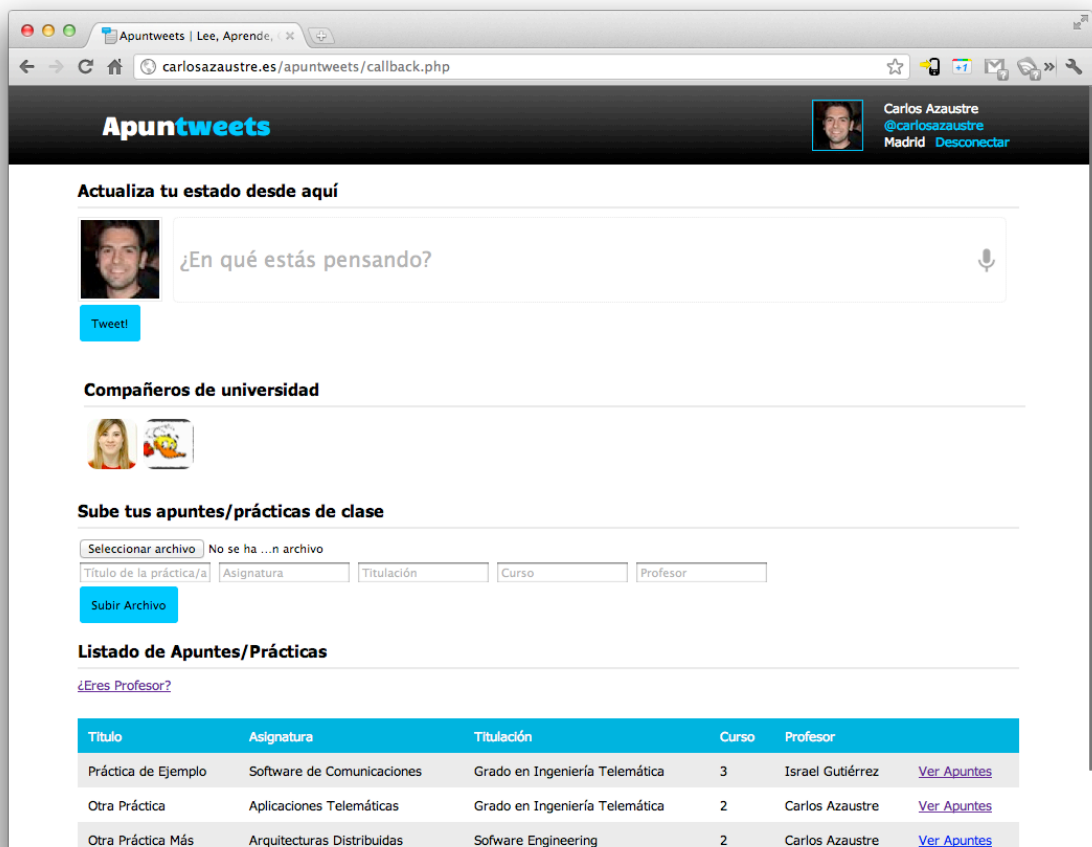


Figura 3.16: Vista principal de la aplicación con rol de administrador

3.5.3. Desarrollo de módulos funcionales

3.5.3.1. Gestión de Usuarios

En la aplicación tenemos dos formas diferentes de autenticarnos. La principal es con la cuenta de Twitter, donde todo el peso de la autenticación cae sobre Twitter ya que no guardaremos ningún registro del usuario en nuestra base de datos. La otra forma es mediante formulario HTML con usuario y contraseña. Esta forma será para los

usuarios con un rol de administrador y serán los únicos con privilegios para poder cargar archivos (Prácticas/Apuntes) en la aplicación.

- **Mediante cuenta de Twitter.**

Desde la página principal se pide que el usuario se autentifique con su cuenta de Twitter. Gracias al mecanismo OAuth obtenemos un token con los datos del usuario y podemos utilizar el API para realizar diversos métodos. Empleando la librería `twitterOAuth` escrita en PHP, podemos acceder al nombre de usuario, nombre real y foto de perfil del usuario autenticado de una manera sencilla. El objeto usuario queda guardado en la sesión PHP y mientras no se desconecte tendremos siempre los datos disponibles

- **Mediante formulario de usuario administrador.**

Es un formulario HTML donde se ingresa el nombre y la contraseña y se devuelve a la página principal, con la diferencia de que sólo estos usuarios podrán ver un módulo de subida de archivos y por tanto subir archivos. El formulario realiza una consulta a la base de datos para verificar que el usuario existe y la contraseña es correcta y si es así poder acceder.

3.5.3.2. Vista principal

Desde la vista principal, la página `callback.php` muestra los datos del usuario autenticado con Twitter y ofrece poder actualizar su estado como opción complementaria a la aplicación. Esto se consigue mediante Javascript y llamadas asíncronas dentro del código y usando el método de actualización de estado de la librería PHP.

Se realiza una consulta a la base de datos, en concreto a la tabla `archivos` para mostrar todos los apuntes y/o prácticas que hay publicados. Se muestran el título del archivo, su autor, asignatura a la que pertenece, titulación y curso, además de un enlace a la vista donde estará accesible el fichero para poderlo comentar.

Si el usuario está autenticado como usuario administrador, tenemos una variable en la sesión que comprueba si ese rol está activado o no para el

usuario. En caso de que sí, se muestra un formulario para poder cargar documentos al servidor y cumplimentarlo con los datos de título, asignatura, curso, titulación y autor para poder almacenarlos en la base de datos y acceder a ellos posteriormente. El script del servidor se encarga de hacer una copia del documento accediendo a la URL que el usuario docente le proporciona y subirla al servidor.

3.5.3.3. Vista de archivos

Si queremos ver un archivo en concreto, pulsamos en el enlace de la tabla de archivos de la vista anterior y nos lleva a esta nueva vista. Aquí mostramos los datos referentes al archivo como son su título, nombre del autor, asignatura, titulación y curso y seguidamente incrustado gracias a PHP, el HTML del archivo copiado/subido desde la vista principal.

Mediante Javascript, recorreremos el archivo incrustado buscando los diferentes párrafos que se encuentran en él. Se les asocia un identificador a cada uno para cuando sean comentados puedan ser almacenados con la referencia a ese párrafo, para su posterior lectura.

A su lado derecho se incluirá el módulo de comentarios que se explica en el siguiente apartado.

3.5.3.4. Módulo de comentarios.

En la vista de archivos dispondremos de este módulo en el lado derecho de la página. Aquí se pueden ver los comentarios que han realizado los usuarios referentes al archivo que estamos viendo. Estos comentarios, además de verse aquí en orden cronológico, son publicados en Twitter en el momento que se realizan.

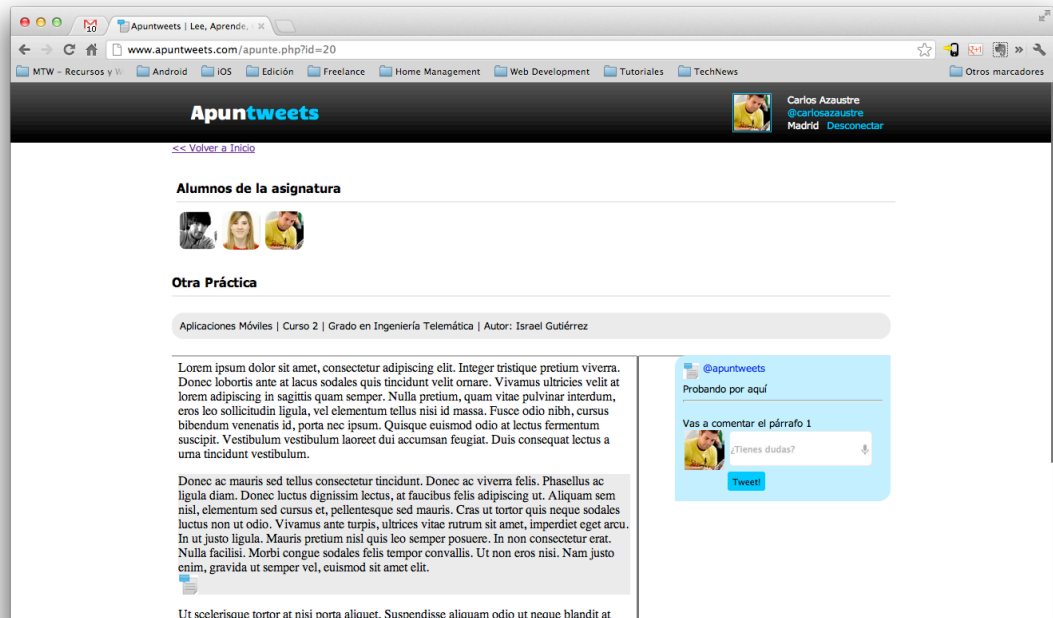


Figura 3.17 Vista de archivo con comentarios.

Para ello en el servidor tenemos función que recoge el texto del “estado” del usuario en la aplicación, lo guarda en la base de datos asociándolo al identificador del archivo y del párrafo donde ha comentado e inmediatamente después se emplea la librería PHP *twitterOAuth* para actualizar ese estado en Twitter.

Capítulo 4

Pruebas

Durante el desarrollo de este proyecto se han realizado diferentes pruebas sobre la aplicación web. En las siguientes secciones se describe en qué consiste cada escenario de prueba y los resultados obtenidos.

4.1. Escenarios de pruebas

En este proyecto se han empleado tres escenarios de pruebas para la aplicación. A continuación se describen en qué consiste cada uno de ellos.

1. Pruebas en local.

Las pruebas en local son aquellas que realizamos en nuestro propio ordenador, instalando un servidor en él y nos permiten implementar la aplicación y depurarla de una manera más rápida y controlada. No es necesario disponer de un servidor alojado en otra máquina. Para estas pruebas hemos utilizado el software XAMPP [19] que se compone de un servidor web Apache, una base de datos MySQL y configuración PHP. Este software puede correr tanto en los sistemas operativos Windows, Linux y MAC OS.

A continuación detallamos los problemas encontrados durante el proceso:

- Para utilizar OAuth y verificar las credenciales de acceso con cuenta de Twitter se necesita que el servidor donde se realiza esta autenticación, disponga de una IP pública. En nuestro caso solo disponemos de una IP privada por lo que estas verificaciones necesitábamos hacerlas en el siguiente entorno de pruebas.
- Otro problema que nos encontramos en las pruebas en local fue la falta de la extensión cURL de PHP[20]. Es un conjunto de librerías entre las que se incluye libcurl, que permite conectarse y comunicarse con diferentes tipos de servidores y diferentes tipos de protocolos. Actualmente, libcurl admite los protocolos http, https, ftp, gopher, telnet, dict, file y ldap. Libcurl también admite certificados HTTPS, http, POST, http, PUT, subidas mediante FTP, subidas basadas en formularios http, proxies, cookies y autenticación usuario+contraseña.

La falta de esta extensión provocaba que no apareciese el botón de login con Twitter y en su lugar se mostrara este mensaje:

```
Fatal error: Call to undefined function
curl_init() in C:xampphtdocs witter witter.php
on line 186
```

Solucionar este inconveniente fue muy sencillo ya que la extensión está instalada en el software XAMPP, lo único que no está habilitada. Para confirmarlo hay que fijarse en el directorio de extensiones de PHP donde está instalado XAMPP (/xampp/php/ext) y observar que está la librería en concreto (php_curl.dll)

Una vez confirmado que está instalada, debemos activarla. Para ello hay que editar el fichero php.ini que se encuentra en el

directorio de PHP y simplemente quitar el punto y coma “;” que hay antes de la línea que carga la librería.

```
;extension=php_curl.dll
```

Por último hay que guardar el fichero php.ini y reiniciar el servidor Apache para que los cambios surtan efecto. Después de eso la librería cURL para PHP se cargará sin problemas.

Otro problema lo tuvimos a la hora de programar el visor de apuntes. Apostamos por programarlo de manera que a través de un iframe de HTML se visualizara el contenido de una URL externa a nuestro servidor. De esta manera el usuario que subía archivos no tenía que preocuparse de cargar el archivo sino simplemente indicar la URL donde está alojado ese archivo. Esto impedía que a través de Javascript pudiésemos acceder al contenido del iframe y modificarlo, ya que por cuestiones de seguridad Javascript no permite modificar nada que no esté alojado desde la máquina donde está ejecutándose.

La solución que implementamos fue algo laboriosa pero nos permitió realizar lo que buscábamos. Para poder acceder al contenido del iframe necesitábamos tener una copia del archivo alojado en una URL externa en nuestro servidor. Para ello programamos el siguiente script PHP que copia el archivo remoto de una URL ajena al servidor y guarda la copia en un directorio local dentro de nuestro servidor.

```
//COPIA DE URL A ARCHIVO LOCAL
$ch = curl_init($url);
$prefijo = substr(md5(uniqid(rand())), 0,6);
$fp = fopen("archivo".$prefijo.".html", "w");
$archivoURL = "archivo".$prefijo.".html";

curl_setopt($ch, CURLOPT_FILE, $fp);
curl_setopt($ch, CURLOPT_HEADER, 0);

curl_exec($ch);
curl_close($ch);
```

```

fclose($fp);

//Una vez copiado lo registramos en la base de
datos
$conectar = mysql_connect($host, $usuario,
$password);

if(!$conectar) {
    die("No pudo conectarse: "
.mysql_error());
}
mysql_select_db($basedatos, $conectar);

$query = "INSERT INTO archivos
VALUES(0, '$titulo.', '$signatura.', '$ti
tulacion.', '$curso.', '$profesor.', '$ar
chivoURL.', '$twitter.');";
mysql_query($query);

```

Posteriormente, cuando se carga el interior de iframe, se llama a la función cargaIframe() que recorre el árbol del DOM y selecciona los párrafos que va encontrando con la intención de añadirles los atributos necesarios para que escuchen eventos y así poder seleccionarlos y operar con ellos.

```

function cargaIframe(){
    var seccionApunte = $("#apunte").contents();
    var parrafos = [];
    parrafos = seccionApunte.find("p");

    for (var i=0; i<parrafos.length; i++){
        parrafos[i].className = "resaltado";
        parrafos[i].setAttribute('ID', i);
        parrafos[i].addEventListener("click",
seleccionarParrafo);

        parrafos[i].addEventListener("onMouseOver",
cargaFotos(i));

        var elemFoto =
document.createElement("span");

```

```

        elemFoto.setAttribute('ID',
"frameFoto"+i);

        parrafos[i].appendChild(elemFoto);
    };

    for(var i=0; i<parrafos.length; i++){

        $("#apunte").contents().find("#frameFoto"+i).
load("listadocomentarios_foto.php?archivo=<?php
echo $_GET['id']; ?>&parrafo="+i);
    }

}

```

De esta manera solucionamos el problema y ya podemos acceder al contenido del archivo incrustado en el iframe.

2. Pruebas en servidor.

El siguiente paso es dar un salto y dejar el entorno local para pasar a un servidor real alojado en otra máquina. Para estas pruebas hemos utilizado un alojamiento web [21] con una configuración igual a la utilizada en el servidor instalado en local (servidor web Apache, base de datos MySQL y configuración PHP). Como en el anterior escenario de pruebas, nos encontramos con algunos problemas que detallamos a continuación:

- El primer servicio de alojamiento web que utilizamos fue Freehostia[22], un servicio de hosting gratuito que tenía la configuración LAMP (Linux + Apache + MySQL + PHP) que necesitábamos. Pero en el momento de probar la aplicación una vez desplegada en el servidor fue el mismo que en el anterior entorno de pruebas, fallaba la extensión cURL porque no estaba configurada en el hosting. Lamentablemente no podíamos acceder a la configuración e instalarla manualmente al tratarse de un hosting gratuito por lo que la decisión tomada fue cambiar de alojamiento a ConfigBox[21] un servicio de pago pero económico que nos permitió disponer de todas las configuraciones necesarias para que la aplicación web funcionase correctamente.

4.2. Pruebas en local

Las pruebas más relevantes realizadas para la aplicación “Apuntweets” en un entorno local se recogen en la tabla 4.1,

Prueba realizada	Resultado	Observaciones
Acceso con cuenta de Twitter	ERROR	No se puede realizar esta prueba en local, porque OAuth necesita una dirección IP pública para autenticarse
Acceso como usuario “Profesor”	OK	Se prueba que el usuario+contraseña accede correctamente a la vista de personal docente de la aplicación
Enviar Tweet desde pantalla principal	ERROR	Al igual que la primera prueba, todo lo relacionado con el acceso a twitter no se puede probar si no se tiene IP pública
Cargar material docente	OK	Se introducen los datos correspondientes al material que se desea subir, en el formulario web y la dirección URL del material propriadamente dicho, se comprueba que los datos son almacenados en la base de datos y que el archivo de la URL externa se ha copiado en

		un directorio del servidor local.
Visualización de material docente	OK	En el visor de material se comprueba que se carga correctamente dentro del iframe contenedor
Selección de párrafos	OK	Se comprueba que al hacer <i>click</i> en los párrafos del material cargado, estos se sombream y el formulario de comentarios de la derecha va cambiando dependiendo del párrafo en el que nos encontremos.
Comentar párrafos	ERROR	Al comentar un párrafo determinado, se observa que se almacena el comentario en la base de datos junto con el nombre de usuario pero este no es enviado a Twitter debido al problema de OAuth con las direcciones IP privadas.
Visualizar párrafos	OK	Al hacer click sobre un párrafo comentado se observa que en la parte derecha se carga el comentario que se había realizado con anterioridad.

Tabla 4.1. Pruebas de funcionamiento en local

4.3. Pruebas en servidor

Las pruebas más relevantes realizadas para la aplicación “Apuntweets” en un entorno local se recogen en la tabla 4.2,

Prueba realizada	Resultado	Observaciones
Acceso con cuenta de Twitter	OK	Se accede mediante OAuth a la aplicación con la cuenta de usuario de Twitter. Se comprueba que en la página principal aparecen el avatar, nombre de usuario y ubicación del usuario. Estos datos son tomados de la cuenta de Twitter.
Acceso como usuario “Profesor”	OK	Se prueba que el usuario+contraseña accede correctamente a la vista de personal docente de la aplicación
Enviar Tweet desde pantalla principal	OK	Se prueba en mandar un mensaje desde la pantalla principal de la aplicación hacia la red Twitter
Cargar material docente	OK	Se introducen los datos correspondientes al material que se desea subir, en el formulario web y la dirección URL del material propiamente dicho, se comprueba que los datos son almacenados en la

		base de datos y que el archivo de la URL externa se ha copiado en un directorio del servidor local.
Visualización de material docente	OK	En el visor de material se comprueba que se carga correctamente dentro del iframe contenedor
Selección de párrafos	OK	Se comprueba que al clickar en los párrafos del material cargado, estos se sombream y el formulario de comentarios de la derecha va cambiando dependiendo del párrafo en el que nos encontremos.
Comentar párrafos	OK	Se comprueba que al comentar un párrafo determinado, se almacene en la base de datos junto con el nombre de usuario y que también es enviado a Twitter.
Visualizar párrafos	OK	Al hacer <i>click</i> sobre un párrafo comentado se observa que en la parte derecha se carga el comentario que se había realizado con anterioridad.

Tabla 4.2. Pruebas de funcionamiento en el servidor

Capítulo 5

Historia del proyecto

Este proyecto surge a partir de un pequeño trabajo de investigación sobre HTML5 sobre carga de documentos en internet y evolucionó hacia la plataforma de materiales educativos que es actualmente. Durante los meses de Enero a Septiembre de 2011 se investigó sobre este uso de tecnologías hasta que en Septiembre de 2011 se empezó a desarrollar el proyecto actual “Apuntweets”.

5.1. Distribución temporal

Este proyecto se ha desarrollado en un tiempo aproximado de 1 año, desde septiembre de 2011 hasta Agosto de 2012. A continuación se describen brevemente las fases del proyecto junto al tiempo dedicado a cada una de ellas. En las figuras 5.1 y 5.2 se muestra y explica el diagrama de Gantt y las fases del proyecto.

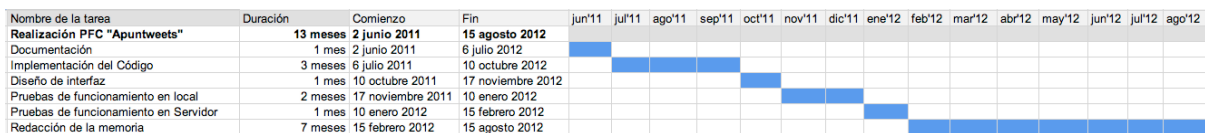


Figura 5.1 Diagrama de Gantt de la realización del proyecto

Nombre de la tarea	Duración	Comienzo	Fin
Realización PFC "Apuntweets"	13 meses	2 junio 2011	15 agosto 2012
Documentación	1 mes	2 junio 2011	6 julio 2012
Implementación del Código	3 meses	6 julio 2011	10 octubre 2012
Diseño de interfaz	1 mes	10 octubre 2011	17 noviembre 2012
Pruebas de funcionamiento en local	2 meses	17 noviembre 2011	10 enero 2012
Pruebas de funcionamiento en Servidor	1 mes	10 enero 2012	15 febrero 2012
Redacción de la memoria	7 meses	15 febrero 2012	15 agosto 2012

Figura 5.2 Fases de desarrollo en el Diagrama de Gantt

5.1.1. Documentación (1 mes)

El primer paso de este proyecto es la documentación en la cual se adquieren los conocimientos en los siguientes campos

- API de Twitter
- Tecnología OAuth para autenticación
- Lenguaje Javascript y su extensión jQuery
- Estructura del lenguaje HTML5

5.1.2. Implementación del código (3 meses)

En esta fase, además de la programación como tal del código, se realizó la búsqueda de la información necesaria para la implementación de las aplicaciones y la solución de problemas específicos que iban surgiendo: problemas con la autenticación de los usuarios de twitter en un servidor web que no soportase la librería cURL de PHP, la necesidad de hacer pruebas en un servidor no local, etc.

5.1.3. Diseño de la interfaz (1 mes)

En esta fase se realizó el diseño visual de la aplicación, consistente en una capa de presentación de los resultados obtenidos mediante las operaciones que realiza el código desarrollado

5.1.4. Pruebas de funcionamiento en local (2 meses)

En esta fase se realizaron todo tipo de pruebas para comprobar el correcto funcionamiento de la aplicación en los módulos de subida de archivos, login de usuarios administradores, marcado de los documentos, lectura y escritura de comentarios, etc.

No se pudo probar el funcionamiento del código basado en los métodos de la API de Twitter ya que para ello debe correr en una máquina externa y no local.

5.1.5. Pruebas de funcionamiento en Servidor (1 mes)

En esta fase se probó todo el código al completo, prestando más atención a la parte dependiente del API de Twitter que no pudimos probar en la fase anterior.

5.1.6. Redacción de la memoria (7 meses)

La realización de la memoria se lleva a cabo en un periodo aproximado de 7 meses teniendo en cuenta que no han sido 7 meses dedicados íntegros a la redacción si no que se ha realizado de forma simultánea con otro trabajo y estudios.

Durante el desarrollo del código y documentación de las tecnologías se redactaron algunas partes de la memoria aunque no está reflejado como tal en el diagrama de Gantt.

5.2. Presupuesto del proyecto

En este apartado se presenta el presupuesto del proyecto, estableciendo tanto el coste material como el del trabajo de las personas que han participado en su desarrollo.

5.2.1 Costes de personal

Los costes de personal incluyen los honorarios del Ingeniero Técnico de Telecomunicación en Telemática encargado del desarrollo del proyecto. La duración de este proyecto ha sido de aproximadamente 14 meses, Aunque los 7 meses de redacción de la memoria no han sido de dedicación completa si no dedicaciones puntuales a lo largo de los 7 meses, podemos aproximar que si estimamos una dedicación de 4 horas diarias continuas, se tratarían de 1,5 meses útiles de redacción de la memoria, por tanto en lugar de 14 meses serían 8,5 meses útiles. Suponiendo 20 días laborables al mes y teniendo en cuenta 20 días laborables menos por motivos de dedicación no exclusiva en periodos como Navidad o verano, se obtiene un total de 170 días laborables, con una jornada laboral aproximada de cuatro horas diarias, la realización del proyecto ha requerido 680 horas.

Según el baremo orientativo del Colegio Oficial de Ingenieros Técnicos de Telecomunicación (COITT) en septiembre de 2009[23] los honorarios de un Ingeniero Técnico de Telecomunicación en Telemática eran de 62 euros la hora, por tanto el coste total ascendería a 42.160€. La siguiente tabla recoge este resultado:

Concepto	Horas	Honorarios	Importe
Ingeniero Técnico, Telecomunicación en Telemática	680	62€/hora	42.160€

Tabla 5.1: Costes de personal

Por supuesto esto sería un presupuesto del coste según los baremos de 2009. Pero esta cantidad es asumible por una empresa/inversor dependiendo del país que se trate. Debido a la crisis financiera que atraviesan los países europeos, el coste por hora de un ingeniero técnico en el mercado actual no corresponde con ese salario, habría que situarlo en sueldo de “programador junior” que rondaría los 20.000€ anuales brutos, siendo el coste por hora mucho menor.

5.2.2. Costes de material

Los materiales empleados durante la realización del proyecto han sido los siguientes:

- Un ordenador portátil con sistema operativo Mac OS X v10.7 (Lion) valorado en 1281 euros.[24]
- Un servicio de hosting con servidor apache, soporte PHP y base de datos MySQL por un valor de 20€ al año.[21]
- Conexión a internet durante la realización del proyecto. Ha sido necesaria para conseguir documentación y para probar la aplicación. Está valorada aproximadamente en 42€ al mes [25], que multiplicado por los meses de trabajo, supone un coste de 546 euros.

Teniendo en cuenta todos estos elementos, los costes de material se detallan en la siguiente tabla:

Concepto	Unidades	Precio/ud	Importe
Ordenador	1	1281€	1000€
Hosting	1	20€	20€
Conexión ADSL	1	42€/mes	546€
TOTAL			1847€

Tabla 5.2: Costes de material

5.2.3. Presupuesto total

El presupuesto final para la realización de este proyecto está formado por los costes de material y de personal presentados anteriormente. Como se observa en la siguiente tabla, el total asciende a 44.007€

Concepto	Importe
Costes de personal	42.160€
Costes de material	1.847€
TOTAL	44.007€

Tabla 5.3: Coste total

Capítulo 6

Conclusiones y trabajos futuros

6.1. Conclusiones

En este proyecto se ha realizado un estudio de la tecnología HTML5, APIs de terceros y su aplicación en el eLearning como plataforma de recursos docentes. Empleando tecnologías *Open Source*, hemos creado una aplicación lo suficientemente potente para ser empleada en el ámbito de una o varias asignaturas docentes.

En el momento de la redacción de esta memoria no se ha experimentado con ella en el aula, por lo que uno de los futuros trabajos a desarrollar sería el despliegue y uso de la aplicación en el ámbito docente, en primer lugar como versión Beta y a partir de los resultados obtenidos depurar y modificar lo necesario para que funcione correctamente.

6.2. Líneas futuras

Tras el desarrollo de la aplicación “Apuntweets” y pensando en proyectos futuros que complementen el proyecto desarrollado, a continuación se proponen algunas posibles líneas de trabajo futuro:

6.2.1. Ordenación de material según las preferencias del usuario.

En la aplicación “Apuntweets” sería muy útil que el usuario pudiese tener una página personal donde poder indicar cuáles son sus prácticas/apuntes destacados, ordenarlos por fecha, importancia, etc.

La aplicación podría personalizar la página principal de manera que el usuario tuviera en un primer vistazo sus apuntes abiertos recientes, últimos comentarios recibidos en apuntes que el haya comentado y también últimos comentarios enviados a la aplicación, por cualquier usuario.

Esta mejora se pensó cuando el proyecto inicial se estaba culminando y por motivos de tiempo no se pudo implementar en la primera versión de este proyecto.

6.2.2. Despliegue de la aplicación en el ámbito docente

Por motivos de tiempo no se ha podido experimentar en el aula con la aplicación y comprobar su funcionamiento a mediana escala. Sería interesante probarlo para comprobar que cumple las necesidades de los usuarios, tanto profesores como alumnos. Y tras su estudio, detectar las vulnerabilidades o necesidades que se produzcan para atajarlas, corregirlas y mejorar la aplicación.

6.2.3. Desarrollo de aplicación móvil nativa en Android/iOS.

Aunque la aplicación está diseñada siguiendo estándares del W3C y puede ser visualizada en cualquier dispositivo móvil, sería interesante desarrollar una aplicación nativa para los principales sistemas operativos móviles presentes en el mercado.

Es interesante por dos razones, una aprender el lenguaje de programación que se elija para desarrollarlo (Android o iOS) de manera que el futuro estudiante que lo amplíe pueda desarrollar su carrera en este ámbito; y por otro lado, las aplicaciones nativas hacen un mejor uso de los servicios que ofrecen los smartphones (teléfonos inteligentes) como son la cámara, la libreta de contactos, la ubicación, etc...

A continuación se muestran una serie de ventajas de las aplicaciones móviles frente a las aplicaciones web que se pueden usar para nuestro beneficio:

Como se ha dicho antes, las aplicaciones nativas, a priori, ofrecen una experiencia de usuario más completa al sacarle partido los servicios nativos de los teléfonos (cámara, contactos, ubicación, etc.)

Las aplicaciones nativas no necesitan necesariamente estar conectadas a internet para ser utilizadas. Al hacer uso de la funcionalidad se puede trabajar en modo offline cuando no hay conexión a internet. Sin embargo algunas aplicaciones pueden requerir una conexión a internet, dependiendo de la funcionalidad y los datos disponibles.

En términos de distribución, las aplicaciones nativas obtienen una buena visibilidad entre los consumidores debido a que son distribuidas a través de las tiendas de aplicaciones del sistema operativo que lleve el teléfono (App Store en el caso de iOS y Play Store, antiguo Android Market, en el caso de Android). Esto puede suponer un modelo de ingresos por las descargas de la aplicación.

Es importante tener en cuenta que si se quiere que los usuarios usen mayoritariamente la plataforma móvil, ésta debe de estar en el mayor número de plataformas móviles disponibles para asegurar la compatibilidad con la gama más amplia de teléfonos. Aunque como en todo, siempre hay unas plataformas que ocupan mayor cuota de mercado que otras y estas son Android, iOS y abriéndose hueco Windows Phone, como podemos ver en la figura 6.1.

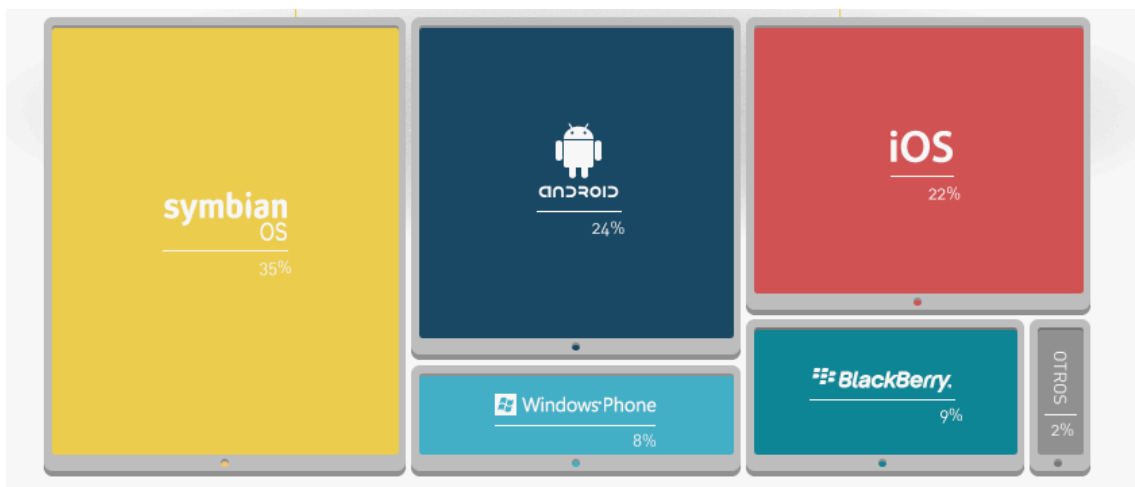


Figura 6.1: Principales sistemas operativos móviles en España (Extraída de [15])

Aunque esta imagen no corresponde con la realidad del uso de aplicaciones móviles en España. Ya que, aunque el sistema operativo de Nokia (Symbian) tiene tanta extensión, la mayoría de las descargas de aplicaciones son en dispositivos iOS y Android como podemos ver en la figura 6.2.

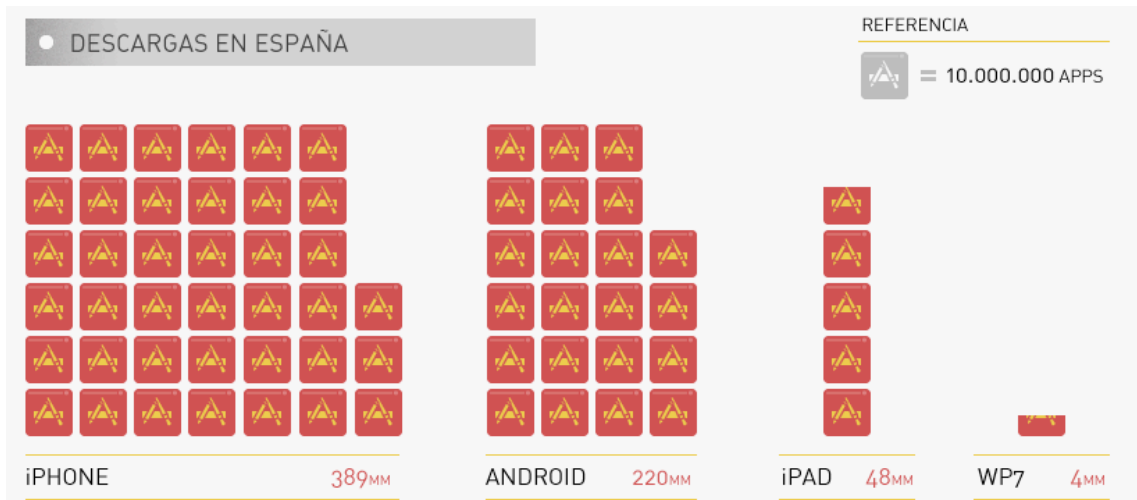


Figura 6.2: Descargas de Apps en España (Extraída de [15])

Por tanto, y como conclusión, es interesante desarrollar una versión móvil nativa de la aplicación web, en iOS (Principalmente iPhone y iPad) y en Android (para smartphones y tabletas).

Apéndice A

Manual de usuario

Este apéndice sirve como manual de usuario para el uso de la aplicación “Apuntweets”. No se requieren conocimientos previos de programación para su utilización, aunque si se necesita algo de práctica en el manejo de redes sociales, como es la plataforma Twitter.

En los subapartados siguientes, explicaremos paso a paso como utilizar la plataforma tanto para personal docente, como estudiantes.

A.1. Requisitos

Es necesario disponer de una cuenta en la red social Twitter. De esta manera los comentarios que se realicen estarán personalizados y tienen un componente social importante al compartirse en esta red con el resto de usuarios.

Para disponer de una cuenta de twitter no hay más que dirigirse a la dirección web <http://twitter.com> y registrarse con un nombre completo, nombre de usuario, contraseña y un email.

A.2. Manejo por personal docente

Para acceder como personal docente, se ha de tener un nombre de usuario y contraseña adicionales. Estos datos son facilitados por el

administrador del sistema ya que no hay habilitado ningún formulario de registro para este uso.

Para acceder a esta opción, hay que hacer *click* en el enlace “¿Eres profesor?” que hay en la pantalla principal de la aplicación después de haber accedido con una cuenta de Twitter. Esto hará que aparezca un formulario web donde se puedan introducir los datos del documento que el personal docente quiera cargar.

A.3. Manejo para todos los usuarios

El resto de usuarios y los que tienen permisos como rol docente pueden acceder a todos los documentos que haya cargados en la aplicación. Para poder leer un archivo sólo hay que hacer *click* en la que se quiera de todos los que hay listados y la aplicación nos llevará al visor de documentos.

Al cargar el documento, podemos ver el documento en sí en la parte izquierda de la pantalla y a la derecha se cargarán los comentarios adjuntos a él.

Para poder leer comentarios que hayan dejado otros usuarios, o comentar nosotros mismos, tenemos que pulsar con el ratón en un párrafo que queramos comentar. A continuación, en la parte derecha se cargará un listado de comentarios si los hubiese y un formulario para escribir nuestro comentario.

Para saber si un párrafo ha sido comentado con anterioridad, en la parte del documento, debajo de cada párrafo habrá un pequeño icono con la imagen de twitter del usuario que haya comentado. De esta manera podemos visualizar que párrafos han sido comentados más fácilmente.

Apéndice B

Glosario de Términos

AJAX (*Asynchronous JavaScript And XML*)

API (*Application Programming Interface*)

COITT (*Colegio Oficial de Ingenieros Técnicos de Telecomunicación*)

CSS (*Cascading Style Sheet*)

DOM (*Document Object Model*)

DTD (*Document Type Definition*)

GIF (*Graphics Interchnage Format*)

HTML (*HyperText Markup Language*).

HTTP (*HyperText Transfer Protocol*).

IDPF (*International Publishing Forum*)

JPEG (*Joint Photographic Experts Group*)

NCX (*Navigation Control file for XML*)

PHP (*Hipertext Pre-Processor*)

PNG (*Portable Network Graphics*)

SVG (*Scalable Vector Graphics*)

W3C (*World Wide Web Consortium*)

WWW (*World Wide Web*)

XHTML (*eXtensible HTML*).

XML (*eXtensible Markup Language*).

Bibliografía

- [1]. HTML5 “A vocabulary and associated APIs for HTML and XHTML” Editor’s Draft 12 February 2011. Accedido en Febrero 2011. Esta especificación está disponible para su descarga en <http://dev.w3.org/html5/spec/Overview.html>

- [2]. Future Web Accessibility: Semantic HTML5 elements. by Aaron Andersen. Accedido en Febrero 2011. Accesible en: <http://webaim.org/blog/future-web-accessibility-html5-semantic-tags/>

- [3]. Compatibility tables for support of HTML5, CSS3, SVG and more in desktop and mobile browsers. Accedido en Mayo 2012. Disponible en: <http://caniuse.com/>

- [4]. SEO con HTML5 y XHTML 2.0. “Joseba Umbelima, SEO y Desarrollo Web, March 1st, 2009”. Accedido en Febrero 2011. Accesible en <http://www.josebaumbelina.com/20090301/seo-con-html5-y-xhtml-20/>

- [5]. Twitter; Artículo de la Wikipedia. Accedido en Septiembre 2011. Accesible en: <http://es.wikipedia.org/wiki/Twitter>

- [6]. Introducción a Javascript. Accedido en Mayo 2012. Disponible en http://www.librosweb.es/javascript/capitulo1/breve_historia.html
- [7]. Deconstruyendo jQuery, MooTools y Prototype. Accedido en Mayo 2012. Disponible en <http://digitta.com/2011/02/vuelvo-deconstruyendo-jquery-mootools-y.html>
- [8]. [Especificación OAuth Core 1.0](http://oauth.net/core/1.0). Accedido en Enero 2012. Disponible en <http://oauth.net/core/1.0>
- [9]. Using OAuth 1.0a – 12/07/2011. Accedido en Septiembre 2011. Accesible en: <https://dev.twitter.com/docs/auth/oauth>
- [10]. “Writing your first twitter application with OAuth” por Snipe en 23/07/2009. Accedido en Septiembre 2011. Accesible en: <http://www.snipe.net/2009/07/writing-your-first-twitter-application-with-oauth/>
- [11]. E-Learning; Artículo de la Wikipedia. Accedido en Septiembre 2011. Accesible en: <http://es.wikipedia.org/wiki/E-learning>
- [12]. 50 ways to use Twitter in the classroom. Accedido en Mayo 2012. Disponible en <http://www.teachhub.com/50-ways-use-twitter-classroom>
- [13]. “Twitter” goes to college. Artículo de US News. Agosto de 2012. Accesible en

- <http://www.usnews.com/education/articles/2010/08/16/twitter-goes-to-college-> Accedido en Septiembre de 2012
- [14]. “Seven free platforms for teaching” Artículo de Freeteach4teachers.com Disponible en <http://www.freeteach4teachers.com/2011/02/seven-free-platforms-for-teaching.html> Accedido en Septiembre de 2012.
- [15]. Informe Apps en España primer trimestre 2012. Febrero 2012. The App Date. Disponible en <http://madrid.theappdate.com/informe-apps/> Accedido en Septiembre 2012.
- [16]. API REST para desarrolladores de Twitter. Accedido en Enero 2012. Disponible en <https://dev.twitter.com/docs/api>
- [17]. Librería “Twitteroauth” PHP de Abraham. Accedido en Septiembre 2011. Open Source. Disponible en <http://github.com/abraham/twitteroauth>
- [18]. Flujo de autenticación OAuth. Twitter API. Accedido en Septiembre 2011. Disponible en <http://dev.twitter.com>
- [19]. Apache Friends. Proyecto de servidor Web Apache. Disponible en <http://www.apachefriends.org/en/index.html>
- [20]. Extensión cURL de PHP. Manual accesible en <http://php.net/manual/es/book.curl.php>. Accedido en Mayo 2012

- [21]. Tarifa de alojamiento Web en ConfigBox. Disponible en <http://www.configbox.com/alojamiento-web-medida.html> Accedido en Julio 2012.
- [22]. Freehostia. Servicio de alojamiento web gratuito. Accesible en <http://www.freehostia.com/free-chocolate.html>
- [23]. Proyecto fin de carrera "Estudio práctico de la tecnología SCWS para tarjetas (U)SIM", Paola García. Noviembre de 2009. Capitulo 6, Historia del proyecto. Accedido en Julio 2012.
- [24]. Ordenador MacBook Air 13". Accedido en Septiembre de 2012. http://store.apple.com/es/browse/home/shop_mac/family/macbook_air/select
- [25]. Tarifas de conexión a internet de ONO. Disponible en http://www.ono.es/resources/files/tarifas/pdfs/residencial_actual.pdf Accedido en Julio 2012.

