



Escola Politècnica Superior  
de Castelldefels

UNIVERSITAT POLITÈCNICA DE CATALUNYA

# TRABAJO FINAL DE CARRERA

**TÍTULO DEL TFC:** Evaluación y pruebas de aplicaciones RIA con AJAX

**TITULACIÓN:** Ingeniería Técnica de Telecomunicación, especialidad Telemática

**AUTOR:** Manel Pérez Mata

**DIRECTOR:** Roc Messeger

**Fecha:** 27 de abril de 2008



**Título:** Evaluación y pruebas de aplicaciones RIA con AJAX

**Autor:** Manel Pérez Mata

**Director:** Roc Messeger

**Fecha:** 27 de abril de 2008

## **Resumen**

Este proyecto tiene como objetivo principal estudiar el funcionamiento de las aplicaciones RIA partiendo de un punto de vista analítico hasta llegar a uno totalmente práctico.

El primer capítulo de este documento, trata de aclarar los conceptos que definen el entorno de trabajo además de describir las ventajas y desventajas de cada uno, de este modo, el lector podrá entender y asimilar los diferentes términos que aparecen a lo largo del documento.

Una vez aclarados estos términos, he elaborado una descripción de la tecnología escogida y a través de una serie de ejemplos sencillos podemos descubrir las ventajas de utilizar como base un framework completo y funcional como jQuery.

Por último, en el capítulo tres, encontramos detallados todos los pasos que he tenido que realizar hasta desarrollar un ejemplo completo de aplicación Web. La presentación de la aplicación y la descripción de los diferentes roles de usuarios nos sirve de punto de partida para entender el trabajo realizado. Posteriormente se detallan las diferentes acciones permitidas y, finalmente, se puede ver el resultado final práctico en una serie de capturas de pantalla que recogen varios puntos interesantes de la aplicación desarrollada.

**Title:** TFC/PFC Model

**Author:** Manel Pérez Mata

**Director:** Roc Messeger

**Date:** April, 27th 2008

## **Overview**

The main objective of this project is to study the performance of applications RIA from an analytical point of view to reach a totally practical one.

The first chapter of this document seeks to clarify the concepts that define the working environment and also describes the advantages and disadvantages of each one. Thus, the reader will be able to understand and assimilate the different terms that appear throughout the document.

Once these terms have been clarified, I have outlined a description of the chosen technology through a series of simple examples in which we can discover the advantages of using as base a complete and functional framework as jQuery.

Finally, in chapter three, we can find all the detailed steps that I had to make till developing a complete example of web application. The presentation of the application and a description of the different roles of users are used as a starting point for understanding the performed work. Afterwards, in this work are detailed the different allowed actions and finally, we can see a practical outcome of a series of screenshots that pick up several interesting points of the developed application.

# ÍNDICE

<b>INTRODUCCIÓN .....</b>	<b>1</b>
<b>CAPÍTULO 1. ENTORNO DEL PROYECTO .....</b>	<b>2</b>
<b>1.1. Aplicaciones Web .....</b>	<b>2</b>
1.1.1. ¿Qué es una aplicación Web? .....	2
1.1.2. Puntos fuertes de una Aplicación Web .....	2
1.1.3. Puntos débiles de una Aplicación Web .....	3
<b>1.2. Concepto RIA .....</b>	<b>5</b>
1.2.1. ¿Qué son las RIA? .....	5
1.2.2. Puntos fuertes de las RIA .....	6
1.2.3. Puntos débiles de las RIA .....	6
<b>1.3. Concepto AJAX .....</b>	<b>7</b>
1.3.1. ¿Qué es AJAX? .....	7
1.3.2. Puntos fuertes de AJAX .....	7
1.3.3. Puntos débiles de AJAX .....	8
<b>1.4. Concepto de CMS .....</b>	<b>9</b>
1.4.1. ¿Qué es un CMS? .....	9
1.4.2. ¿Qué es Joomla!? .....	9
1.4.3. Joomla! en este proyecto .....	10
<b>CAPÍTULO 2. JQUERY EN PRÁCTICA .....</b>	<b>11</b>
<b>2.1. Framework jQuery .....</b>	<b>11</b>
2.1.1. ¿Qué es jQuery? .....	11
2.1.2. ¿Por qué jQuery? .....	11
2.1.3. Sintaxis básica .....	11
2.1.4. Selectores .....	12
2.1.5. Eventos .....	13
2.1.6. Interacción con Ajax .....	13
2.1.7. Efectos visuales .....	14
2.1.8. Otras utilidades .....	14
<b>2.2. Diferentes ejemplos de uso de jQuery .....</b>	<b>16</b>
2.2.1. FishEye Menú .....	16
2.2.2. Diferentes efectos para capas .....	16
2.2.3. Menú acordeón .....	17
2.2.4. Puzzle .....	18
2.2.5. Zoom de imagen .....	19
2.2.6. Cajas arrastrables .....	20
2.2.7. Selección parcial .....	21
2.2.8. Ayuda mientras se escribe .....	22

<b>CAPÍTULO 3. CREANDO UNA APLICACIÓN WEB.....</b>	<b>23</b>
<b>3.1 Presentación de la aplicación .....</b>	<b>23</b>
<b>3.2 Casos de uso de la aplicación .....</b>	<b>23</b>
3.2.1 Usuarios anónimos.....	23
3.2.2 Usuarios autenticados.....	24
3.2.3 Administradores.....	25
<b>3.3 Detalle de acciones posibles.....</b>	<b>25</b>
<b>3.4 Resultado final.....</b>	<b>31</b>
<b>CONCLUSIÓN .....</b>	<b>33</b>
<b>REFERENCIAS BIBLIOGRÁFICAS.....</b>	<b>34</b>
<b>ANEXO 1. MANUAL DE USUARIO DE LA APLICACIÓN.....</b>	<b>35</b>

# INTRODUCCIÓN

Las aplicaciones Web suponen un paso adelante en el mundo de Internet mediante la creación de páginas Web mucho más robustas y funcionales. Frente los tradicionales diseños, proporcionan un entorno más potente y flexible asemejando una Web a una aplicación de escritorio común. Sin embargo, sin el concepto RIA, la idea de desarrollar una aplicación de escritorio para ser ejecutada desde un navegador Web, quedaría solamente en una eso, una idea.

Hasta el nacimiento de las RIA, una página Web se limitaba a una serie de páginas enlazadas mediante hipervínculos por las cuales el usuario iba solicitando contenido a medida que iba saltando de un enlace a otro. Como podremos ver en este proyecto, con el nuevo concepto RIA, la petición de una información concreta, no supone la descarga total de la información, sino que podemos solicitar unidades de información sin que el usuario tenga la sensación de estar cambiando de página.

Actualmente existen varios desarrollos de gran envergadura como el Escritorio Web eyeOS en el cual, se ha conseguido desarrollar un sistema operativo vía Web que asemeja en todos los aspectos a un sistema operativo convencional con aplicaciones tan utilizadas como editores de texto, gestores de archivos, juegos, hojas de cálculo y un largo etcétera de programas, pero con la ventaja de que todo ello está funcionando en un servidor Web, con lo cual puedes acceder a tu escritorio desde cualquier máquina conectada a Internet.

Para poder llevar a cabo este proyecto, realicé un análisis de las diferentes tecnologías existentes del lado de cliente y servidor, que nos permiten desarrollar mediante el concepto RIA una aplicación Web completa.

De los diferentes frameworks disponibles que he encontrado del lado del cliente me decanté por jQuery, una librería JavaScript que nos permite realizar desarrollos complejos de manera sencilla. A lo largo de los diferentes ejemplos desarrollados y que detallo más adelante, podemos hacernos una idea de la potencia de dicha librería.

Para el desarrollo de la aplicación Web, decidí elaborar un gestor de archivos sencillo similar al explorador de Windows. En este caso, necesitaba de una tecnología del lado del servidor; aposté por php, que es un lenguaje de programación ampliamente extendido por su ligereza, velocidad y sencillez además de su grandes capacidades como lenguaje de programación para desarrollos Web.

# CAPÍTULO 1. ENTORNO DEL PROYECTO

## 1.1. Aplicaciones Web

### 1.1.1. ¿Qué es una aplicación Web?

Las aplicaciones Web son aplicaciones convencionales que se ejecutan desde un servidor Web externo.

Utilizan la Web como mecanismo de transmisión de información entre la aplicación y el usuario, convirtiendo al navegador Web en la interfaz primaria de la aplicación, lo cual tiene la ventaja de que dicha aplicación será siempre multi-plataforma ya que la mayoría de sistemas operativos actuales lleva integrados o se les puede instalar un navegador Web.

Además, cuando una aplicación adopta al navegador Web como interfaz de usuario, elimina la necesidad de que sus usuarios tengan que instalar software adicional para poder correr dicha aplicación. Y, a medida que se desarrollen y distribuyan nuevas versiones de la aplicación en el servidor, los usuarios comienzan a utilizarlas de forma automática, sin que sea necesario hacer actualizaciones.

### 1.1.2. Puntos fuertes de una Aplicación Web

- **Movilidad**

La movilidad es una de las grandes ventajas de las aplicaciones Web, ya que nos da la libertad de poder acceder a nuestra aplicación desde cualquier máquina que disponga de un navegador Web y una conexión a Internet.

Además, no necesitaremos transportar los documentos generados debido a que estos también se almacenarán en los servidores, por lo tanto, sé que utilice la aplicación desde el ordenador que la utilice, siempre podré acceder a documentos iniciados anteriormente.

- **Multiplataforma**

El único requerimiento en la máquina del usuario es que disponga de un navegador Web, esto hace que las aplicaciones Web puedan funcionar perfectamente en cualquier plataforma actual.



- **Actualización automática**

Cualquier cambio en el programa será visible a todos los usuarios sin necesidad de descargar e instalar ninguna actualización.

- **Compartir información**

Siempre que la aplicación lo permita, compartir archivos, información, documentos... con otros usuarios es algo muy sencillo debido a que todos están alojados en la misma máquina.

- **Seguridad**

Debido a la centralización del programa que está únicamente en una máquina (o si la aplicación es muy grande, en un número pequeño de máquinas), los esfuerzos para mantener la aplicación segura recalcan únicamente en los servidores, por lo tanto el gasto de dinero y esfuerzo para mantener la máquina libre de ataques y virus es mucho menor.

Así mismo, este punto también tiene su parte negativa, esto lo analizaré en los puntos débiles de las aplicaciones Web.

### 1.1.3. Puntos débiles de una Aplicación Web

- **Centralización**

La centralización, el punto que nos permite estar hablando en este documento de aplicaciones Web, es también uno de los puntos débiles.

Al estar toda la información concentrada en una única máquina, el mal funcionamiento de esta puede ser fatal para todos los usuarios, por ejemplo, un mal funcionamiento de la red, puede impedir que todos los usuarios de una empresa trabajen de manera correcta, o incluso peor, un problema en el disco duro de dicha máquina, puede provocar la pérdida de todo el trabajo realizado hasta el momento.

Para solventar esto, es conveniente disponer de buenos sistemas de backup y accesos alternativos a la aplicación.

- **Seguridad**

Tal y como menciono anteriormente, este punto se puede ver desde las dos vertientes, como punto fuerte o débil.

Al estar todo centralizado en una máquina conectada todo el tiempo a una red, esta está expuesta siempre a posibles ataques y, por ejemplo, la toma de control de un hacker al servidor, puede significar el acceso a toda la información de una la empresa.

Es de vital importancia centrar todos los esfuerzos en que la máquina o máquinas centrales estén protegidas al máximo.

- **Lentitud**

En aplicaciones pesadas, que manejan grandes volúmenes de información, el acceso desde un navegador Web puede ser realmente lento y llegar incluso a imposibilitar su uso.

## 1.2. Concepto RIA

### 1.2.1. ¿Qué son las RIA?

**RIA**, acrónimo de **Rich Internet Applications** (Aplicaciones Ricas en Internet), es un tipo de aplicación Web con más ventajas que los tradicionales desarrollos Web.

Dicho termino aparece en el año 2002 en un documento publicado por Macromedia, ahora Adobe. Posteriormente el concepto RIA se ha ido haciendo un hueco en la red debido a las grandes posibilidades que nos ofrece.

Poniendo en práctica la idea de RIA podemos realizar programas con las funcionalidades de una aplicación de escritorio común, con la gran ventaja de que no es necesario que el usuario tenga instalada la aplicación en su máquina (con el consiguiente ahorro de espacio, compra de licencias, posibilidad de infección de virus...), sino que son accesibles desde cualquier navegador Web, con lo cual obtendremos una aplicación multiplataforma que funcionará en cualquier máquina con un navegador Web y conexión a Internet.

La principal idea de este tipo de aplicaciones es la de separar el método en que se envía y recibe el flujo de información entre la máquina cliente y el servidor.

Las aplicaciones Web tradicionales basan todo el procesamiento de datos en el servidor, recargando toda la página con cualquier acción que haga el usuario, aún si el cambio realizado es mínimo, mientras que en las RIAs sólo es necesario cargar la interfaz una sola vez, y para cada acción del usuario hacemos una petición específica de la información solicitada al servidor sin la necesidad de recargar toda la página. De este modo conseguimos reducir el tiempo de espera del usuario, el ancho de banda consumido, la carga del servidor... y teniendo la posibilidad de integrar un feedback para que el usuario este enterado de que esta pasando en la aplicación en cada momento.

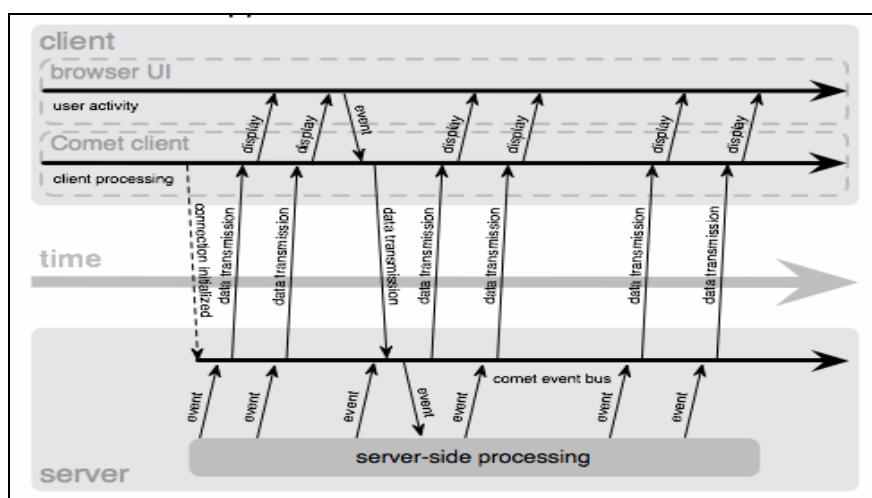


Fig. 1 - Modelo de aplicación RIA

### 1.2.2. Puntos fuertes de las RIA

- **Centralización de la aplicación**

La aplicación es accesible desde cualquier ordenador que disponga de un navegador Web y conexión a Internet.

- **Herramientas más potentes**

Las tecnologías que implementan el concepto RIA suelen facilitar la generación de efectos que hacen la navegación mucho más cómoda.

- **Mayor capacidad de respuesta**

Al no tener que recargar la página cada vez, la navegación por la Web es mucho más ágil.

- **Eficiencia de uso de la red**

El intercambio de información entre cliente y servidor es mucho menor ya que únicamente pedimos la información que necesitamos en cada momento y no información que ya habíamos solicitado anteriormente.

### 1.2.3. Puntos débiles de las RIA

- **Desorientación del usuario**

A un usuario acostumbrado a un modelo tradicional de interacción web, el refresco de página le informa que el sistema ha recibido su petición. Con RIA conviene replantear el modelo de respuesta de la pantalla para que el usuario perciba que el sistema le está atendiendo.

- **El uso del botón “Atrás” del navegador**

Gran parte de los usuarios acuden instintivamente a él cuando se equivocan. En aplicaciones basadas en RIA existe un riesgo de malfuncionamiento, ya que el botón “Atrás” funciona para peticiones de páginas diferentes.

- **Acceso a información mediante indexación**

Con RIA hay que ser muy cautos al diseñar flujos de pantallas ya que se puede caer en el error de ocultar información importante y perder así la correcta indexación en los buscadores Web.

## 1.3. Concepto AJAX

### 1.3.1. ¿Qué es AJAX?

**AJAX**, acrónimo de **A**synchronous **J**avaScript **A**nd **X**ML (JavaScript y XML asíncronos), es un conjunto de tecnologías que usadas convenientemente nos permite crear aplicaciones interactivas o RIA.



Fue nombrado por primera vez en el año 2005 agrupando en este término varias tecnologías ampliamente extendidas.

Las tecnologías que nos permiten hablar de Ajax como un concepto completo son estándares como HTML, XHTML y CSS para la presentación de la información, DOM y JavaScript que permite interactuar dinámicamente con dicha información y XMLHttpRequest que nos facilita el canal mediante el cual comunicar cliente y servidor Web.

Siguiendo la lógica propuesta por RIA, Ajax no es más que una técnica de desarrollo web para crear aplicaciones interactivas que se ejecutan en el navegador del usuario, y mantiene una comunicación asíncrona con el servidor en segundo plano para recargar partes de la página sin necesidad de recargarla toda entera. De esta manera aumentamos en gran medida la interactividad, velocidad y usabilidad en la misma.

De la enorme variedad de tecnologías que siguen el modelo propuesto por RIA, Ajax se está desmarcando con diferencia. El número de páginas Web que utilizan Ajax crece de manera exponencial debido en gran medida, al impulso que está recibiendo por parte de dos gigantes de Internet: Google y Yahoo. Ambas empresas están basando en esta tecnología todas sus herramientas y su impacto se ha visto reflejado de manera inmediata.

### 1.3.2. Puntos fuertes de AJAX

Cómo subconjunto de RIA, AJAX asimila todos los puntos fuertes mencionados anteriormente para las RIA. Además de estos, podemos encontrar:

- **Independencia de la tecnología usada en el servidor**

Es perfectamente compatible con cualquier tipo de servidor estándar y lenguaje de programación Web. PHP, ASP. ASP.Net, Perl, JSP, Cold Fusion...

Esta característica ha ayudado en gran medida a que su uso sea cada vez más habitual.

- **Mejora la estética de la Web**

Mediante la utilización de esta tecnología se pueden conseguir efectos visuales muy logrados que nos permiten simular a la perfección una aplicación de escritorio convencional.

- **Compatible con Flash**

Existe un debate abierto entre el uso de Flash o AJAX. Definitivamente hay ventajas y desventajas en ambas tecnologías según la situación que se de pero también hay muchas posibilidades y muy buenas para que ambas funcionen en conjunto.

### **1.3.3. Puntos débiles de AJAX**

Del mismo modo que en los puntos fuertes, AJAX asimila todos los puntos débiles mencionados anteriormente para las RIA. Además de estos, podemos encontrar:

- **Navegadores sin JavaScript**

Si no diseñamos correctamente nuestra aplicación, se corre el riesgo de que los usuarios que tienen deshabilitado JavaScript no puedan navegar correctamente por nuestra Web.

- **Tiempo de espera en cargas de página**

Una tentación para muchos desarrolladores es incluir gran cantidad de código Javascript. Hay que ser prudentes y evitar caer en esta tentación ya que implica no sólo un consumo de ancho de banda importante y su correspondiente tiempo de carga, sino que también implica tiempo de procesador que puede llegar a un desbordar la máquina del cliente al no ser capaz de tratar a todas las instrucciones.

## 1.4. Concepto de CMS

### 1.4.1. ¿Qué es un CMS?

Un Sistema de gestión de contenidos o Content Management System en inglés (abreviado **CMS**) permite la creación y administración de contenidos principalmente en páginas Web.

Consiste en una interfaz que controla una o varias bases de datos donde se aloja el contenido de un sitio Web.

El sistema permite manejar de manera independiente el contenido y el diseño. Así, es posible manejar el contenido y darle en cualquier momento un diseño distinto al sitio sin tener que darle formato al contenido de nuevo, además de permitir la fácil y controlada publicación a varios editores.

Un ejemplo clásico es el de editores que cargan el contenido al sistema y otro de nivel superior que permite que estos contenidos sean visibles a todo público.

Actualmente existen gran cantidad de CMS gratuitos y de pago en Internet y programados en diferentes tecnologías, así, podemos encontrar algunos programados en java, como por ejemplo OpenCMS, Magnolia, OpenEdit... o en php PHP-Nuke, pHNews o Joomla! por citar algunos ejemplos.

### 1.4.2. ¿Qué es Joomla!?

Tal y como he mencionado anteriormente, Joomla es un CMS que permite a cualquier usuario con unos conocimientos muy básicos de desarrollo Web, crear en unos pocos pasos un portal totalmente funcional para colgar noticias, artículos, opiniones... o incluso, si se le añade un módulo específico, crear una tienda on-line.



Este gestor de contenidos, nace a mediados de 2005, cuando los administradores del equipo de desarrollo de Mambo (CMS antecesor de Joomla!) decidieron limitar las condiciones de la licencia de uso de su software.

Ante el desacuerdo de los principales desarrolladores de la organización, más de 30 miembros principales del equipo renunciaron de inmediato y fundaron Joomla! (palabra de origen africano que significa "todos juntos") de software libre.

### **1.4.3. Joomla! en este proyecto**

Para el desarrollo del gestor de archivos decidí basarme en una pequeña herramienta que dispone este CMS para gestionar las diferentes imágenes de la Web. A pesar de que en este proyecto no se demuestra todo el potencial de Joomla!, este punto es ideal como base para la creación de mi gestor de archivos.

Dado que la finalidad del proyecto es la de poder utilizar la creación del gestor de archivos como escaparate de las diferentes posibilidades que nos ofrece la idea de aplicación Web y RIA, y no la de la creación del gestor en sí, el tutor del proyecto y yo creímos conveniente aprovechar la estructura que facilita Joomla! para añadirle nuevas funcionalidades. Por ese motivo, no me extenderé mucho en la descripción de este CMS.



## CAPÍTULO 2. JQUERY EN PRÁCTICA

### 2.1. Framework jQuery

#### 2.1.1 ¿Qué es jQuery?

jQuery es un framework JavaScript muy ligero que simplifica el desarrollo de la parte de cliente en las aplicaciones web. Mediante una serie de sencillas herramientas nos permite crear fácilmente aplicaciones dinámicas complejas.



A principios de 2006, John Resig presentaba en Internet la primera versión de jQuery; desde ese día gran cantidad de programadores han confiado en él llegando a igualar o incluso superar en usuario frameworks similares, ampliamente difundidos, como Prototype.

La idea básica que persigue esta librería, es la de poder generar un código reutilizable, que permita mejorar los resultados de la programación JavaScript convencional y reducir el número de líneas programadas.

#### 2.1.2 ¿Por qué jQuery?

La primera tarea a realizar al empezar este proyecto, era hacer una búsqueda de los diferentes frameworks del lado del cliente que existen en la actualidad y elegir uno de ellos.

De entre la veintena de resultados que encontré, jQuery me pareció el más equilibrado entre potencia y facilidad de uso. Como su propio autor indica, este framework pretende que los desarrolladores Web puedan “*hacer más escribiendo menos*” y, a lo largo de los diferentes ejemplos que he ido desarrollando en este proyecto, he podido comprobar que esto es cierto, pues con unas cuantas líneas se pueden crear efectos visuales tan complejos como selecciones parciales de imágenes con marcos de tamaño ajustable.

#### 2.1.3 Sintaxis básica

Una de las características que facilitan el aprendizaje de su sintaxis está en el uso de la función `$()`, que se podría comparar con `document.getElementById()` de JavaScript, pero con la diferencia de que soporta selectores CSS y es capaz de devolver arrays.

En la siguiente tabla podemos ver la nomenclatura utilizada:

Tipo de elemento Modo de llamado	Método llamado
Etiqueta Nombre de la etiqueta.	div
Elemento con un Id Id precedido por el signo "#".	#principal
Elemento con una clase (class) Clase precedida por un punto (.)	.codigo

Cada comando comienza con el símbolo "\$". Ahora, la forma completa de una sentencia es la siguiente:

```
$(elemento).evento(funcion-o-parametro);
```

## 2.1.4 Selectores

Mediante los selectores, podemos interactuar con los elementos de la página html para crear efectos, borrar contenido, etc.

```
// Seleccionar un elemento
$("#miParrafo");

// Seleccionar todos los enlaces
$("a");

// Seleccionar todo lo anterior
$("#miParrafo, a")

// Seleccionar mediante la clase CSS
$(".miClase")

// Seleccionar todos los párrafos de la página que tengan al menos un
enlace
$("p[a]")

// Seleccionar todos los "div" que no estén ocultos
$("div:visible")

// Seleccionar todos los radio buttons que han sido seleccionados
$("input[@type=radio][@checked]")

// Seleccionar varios elementos mediante sus "id" y "class" (devuelve
un array)
$("p.importante, div#menu, span#introduccion p.especial a")

// Seleccionar todos los párrafos de la página
$(" /html/body//p");

// Seleccionar todos los enlaces cuyo atributo "rel" valga "nofollow"
```

```
$( "//a[@ref='nofollow']" );

// Selecciona todos los elementos pares de una lista
$( "ul#menu li:even" )

// Selecciona los 4 primeros párrafos de la página
$( "p:lt(4) " )

// Selecciona todos los enlaces con el texto "pincha aquí"
$( "a:contains('pincha aquí')" )
```

## 2.1.5 Eventos

También podemos hacer llamadas a eventos javascript si hacemos uso de la sintaxis adecuada.

```
// Al pinchar sobre un párrafo de la página, se muestra un alert() con
su contenido
$( "p" ).bind( "click", function() {
    alert( $( this ).text() );
} );

// Ejecuta el evento "onclick" en todos los párrafos de la página
$( "p" ).click();

// Asocia una función al evento "onclick" de todos los párrafos de la
página
$( "p" ).click( function() {
    alert( $( this ).text() );
} );
$( "p" ).toggle( function() {
    alert( "Me acabas de activar" );
}, function() {
    alert( "Me acabas de desactivar" );
} );
```

## 2.1.6 Interacción con Ajax

jQuery implementa un potente módulo AJAX, mediante el cual podemos realizar conexiones asincronas con el servidor para recoger información, por ejemplo, de una base de datos o un fichero externo.

```
// Cargar el contenido de una página HTML en un elemento
$( "div#noticias" ).load( "noticias.html" );

// Petición GET o POST al servidor con parámetros y función de
respuesta
$.ajax({
    type: "POST",
    url: "formulario.php",
```

```
    data: "nombre=Ana&apellido=Osuna",
    success: function(datos){
    alert( "Se guardaron los datos: " + datos);
    }});
```

### 2.1.7 Efectos visuales

Uno de los grandes atractivos de jQuery es su simplicidad a la hora de realizar efectos visuales. En el siguiente recorte de código, podemos ver algunos de los efectos más utilizados:

```
// Ocultar un elemento
$("#p").hide();

// Mostrar un elemento
$("#p").show();

// Si estaba oculto, mostrarlo; si era visible, ocultarlo
$("#p").toggle();

// Mostrar el elemento con una animación muy suave
$("#p").show("slow");
// Desplegar/ocultar el elemento con una animación muy rápida
$("#p").slideToggle("fast");

// El elemento desaparece con un fundido muy suave
$("#p").fadeOut();
```

### 2.1.8 Otras utilidades

```
// Obtiene el valor de la propiedad "color" de CSS del primer párrafo
de la página
$("#p").css("color");

// Establece el valor de la propiedad "color" de CSS
$("#p").css("color","red");

// Establece varias propiedades CSS de un elemento
$("#p").css({ color: "red", background: "blue", font-weight: "bold" });

// Oculta todos los elementos de un formulario
$(miFormulario.elements).hide();

// Devuelve "true" si el navegador es alguna versión de Internet
Explorer
$.browser.msie

// Devuelve "true" si el navegador es alguna versión de Safari
$.browser.safari
```

```
// Obtiene o establece el valor del "innerHTML" del elemento
$("#contenidos").html();
$("#contenidos").html("<h1>Contenidos</h1><p>Cargando...</p>");

// Añade la clase CSS a todos los elementos o la quita si ya la tenían
establecida
$("p").toggleClass("seleccionado");
```

## 2.2 Diferentes ejemplos de uso de jQuery

### 2.2.1 FishEye Menú

El primer ejemplo desarrollado se trata del clásico menú tan popular en los sistemas operativos Linux y Mac OS, que agranda los iconos (en éste ejemplo en concreto los iconos són números) al pasar por encima el puntero del mouse.



Fig. 2 FishEye Menu

Crear un efecto como este con jQuery es tan sencillo como hacer una llamada a la función Fisheye pasándole una serie de parámetros, a continuación, un ejemplo del código utilizado:

```
$(document).ready(  
  function() {  
    $('#menu2').Fisheye({  
      maxWidth: 60,  
      ...  
    })  
  }  
);
```

### 2.2.2 Diferentes efectos para capas

En este ejemplo he recogido una serie de efectos que permiten interactuar con los elementos de la página, por ejemplo, abrir una div de izquierda a derecha, realizar el parpadeo de una imagen, o hacer que se atenue hasta desaparecer, por citar varios ejemplos.

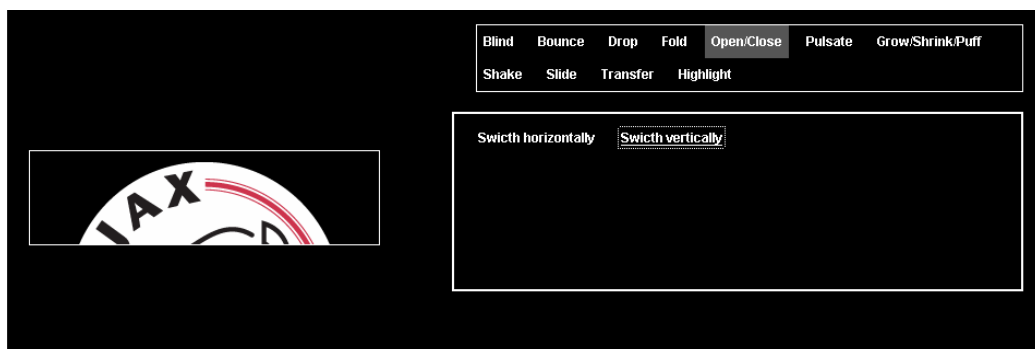


Fig. 3 Imágen cerrándose con un efecto Switch vertical



```
$(items).not(active).children(contents).hide();
$(items).not(active).each(onHide);
$(active).each(onShow);
```

## 2.2.4 Puzzle

Convinando una imagen y una serie de capas con jQuery, se puede crear un puzzle como el que podemos ver en la figura 5. Cada una de las piezas permite seleccionar y arrastrar para ir componiendo el resultado final.



Fig. 5 Puzzle con jQuery

A partir de una imagen troceada y añadida al código con la clase “*sortableitem*” montamos la estructura básica del puzzle. El efecto de coger y arrastrar las diferentes piezas es mérito de la función “*Sortable*” configurada convenientemente según una serie de parámetros.

```
<div>
  
  
  
  ...
</div>

...

$(document).ready(
  function () {
    $('div').Sortable( {
      accept :           'sortableitem',
      helperclass :     'sorthelper',
      activeclass :     'sortableactive',
      hoverclass :     'sortablehover',
      opacity:         0.8,
      revert:           true,
      floats:           true,
      tolerance:       'pointer'
    })
  }
);
```



## 2.2.5 Zoom de imagen

Haciendo uso de la barra superior, podemos hacer grande o pequeño cualquier contenido de nuestra web. En el ejemplo, hemos utilizado una imagen.



Fig. 6 Zoom de imagen

Lo primero que se debe hacer, es incluir la imagen dentro de una capa identificada con un id que nos permitirá interactuar con ella a través de la función “Slider” de jQuery.

Como podemos ver en el siguiente fragmento de código, hago uso de un parámetro onSlide que implementa en su interior una función de callback encargada de definir el comportamiento de la imagen al interactuar con el slide superior.

```
<div id="imageHolder">
    
</div>

...

$('.slider4').Slider({
    accept : '.indicator',
    opacity: 0.8,
    onSlide : function( cordx, cordy) {
        cordx/100);
        $('#imageHolder img').attr('width', 50 + 250 *
    },
    values: [
        [290,0]
    ]
} );
```

## 2.2.6 Cajas arrastrables

Este ejemplo está bastante de moda a raíz de su uso en webs tan importantes como google o yahoo.

Se trata de una serie de cajas que se pueden seleccionar y arrastrar para colocarlas en el orden que se desee. Mediante los controles superiores, se pueden expandir o contraer el contenido de todas ellas o incluso ocultarlas.



Fig. 7 Ejemplo de cajas arrastrables

De manera similar al menú acordeón, en este ejemplo he tenido que crear mediante capas y una hoja de estilos CSS el formato de las diferentes cajas. Para permitir el movimiento recurrimos, como en el ejemplo del puzzle, a hacer uso de la función “Sortable”. Además, debemos hacer uso de unas funciones específicas que nos permitirán minimizar, maximizar u ocultar las diferentes capas.

```
// Controls Drag + Drop
$('#columns td').Sortable( {
  ...
} );

...

// Expand All Portlets
$('#a#all_expand').click(function() {
  $('#div.portlet_content:hidden').show();
  return false;
});

// Collapse All Portlets
$('#a#all_collapse').click(function() {
  $('#div.portlet_content:visible').hide();
  return false;
} );

...
```

## 2.2.7 Selección parcial

Un efecto, especialmente vistoso, es el que podemos ver en la figura 8. Con unas pocas líneas, podemos crear un marco de tamaño variable y que se puede arrastrar para seleccionar la parte de la imagen que más nos guste.

Modificando unos parámetros, podemos oscurecer más o menos el fondo o permitir que el marco se pueda agrandar hasta un máximo tanto de ancho como de alto.

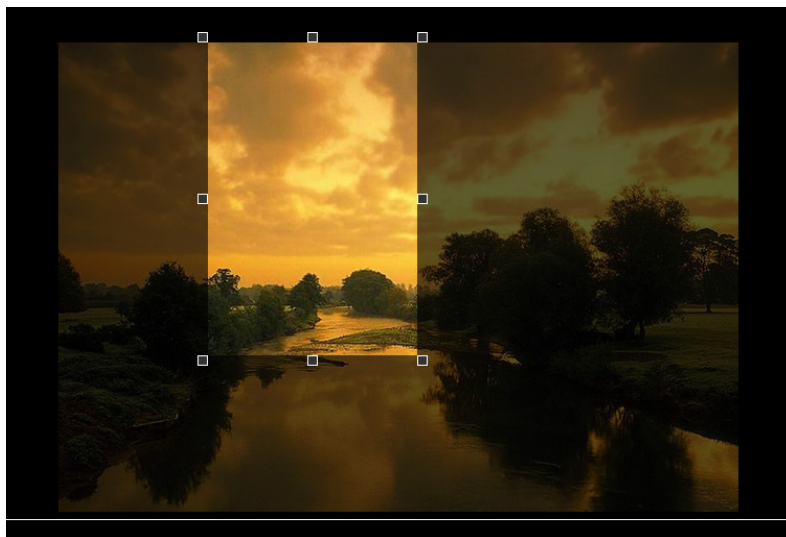


Fig. 8 Selección parcial

Con una serie de capas he creado una pequeña ventana expansible y mediante la función “*Resizable*” se define la zona de arrastre, el ancho máximo y mínimo que se puede conseguir, la posición inicial, si queremos que se pueda arrastrar...

Por otro lado, para completar el ejemplo necesitamos dos imágenes, una con el color natural y otra oscurecida. Con la ventana que comentaba anteriormente, se define un área en la que se muestra la imagen de color natural, con esta pequeña “trampa” se crea este efecto tan vistoso.

```
<div id="resizeMe">
  <div id="resizeSE"></div>
  <div id="resizeE"></div>
  ...
</div>

...

$(document).ready(
  function() {
    $('#resizeMe').Resizable( {
      minWidth: 50,
      maxWidth: 400,
      ...
    }
  )
}
```

## 2.2.8 Ayuda mientras se escribe

jQuery incorpora un completo módulo Ajax que facilita enormemente la tarea de realizar conexiones asincronas con el servidor. En este ejemplo he querido recoger la facilidad de programar conexiones ajax mediante este framework javascript.

En el campo de texto se nos solicita que insertemos el nombre de un país. Mientras escribimos, nos aparece una pequeña capa con sugerencias sobre los nombres que podríamos insertar.

Aparentemente, todas las sugerencias podrían estar contenidas en la página Web de manera estática y simplemente mostrarse u ocultarse mediante javascript, sin embargo, lo que sucede realmente es algo más complejo:

- 1) Cuando el usuario teclea cualquier caracter en el campo de texto, se hace una llamada a una función JavaScript que realiza una petición al servidor Web.
- 2) El servidor Web, hace las comprobaciones pertinentes, en este caso solicitar los paises a sugerir y los devuelve a la página que estamos visualizando.
- 3) Una vez la función JavaScript recibe la información, la hace visible al usuario.

Todo este proceso es totalmente transparente al usuario y sin necesidad de recargar la página. A pesar de la sencillez de este ejemplo, podemos hacernos una idea del potencial de utilizar conexiones asincronas con el servidor, ya que en lugar de hacer una simple consulta a base de datos, podríamos hacer cosas más complejas como solicitar información a un feed RSS y mostrarlo al usuario en tiempo real, por citar un ejemplo.

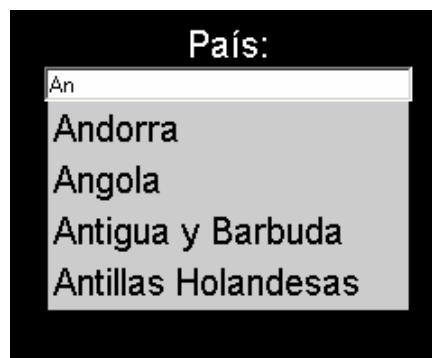


Fig. 9 Caja de sugerencias solicitada mediante Ajax

Podemos ver un ejemplo de uso de las funciones Ajax de jQuery en el punto 2.1.6 *Interaccion con Ajax* de este documento.

## CAPÍTULO 3. CREANDO UNA APLICACIÓN WEB

### 3.1 Presentación de la aplicación

La idea de este proyecto era la de crear una aplicación Web lo más similar posible al Explorador de Windows pero con la característica de que las carpetas y archivos contenidos pudiesen ser compartidos por los usuarios, si estos así lo solicitan.

Para realizar esto, he diseñado una aplicación con varios niveles de usuarios que tienen unos privilegios u otros. De este modo, el programa consta de:

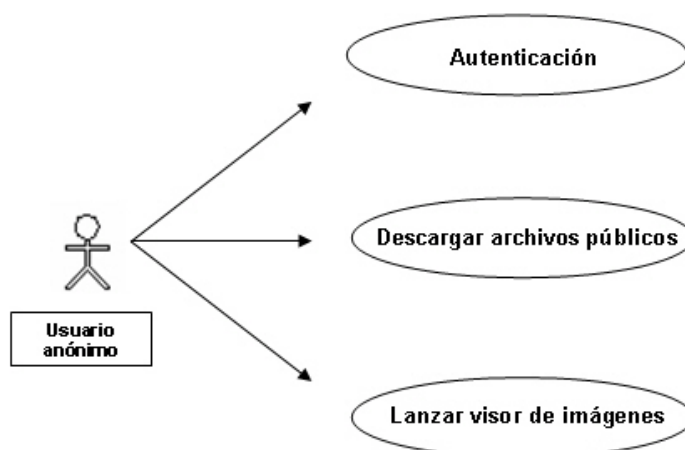
- Uno o varios super usuarios que pueden realizar controles globales y gestionar usuarios y contenido.
- Usuarios registrados, que pueden subir o descargar archivos.
- Usuarios públicos, que únicamente pueden ver los archivos públicos subidos por los administradores o usuarios registrados.

Ayudándome del código de Joomla (que tiene creada la jerarquía de usuarios) y del framework jQuery, he desarrollado una aplicación Web envedida dentro de un navegador, que contiene efectos como el de selección y arrastre de archivos, visualización de información del archivo al pasar el ratón por encima...

### 3.2 Casos de uso de la aplicación

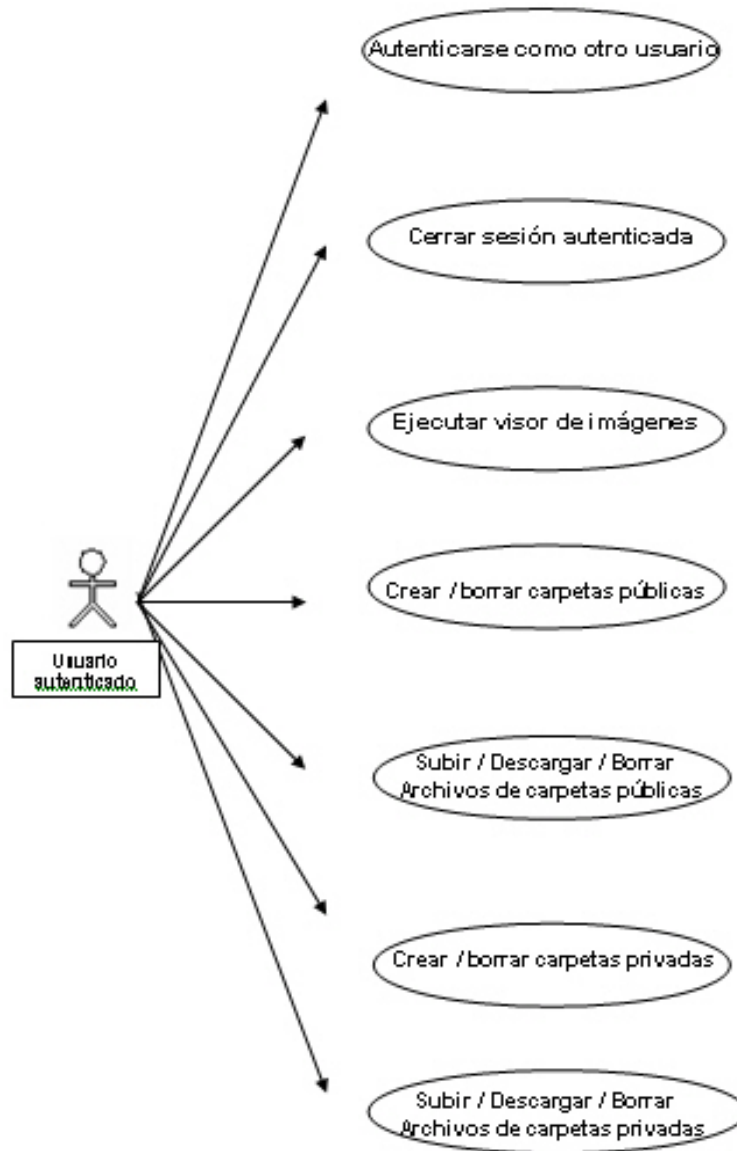
#### 3.2.1 Usuarios anónimos

Los usuarios anónimos únicamente pueden realizar acciones de lectura sobre los elementos públicos.



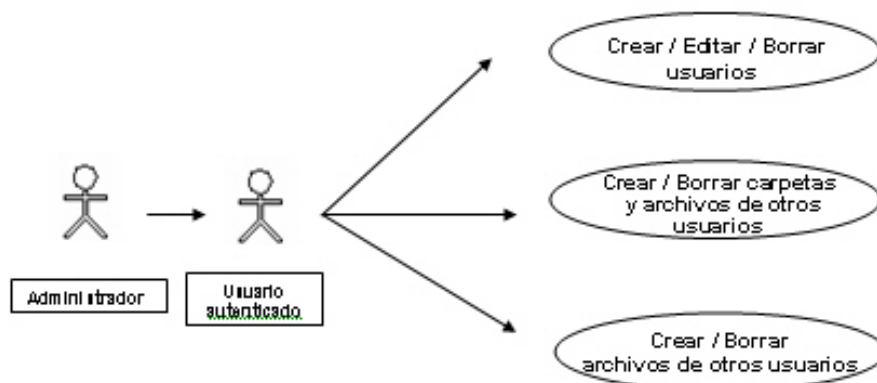
### 3.2.2 Usuarios autenticados

Un usuario dado de alta previamente por uno de los administrativos podrá realizar tareas de lectura y de escritura únicamente sobre sus carpetas y ficheros siempre que esté autenticado.



### 3.2.3 Administradores

Además de los privilegios propios de los usuarios anónimos y autenticados, los administradores tienen una serie de privilegios especiales tales como la creación de usuarios registrados, o la administración de las carpetas de otros usuarios incluidos las de otros administradores.



### 3.3 Detalle de acciones posibles

En este punto detallaré con más profundidad las acciones permitidas en el sistema, cual es el flujo normal de funcionamiento y que usuarios pueden realizar cada una de las acciones.

<b>Nombre</b>	Autenticación
<b>Descripción</b>	Permite a un usuario autenticarse en la aplicación
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios anónimos</li> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	Ninguna
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1- El actor introduce el nombre de usuario y la contraseña</li> <li>2- El sistema comprueba la validez de los datos</li> <li>3- El sistema reenvía al usuario a su zona privada</li> </ol>
<b>Excepciones</b>	Si en el punto 2 la información no es correcta, envía al usuario a una pantalla de error.
<b>Postcondiciones</b>	Ninguna

<b>Nombre</b>	Cerrar sesión
<b>Descripción</b>	Permite a un usuario cerrar una sesión previamente autenticada
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	El usuario debe estar autenticado en el sistema
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1- El actor clicla en la opción <i>Sesión</i> del menú superior</li> <li>2- El actor clicla en la subopción <i>Cerrar sesión</i></li> <li>3- El sistema cierra la sesión para ese usuario y lo reenvía a las carpetas públicas</li> </ol>
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Ninguna

<b>Nombre</b>	Descarga de archivos públicos
<b>Descripción</b>	Permite a un usuario descargar un archivo colocado en una de las carpetas públicas
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios anónimos</li> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	Debe existir algún archivo en las carpetas públicas
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1- El actor accede a la carpeta pública</li> <li>2- El actor clicla un archivo que desea descargar</li> <li>3- Si el archivo se puede visualizar en el navegador, el sistema lo muestra, en caso contrario, lanza una ventana de descarga</li> </ol>
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Ninguna



<b>Nombre</b>	Lanzar visor de imágenes
<b>Descripción</b>	Permite a un usuario ejecutar el visor de imágenes
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios anónimos</li> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	Debe existir alguna imagen en la carpeta en que se desea lanzar el visor de imágenes.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1- El actor accede a una carpeta</li> <li>2- El actor clicca en el enlace 'Visualizar imágenes en visor'</li> <li>3- El sistema lanza una pantalla secundaria en la que se pueden visualizar a modo de diapositivas las imágenes contenidas en la carpeta</li> </ol>
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Ninguna

<b>Nombre</b>	Crear carpetas (públicas o privadas)
<b>Descripción</b>	Permite a un usuario crear una carpeta dentro de la ruta de navegación actual
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Flujo Normal</b>	<ol style="list-style-type: none"> <li>1- El actor rellena el campo de texto <i>Crear directorio</i> con el nombre deseado para la nueva carpeta</li> <li>2- Posteriormente clicca en el icono con la descripción <i>Crear</i></li> <li>3- El sistema verifica que el nombre es correcto, crea la nueva carpeta y recarga la página en la que se muestran los cambios.</li> </ol>
<b>Excepciones</b>	El nombre de carpeta contiene algún carácter erróneo
<b>Postcondiciones</b>	Aparece una nueva carpeta en la ruta actual de navegación

<b>Nombre</b>	Eliminar carpetas (públicas o privadas)
<b>Descripción</b>	Permite a un usuario eliminar una carpeta dentro de la ruta de navegación actual
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>· El usuario debe estar autenticado.</li> <li>· Debe existir, al menos, una carpeta para poder ser eliminada</li> </ul>
<b>Flujo Normal</b>	<p>1- El actor accede a la carpeta en la que se encuentra la carpeta a eliminar.</p> <p>2- Para eliminar la carpeta se pueden realizar dos acciones:</p> <p style="padding-left: 40px;">2.1- Hacer clic en el icono de la papelera debajo de la carpeta</p> <p style="padding-left: 40px;">2.2- Hacer clic sobre el icono de la carpeta y arrastrarlo a la papelera</p> <p>3- El sistema verifica que la carpeta se puede eliminar, la elimina y recarga la página en la que se muestran los cambios</p>
<b>Excepciones</b>	Si la carpeta a eliminar no está vacía, muestra una pantalla informando de que se eliminarán todos los archivos y carpetas contenidos en su interior
<b>Postcondiciones</b>	Desaparece la carpeta que deseábamos eliminar y se pierde toda su información

<b>Nombre</b>	Subir archivos
<b>Descripción</b>	<p>Permite a un usuario subir un nuevo archivo a una carpeta de dominio público o a su carpeta privada.</p> <p>Si es un administrador, se le permite subir archivos a cualquier carpeta.</p>
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	El usuario debe estar autenticado.
<b>Flujo Normal</b>	<p>1- El actor rellena el campo de texto <i>Subir archivo</i> insertando la ruta en la que se encuentra dicho archivo en su máquina o buscándola haciendo clic en el botón <i>Examinar</i>.</p> <p>2- Posteriormente clica en el icono con la descripción</p>

	<p><i>Subir</i></p> <p>3- El sistema verifica que el archivo es correcto, lo copia en la carpeta actual y recarga la página en la que se muestran los cambios.</p>
<b>Excepciones</b>	<ul style="list-style-type: none"> <li>· Si el tamaño del archivo es superior a 16Mb el sistema devuelve un error.</li> <li>· Si el archivo corresponde a un tipo potencialmente peligroso (archivo javascript, php...) el sistema devuelve un error.</li> </ul>
<b>Postcondiciones</b>	Aparece un nuevo archivo en la carpeta de navegación actual.

<b>Nombre</b>	Eliminar archivos
<b>Descripción</b>	<p>Permite a un usuario eliminar un archivo de una carpeta de dominio público o de su carpeta privada.</p> <p>Si es un administrador, se le permite eliminar archivos de cualquier carpeta.</p>
<b>Actores</b>	<ul style="list-style-type: none"> <li>· Usuarios autenticados</li> <li>· Administradores</li> </ul>
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>· El usuario debe estar autenticado.</li> <li>· Debe existir, al menos, un archivo para poder ser eliminado</li> </ul>
<b>Flujo Normal</b>	<p>1- El actor accede a la carpeta en la que se encuentra el archivo a eliminar.</p> <p>2- Para eliminarlo se pueden realizar dos acciones:</p> <p style="padding-left: 40px;">2.1- Hacer clic en el icono de la papelera debajo del archivo a eliminar</p> <p style="padding-left: 40px;">2.2- Hacer clic sobre el archivo y arrastrarlo a la papelera</p> <p>3- El sistema verifica que el archivo se puede eliminar, lo elimina y recarga la página en la que se muestran los cambios</p>
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	Desaparece el archivo que deseábamos eliminar

<b>Nombre</b>	Crear usuario
<b>Descripción</b>	Permite a un administrador crear un nuevo usuario de tipo normal o administrador
<b>Actores</b>	· Administradores
<b>Precondiciones</b>	El usuario debe estar autenticado como administrador
<b>Flujo Normal</b>	<p>1- El actor hace clic en la opción <i>Usuarios</i> del menú superior</p> <p>2- Posteriormente hace clic en la subopción <i>Crear usuario</i></p> <p>3- Rellena el formulario y hace clic en el icono con la descripción <i>Guardar</i></p> <p>4- El sistema verifica los datos del formulario y si todo es correcto, crea el nuevo usuario</p>
<b>Excepciones</b>	Si el nombre de usuario ya está escogido, el sistema devuelve un error
<b>Postcondiciones</b>	Se ha creado un nuevo usuario que puede acceder al sistema y modificar o borrar carpetas públicas, en caso de ser usuario normal, y cualquier carpeta, en caso de ser administrador

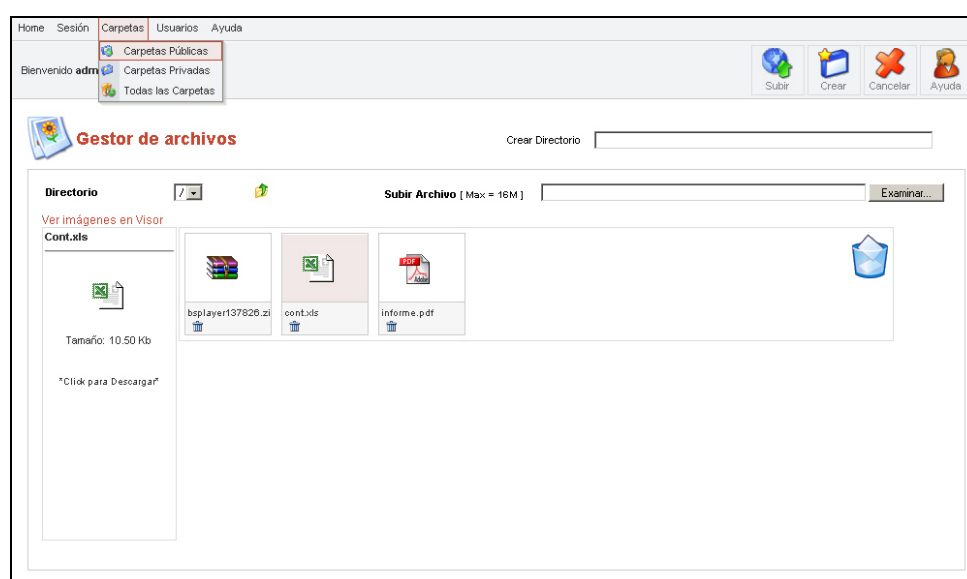
<b>Nombre</b>	Editar usuario
<b>Descripción</b>	Permite a un administrador editar un usuario ya existente en el sistema
<b>Actores</b>	· Administradores
<b>Precondiciones</b>	<ul style="list-style-type: none"> <li>· El usuario debe estar autenticado como administrador</li> <li>· Debe existir al menos un usuario a modificar</li> </ul>
<b>Flujo Normal</b>	<p>1- El actor hace clic en la opción <i>Usuarios</i> del menú superior</p> <p>2- Posteriormente hace clic en la subopción <i>Administrar usuarios</i></p> <p>3- Selecciona el usuario a editar de la lista y hace clic en el icono con la descripción <i>Editar</i></p> <p>4- Aparece un formulario con la información de usuario actual. Modifica los campos a editar y hace clic en el icono con la descripción <i>Aplicar</i></p> <p>4- El sistema verifica los datos del formulario y si todo es correcto, modifica el usuario</p>
<b>Excepciones</b>	Si se modifica el nombre de usuario y este ya está escogido, el sistema devuelve un error
<b>Postcondiciones</b>	Si se modifica el nombre de usuario y/o contraseña, el usuario afectado será expulsado del sistema en caso de estar navegando en ese instante y deberá volver a autenticarse

<b>Nombre</b>	Eliminar usuarios
<b>Descripción</b>	Permite a un administrador eliminar un usuario ya existente en el sistema
<b>Actores</b>	· Administradores
<b>Precondiciones</b>	· El usuario debe estar autenticado como administrador · Debe existir al menos un usuario a eliminar
<b>Flujo Normal</b>	1- El actor hace clic en la opción <i>Usuarios</i> del menú superior 2- Posteriormente hace clic en la subopción <i>Administrar usuarios</i> 3- Selecciona el usuario a editar de la lista y hace clic en el icono con la descripción <i>Eliminar</i> 4- El sistema elimina el usuario seleccionado
<b>Excepciones</b>	Ninguna
<b>Postcondiciones</b>	El usuario afectado será expulsado del sistema en caso de estar navegando en ese instante

### 3.4 Resultado final

Una vez detallado todo el proceso de desarrollo, acciones y roles de usuarios, he creído conveniente mostrar en un par de capturas de pantalla el resultado final de la aplicación.

Si bien, el lector dispone de más información y capturas de pantalla, en un detallado manual de uso que se adjunta en el Anexo 1.



**Fig. 10** Ejemplo de pantalla de administrador

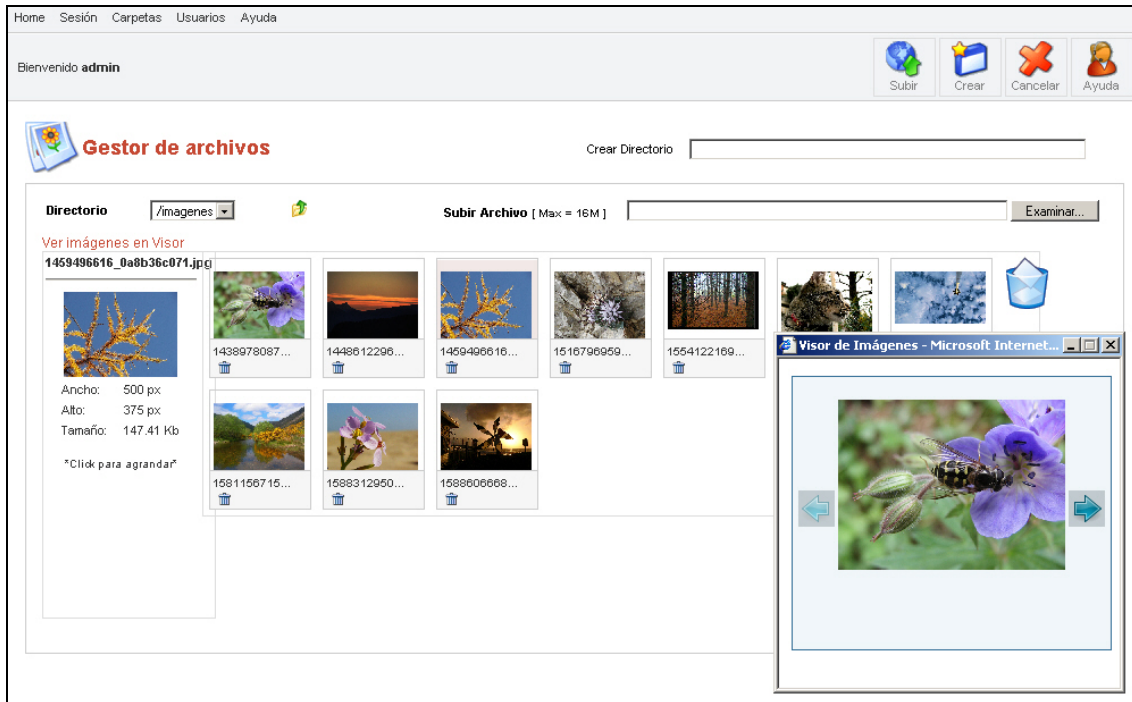


Fig. 11 Ejemplo de carpeta de imágenes con detalle del visor

The screenshot shows a "User Manager" administration interface. At the top, there are navigation links: Home, Sesión, Carpetas, Usuarios, Ayuda. Below this, a welcome message says "Bienvenido admin". On the right, there are icons for "Eliminar", "Editar", "Nuevo", and "Ayuda". The main area is titled "User Manager" and includes a "Filtro:" input field and two dropdown menus: "- Selecciona Grupo -" and "- Selecciona Log Status -". Below this is a table with the following columns: #, Nombre, Nombre de Usuario, En Línea, Activo, Grupo, E-Mail, Última Visita, and ID. The table contains 7 rows of user data. At the bottom, there are navigation links: "<< Primero < Anterior 1 Siguiente > Último >>" and "Mostrando # 30 Resultados 1 - 7 de 7".

#	Nombre	Nombre de Usuario	En Línea	Activo	Grupo	E-Mail	Última Visita	ID
1	Administrador	admin	✓	✓	Super Administrator	info@desenvol.com	2008-04-19 22:31:01	62
2	tete	tete		✓	Administrator	tete@tete.tete	2008-04-19 21:02:59	64
3	juan	juan		✓	Administrator	juan@juan.com	2007-10-16 17:00:19	66
4	romo	romo		✓	Administrator	romo@mailinator.com	1999-11-30 00:00:00	67
5	momo	momo		✓	Administrator	momo@gmail.com	1999-11-30 00:00:00	68
6	memo	memo		✓	Administrator	memo@google.es	1999-11-30 00:00:00	69
7	nono	nono		✓	Administrator	nono@gmail.com	1999-11-30 00:00:00	70

Fig. 12 Pantalla de administración de usuarios

## CONCLUSIÓN

A lo largo de esta memoria he aprendido que es una aplicación Web, partiendo desde un punto de vista teórico hasta un punto de vista más práctico. He podido comprobar que el futuro de este tipo de aplicaciones parece bastante bueno y que ya se está trabajando en diferentes proyectos de envergadura con perspectivas de llegar a buen puerto además de los que ya están en funcionamiento, como podría ser, por citar un ejemplo google maps.

En segundo lugar he descubierto las ventajas de utilizar un framework tan potente como lo es jQuery y de su facilidad de manejo.

Por último he podido comprobar el proceso de desarrollar una aplicación Web completa desde el análisis y la captura de casos de usos hasta el diseño e implementación de la aplicación. Todo ello haciendo uso de diferentes tecnologías que interactúan de manera perfecta entre sí para dar como resultado un programa funcional.

En vista de todo esto creo poder afirmar que se han alcanzado satisfactoriamente los objetivos fijados al inicio del proyecto.

## REFERENCIAS BIBLIOGRÁFICAS

[1] <http://es.wikipedia.org/wiki/Portada>

[2] <http://jquery.com/>

[3] <http://www.joomla.org>

[4] <http://www.anieto2k.com>

[5] <http://www.programatium.com/manuales/ajax>

[6] <http://www.jestudis.com/blogProgramadorWeb.php>

[7] <http://www.codeproject.com>



## ANEXO 1. MANUAL DE USUARIO DE LA APLICACIÓN

El manual de usuario está hecho a base de capturas de pantalla con explicaciones. Creo que este es el modo más sencillo e intuitivo de entender el funcionamiento de la aplicación.

### Menús superiores

Cada usuario puede ver un menú superior diferente en función de su rango de usuario dentro de la aplicación. A continuación detallaré los diferentes menús mostrados según tipo de usuario.

#### - Usuario anónimo:

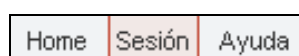


Fig. 13 Menú de usuario anónimo

- Home: Enlace a la página por defecto de la aplicación.
- Sesión: Permite a un usuario anónimo abrir una sesión como usuario registrado o administrador.



Fig. 14 Detalle de submenú Sesión

- Ayuda: Muestra una pequeña ventana de ayuda según la página en la que estemos.

#### - Usuario registrado:

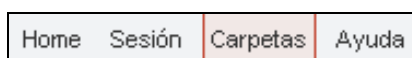


Fig. 15 Menú de usuario registrado

- Home: Enlace a la página por defecto de la aplicación.
- Sesión: Permite a un usuario anónimo abrir una sesión como usuario registrado o administrador.
- Carpetas: El usuario registrado puede navegar por sus carpetas privadas o por las carpetas públicas si lo desea.



Fig. 16 Detalle de submenú Carpetas

· Ayuda: Muestra una pequeña ventana de ayuda según la página en la que estemos.

- Administrador:



Fig. 17 Menú de administrador

· Home: Enlace a la página por defecto de la aplicación.

· Sesión: Permite a un usuario anónimo abrir una sesión como usuario registrado o administrador.

· Carpetas: Un usuario administrador tiene privilegios para navegar por sus carpetas privadas, por las carpetas públicas o por las carpetas de los diferentes usuarios.



Fig. 18 Detalle de submenú carpetas para administrador

· Usuarios: A los usuarios administradores se les permite administrar a los usuarios del sistema y crear nuevos.

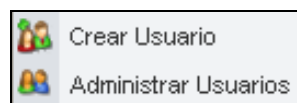


Fig. 19 Detalle de submenú Usuarios

## Login

Cuando un usuario hace click en el submenú Abrir Sesión del menú superior, se le redirige a una página de autenticación donde se le pide que introduzca su nombre de usuario y contraseña.

Para poder acceder, el usuario debe haber sido dado de alta previamente por un administrador.



Fig. 20 Detalle del formulario de autenticación

### Crear una nueva carpeta



Fig. 21 Intranet de usuario registrado o administrador con detalle de creación de carpeta

Para crear una carpeta, un usuarios registrado o administradores debe rellenar el campo de texto “Crear directorio” (1) y hacer click en el icono “Crear” (2).

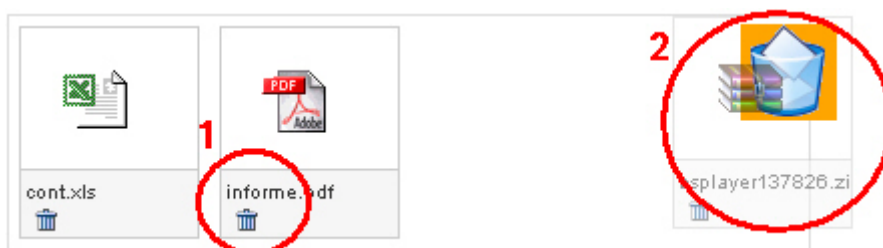
### Subir un archivo



Fig. 22 Intranet de usuario registrado o administrador con detalle de alta de archivo

Para subir un archivo, el usuarios registrado o administradores debe navegar hasta la carpeta en la que quiere guardar el archivo, buscar el archivo en su disco duro en el campo “Subir archivo” (1) y hacer click en el icono “Subir” (2).

### Eliminar archivos o carpetas



Se pueden eliminar archivos o carpetas de dos maneras. La primera opción es hacer click en el icono de una papelera situado debajo de todas las carpetas y archivos (1). La segunda opción es clickar y arrastrar un icono hasta la papelera situada en la esquina superior derecha de la página (2).

## Visor de imágenes



Fig. 21 Detalle de enlace al visor

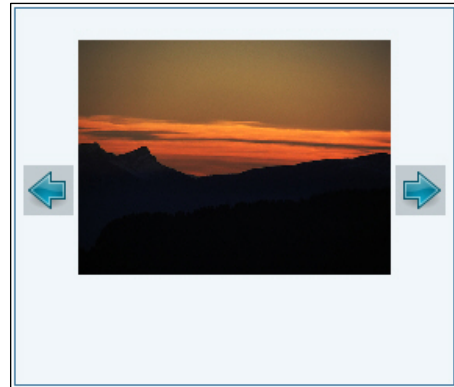


Fig. 22 Detalle del al visor

Si accedemos a una carpeta con al menos una imagen, podemos visualizarla en el visor de imágenes haciendo click en el enlace "Ver imágenes en el visor" (1), esto desplegará una pequeña ventana (Fig. 22) en la que aparece la imagen y va rotando automáticamente.

## Crear un nuevo usuario

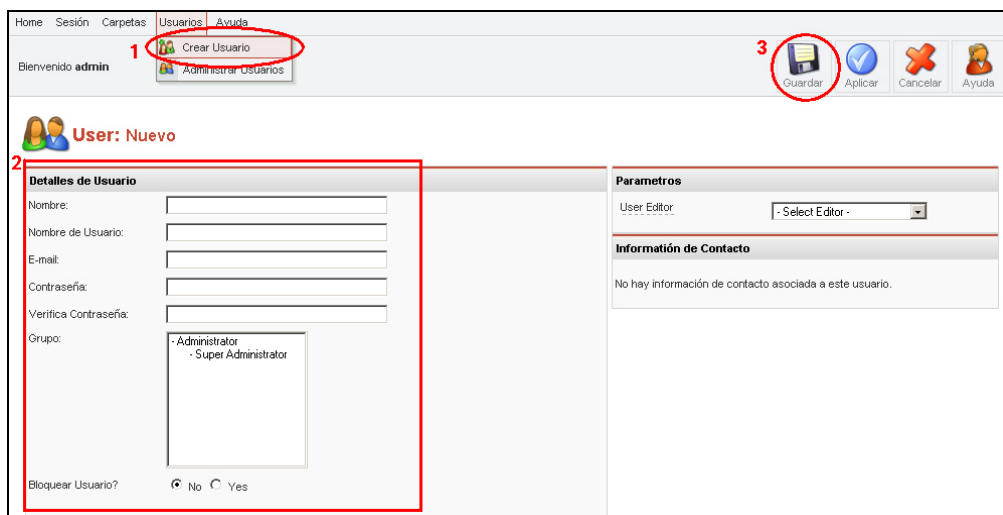
The image shows a web interface for creating a new user. The "Usuarios" menu is highlighted with a red circle and labeled "1". The "Crear Usuario" link is also highlighted with a red circle and labeled "2". The "Guardar" button is highlighted with a red circle and labeled "3". The form contains fields for "Nombre", "Nombre de Usuario", "E-mail", "Contraseña", "Verifica Contraseña", and "Grupo". The "Grupo" dropdown is set to "Administrator". There is also a "Bloquear Usuario?" checkbox with "No" selected.

Fig. 23 Detalle del alta de un usuario

Para dar de alta un nuevo usuario, un administrador debe acceder al formulario de alta en la opción "Crear Usuario" del menú superior (1), rellenar el formulario (2) y hacer click en el icono "Guardar" (3).

## Eliminar o Editar un usuario



Fig. 234 Imágen de la administración de usuarios

Un administrador puede modificar o eliminar un usuario haciendo click en el submenú “Administrar Usuarios” del menú superior (1), seleccionando el usuario que desea modificar o borrar (2) y haciendo click en el icono “Eliminar” o “Editar” (3) según nuestras necesidades.