

## Unidad Didáctica VII.- Generación de aplicaciones para la web (I)

### CASO.

En **SI Informática** ya han finalizado la aplicación de **Gestión del Taller Informático** que le encargó el taller **El PC Feliz**. Tras más de un año de utilización, los encargados de **El PC Feliz**, se vuelven a poner en contacto con nuestros amigos para pedir unas modificaciones. Desean además de añadir unas nuevas funcionalidades, ver si hubiera alguna manera de usar la aplicación, o algunas de sus herramientas, desde fuera de la sede.

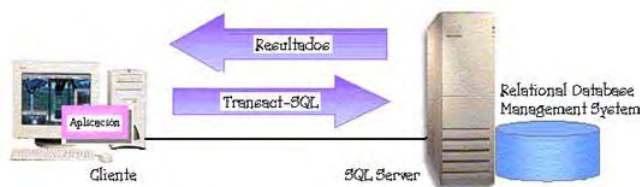


**Carmen**, tras pensar un poco en el nuevo proyecto llegó a la conclusión de que puede que la mejor solución sea el uso de aplicación web. Ya que la aplicación está basada en una base de datos, no es demasiado difícil implementar una aplicación web, que solucione el nuevo problema. Además el tiempo empleado en modificar y actualizar todos los clientes para poder sacarle todo el partido a las nuevas funcionalidades sería mucho menor. Además, esto haría posible el uso de la aplicación, o parte de ella, no sólo desde fuera de la sede, si no desde cualquier rincón del mundo y desde cualquier ordenador con conexión a Internet y un navegador web convencional.

Generación de aplicaciones para la web (I)

### ¿Qué es una aplicación web?

Hasta hace unos años, a la hora de gestionar [aplicaciones distribuidas](#) la mejor y única solución no era otra que el uso de la estructura cliente-servidor, esto es, que la aplicación tenía su propio programa cliente y un interfaz de usuario, los cuales debían de ser instalados en cada estación de trabajo donde deseábamos poder acceder a esa aplicación. Cuando realizábamos una mejora en el servidor, por ejemplo insertar una nueva funcionalidad o una nueva herramienta, requería típicamente una mejora de los clientes instalados en cada una de las estaciones de trabajo, lo cual implicaba un coste de soporte técnico, disminuyendo la eficiencia del personal.



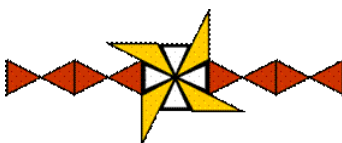
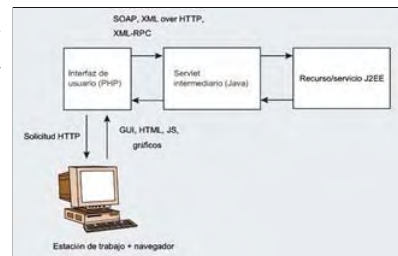
En contraste a esta estructura, las aplicaciones web generan dinámicamente una serie de páginas en un formato estándar, soportado por navegadores web comunes mediante el código HTML. Estas páginas podríamos identificarlas como la aplicación cliente y su interfaz, con el considerable beneficio de no tener que instalar una aplicación cliente en cada una de las estaciones de trabajo de los usuarios, además de que un cambio en el servidor no implica actualizaciones en cada una de estas aplicaciones cliente. Para ello se utilizan lenguajes interpretados del lado del cliente (navegador web) tales como JavaScript, para añadir [elementos dinámicos](#) a la interfaz de usuario y hacerla más interactiva, amigable y eficiente. Generalmente cada página web individual es enviada al cliente como un documento estático, pero la secuencia de páginas proporciona una "experiencia interactiva", es decir da la impresión de que tenemos una aplicación ejecutándose en nuestro equipo que responde a nuestras peticiones inmediatamente.

Generación de aplicaciones para la web (I)

### Pero entonces ¿qué es una aplicación WEB?

Una **aplicación web** es un sistema informático que los usuarios utilizan accediendo a un **servidor web** a través de un sistema de red, ya sea Internet o de una intranet. Las aplicaciones web son populares debido a lo práctico que resulta el navegador web como cliente ligero. La facilidad para actualizar y mantener aplicaciones web sin distribuir e instalar software en miles de potenciales clientes es otra razón de su popularidad.

Las interfaces web tienen ciertas limitaciones en la funcionalidad del cliente. Métodos comunes en las aplicaciones de escritorio como dibujar en la pantalla o arrastrar-y-soltar no están soportadas por las tecnologías web estándar. Para intentar suplir esto, los desarrolladores web comúnmente utilizan lenguajes interpretados del lado del cliente para añadir más funcionalidad, en especial para intentar crear una especie de "recuerdo" de la interacción que no requiera recargar la página continuamente. Recientemente se han desarrollado tecnologías para coordinar estos lenguajes con tecnologías del lado del servidor, como por ejemplo PHP o AJAX, pero esto ya lo veremos más adelante.

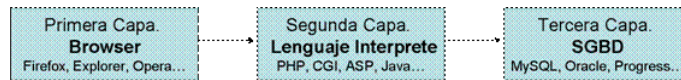


Una ventaja significativa en las aplicaciones web que soporten las características de los navegadores estándar es que deberían funcionar igual independientemente de la versión del sistema operativo instalado en el cliente. En vez de crear clientes específicos para Windows, Mac OS X, GNU/Linux, u otros sistemas operativos, la aplicación es escrita una vez y es mostrada en cualquier navegador web y, por tanto, en cualquier sistema operativo en el que se pueda ejecutar dicho navegador.

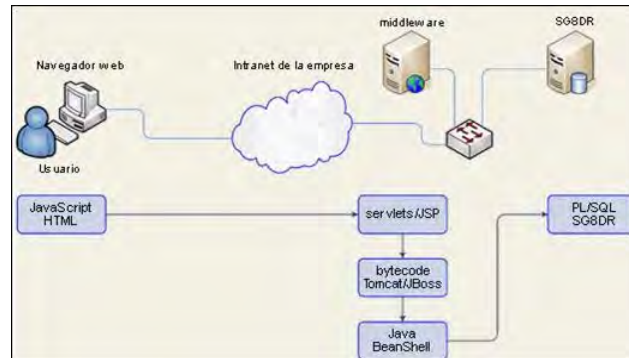
Generación de aplicaciones para la web (I)

## Estructura

Aunque muchas variaciones son posibles, una aplicación web está comúnmente estructurada como una aplicación de tres-capas. En su forma más común:



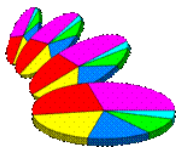
- El navegador web es la primera capa. Como cliente en el que se visualiza el resultado final del trabajo.
- Un motor usando alguna tecnología web dinámica (ejemplo: CGI, PHP, Java Servlets o ASP) es la capa media. Normalmente se sitúa en el servidor y proporciona instrumentos que no es posible utilizar en la capa anterior.
- Una base de datos como última capa. No todas las páginas web tienen esta capa, pero aporta una gran potencia desde el punto de vista del dinamismo y la posibilidad de automatizar muchas de las tareas al hacer uso de datos almacenados, ya sean de configuración o de información para los usuarios.



Si observamos la imagen, vemos cómo el navegador web manda peticiones a la capa media usando HTML y Javascript.

La capa media, **middleware** actúa como interfaz entre el navegador (browser) y la base de datos, traduciendo las solicitudes del usuario en formato HTML a consultas SQL que puede resolver nuestra base de datos.

Finalmente el sistema de gestión base de datos (**SGBD**) es la última capa. Tras recibir una sentencia SQL sobre los datos que tiene almacenados, la ejecuta y pasa el resultado a la capa intermedia, y esta genera una página HTML con el resultado devuelto por la tercera capa.



En las páginas web estáticas, que normalmente se utilizan para publicidad, anuncio de eventos o para mostrar información invariable, sólo hacemos uso de la primera de estas capas.

También existen páginas en las que sólo se hace uso de las dos primeras capas. Es el caso de aquellas que no necesitan utilizar datos almacenados para realizar una tarea, esto ocurre cuando se piden datos de entrada, se realizan cálculos con esos datos y devuelve una serie de valores obtenidos tras esos cálculos, pero no hace uso de datos almacenados. Por ejemplo una calculadora o un conversor de monedas.

Generación de aplicaciones para la web (I)

## Lenguajes más utilizados.

Existen numerosos lenguajes de programación empleados para el desarrollo de Aplicaciones Web, entre los que destacan:

1. **PHP** usa una mezcla entre interpretación y compilación para intentar ofrecer a los programadores la mejor relación entre rendimiento y flexibilidad. Es uno de los lenguajes de programación para Internet más populares actualmente, la gran fluidez y rapidez de sus scripts y su prometedor futuro, lo han convertido en uno de los más importantes.
2. **ASP/ASP.NET** (aunque realmente es una arquitectura que soporta varios lenguajes) es un intento de Microsoft para introducirse en el mercado del desarrollo Web, y viene a ser como su estándar para su servidor Web, Internet Information Server-IIS.
3. **Java**, con sus tecnologías Java Servlets y JavaServer Pages (**JSP**). Hay que decir que Java no es sólo un lenguaje para la programación Web, pero se adapta perfectamente a ella con toda la potencia de un lenguaje de programación completo.
4. **Perl**. Probablemente es la alternativa principal a PHP, es también el más antiguo de los lenguajes para Internet. Es utilizado en muchas de las aplicaciones open source y presenta la ventaja de que existen múltiples módulos ya programados que se pueden utilizar libremente.
5. **Ruby on Rail**. También conocido como "RoR". Sin lugar a dudas es la solución que más está evolucionando, probablemente por su sencillez y funcionalidad. Acorta considerablemente los tiempos de desarrollo con el uso de "Active Record".
6. **Python**. Mantiene desde hace tiempo una rivalidad amistosa con Perl. Algunos consideran que Python es mucho más limpio y elegante para programar. Las aplicaciones se dividen en módulos reutilizables.





#### Para saber más

En los siguientes enlaces podemos encontrar las páginas oficiales (la mayoría en inglés) de los principales lenguajes de programación para la Web. Es conveniente tenerlas localizadas, especialmente la del lenguaje por el que optemos finalmente, porque desde ellas nos pueden resolver muchos problemas y algunas dudas que, sin duda surgirán durante nuestro trabajo. Sobre todo vas a encontrar la documentación en línea de cada uno de ellos.

Aquí encontrarás zona de descarga, documentación, faq, y muchas cosas más, de entre las que destacaría el "php net sites" donde aparece un importante listado de sitios sobre este lenguaje.

[Web oficial del lenguaje PHP](#)

Esta es su versión en español y en ella me gustaría destacar el enlace "aprender", donde puedes encontrar tutoriales, artículos y cursos sobre ASP.

[Web oficial de la plataforma ASP.NET](#)

Es probablemente la más completa de las páginas, porque además de todo lo visto en las anteriores hay un importante apartado sobre Especificaciones de la API.

[Web oficial de Java Server Pages](#)

Está enfocada esta página como un directorio en el que tenemos disponible todo lo relacionado con este lenguaje de programación.

[Web oficial de Perl](#)

Está en español y llama la atención su sencillez. Destacaría el apartado "Ruby en 20 minutos" para aprender a utilizarlo, y realmente son 20 minutos.

[Web oficial de Ruby](#)

Similar a todos los demás sitios oficiales, pero destacaría un vídeo de iniciación (a modo de tutorial) que podemos encontrar en la Documentación.

[Web oficial de Python](#)

Algunos ejemplos de aplicaciones Web los puedes encontrar en:

Gmail.



<http://www.gmail.com/>

Correo electrónico

Skynet ERP.



<http://www.skyneterp.com/>

Software de gestión empresarial

MyWebDesktop.net.



<http://www.mywebdesktop.net/>

Escritorio virtual colectivo

#### Para saber más

En estos enlaces puedes aprender algo más sobre aplicaciones Web, podemos aclarar con algunos ejemplos el concepto, así como ver algunos ejemplos más. En el primero de ellos encontrarás otros enlaces que te van a permitir conocer, entre otras cosas, la Arquitectura Web y algunos aspectos sobre los navegadores.

[Introducción a Aplicaciones Web por Jesús Vegas](#)

En la siguiente biblioteca encontrarás una serie de artículos clasificados por temas para cada uno de los lenguajes de programación (recuerda que debes pulsar sobre el enlace "Leer más...").

[Biblioteca por temas de WebViva](#)

#### Autoevaluación

¿Qué ventajas aporta una aplicación web?

- a) Es distribuida y por tanto podemos usarla desde cualquier ordenador con un navegador y una conexión a Internet.
- b) Si realizamos una modificación en la funcionalidad del servidor, no necesitamos modificar nada en los ordenadores de los usuarios.
- c) Podemos utilizarla en prácticamente cualquier sistema operativo.
- d) Todas las respuestas anteriores son correctas.

[Comprobar](#)

Generación de aplicaciones para la web (I)

#### CASO.

**Antonio** necesita aumentar los beneficios del departamento de ventas, ha pensado que quizás lo mejor sea realizar una página web para abrir un nuevo campo. Llama a **Luis** para pedirle consejo. **Luis** le recomienda que use el lenguaje HTML para la realización de la web, un lenguaje de marcas para texto, estándar para la realización de páginas web ya que es soportado por la totalidad de navegadores del mercado. **Antonio** le comenta la idea a **Víctor**, él cual le confirma sus expectativas.



Además le agrega que tiene gran cantidad de posibilidades y que es capaz de llevar a la realidad prácticamente cualquier idea y es muy fácil de aprender y usar.



Es el acrónimo inglés de **HyperText Markup Language**, que se traduce al español como **Lenguaje de Marcas Hipertextuales (de Hipertexto)**. Es un lenguaje de marcación diseñado para estructurar textos y presentarlos en forma de hipertexto, que es el formato estándar de las páginas web. Gracias a Internet y a los navegadores más comunes, IExplorer, Netscape, Mozilla, Safari, etc., el HTML se ha convertido en uno de los formatos más populares y fáciles de aprender que existen para la elaboración de documentos para la web.

A principios de 1990, Tim Berners-Lee define HTML como un subconjunto del conocido lenguaje de etiquetas **SGML** y crea algo más valioso aún, el World Wide Web. En 1991, Tim Berners-Lee crea el primer navegador de HTML que funcionaría en modo texto para UNIX. Poco a poco han ido apareciendo nuevas versiones, intentando aportar más funcionalidad y dar soporte a las nuevas necesidades. Así apareció HTML2, más conocido como HTML+, HTML3 en le 1995, pero no llegó a presentarse a estándares.

HTML utiliza etiquetas o marcas, que consisten en breves instrucciones de comienzo y final, mediante las cuales se determinan la forma en la que debe aparecer en su navegador el texto, así como también las imágenes y los demás elementos, en la pantalla del ordenador. En la imagen siguiente puedes ver esas etiquetas en azul, pero las vamos a tratar en los siguientes apartados. Haz clic sobre ella para ampliarla.

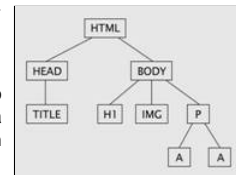


Generación de aplicaciones para la web (I)

### Estructura básica de un documento HTML: Cabecera y cuerpo

Tres son las **tags** que describen la estructura general de un documento y dan una información sencilla sobre él. Estas tags no afectan a la apariencia del documento y solo interpretan y filtran los archivos HTML.

1. **<html>** Limita el documento e indica que se encuentra escrito en este lenguaje.
2. **<head>** Especifica el prólogo del resto del archivo. Son pocas las tags que van dentro de ella, destacando la del título **<title>** que será utilizado por los marcadores del navegador e identificará el contenido de la página. Sólo puede haber un título por documento, preferiblemente corto aunque significativo, y no caben otras tags dentro de él. En head no hay que colocar nada del texto del documento.
3. **<body>** Encierra el resto del documento, el contenido que muestra el navegador, que puede ser texto, imágenes, sonidos, animaciones, enlaces, gráficos..., y en definitiva cualquier objeto que podamos obtener de la red.



```
<html>
Cabecera del documento  <head>
                          <title>Ejemplo 1 </title>
                          </head>
Cuerpo del documento    <body>
                          Hola mundo.
                          </body>
</html>
```

A continuación se muestra un documento HTML que podría utilizarse como plantilla a la hora de crear este tipo de documentos. Puedes ver su contenido con un editor de texto, pero también puedes visualizarlo en un navegador. En este caso, verás cómo se ejecuta esa página en el navegador, es decir, el resultado que produce, pero también podrás ver su codificación mediante el propio navegador (Ver-código fuente, si es IExplorer, o Ver-Código fuente de la página, si tienes Firefox) Te mostramos una imagen con el código, y otra que muestra el aspecto en el navegador. Pinchando sobre la segunda, abrirás el archivo html con tu navegador.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<!--Este es un comentario y no será tomado en cuenta por el navegador -->
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="es" lang="es">
  <head>
    <title>Titulo de la página, que se muestra en la barra azul de titulo</title>
  </head>
  <body>
    <h1>Titulo principal de la página</h1>
    <p>Primer documento XHTML. Hola mundo</p>
  </body>
</html>
```





## Modelo de Plantilla de documento HTML

Generación de aplicaciones para la web (I)

### Creación y edición de código HTML

Podemos decir que los documentos HTML son creados para ser visualizados con unos programas especiales que son los navegadores de Internet. Para crearlos podemos utilizar cualquier editor de texto plano, como puede ser Gedit, GNU Emacs, Bloc de notas, Vim, etc., en el que deberíamos escribir el código HTML con la información que debe aparecer en la página web.

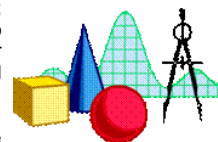


WYSIWYG

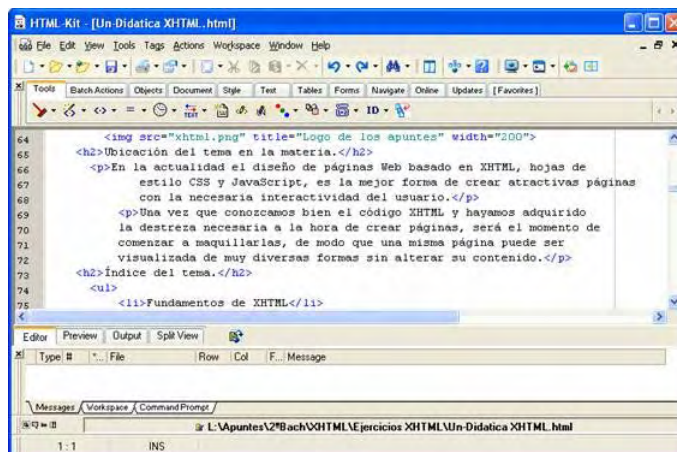
Pero además de poder editar el código así, ahora tenemos a nuestra disposición otros programas para la realización de sitios Web o edición de código HTML, como por ejemplo Microsoft FrontPage, o el famoso software de Macromedia Dreamweaver. Estos programas se les conoce como editores **WYSIWYG** o **What You See Is What You Get** (en español: "lo que ves es lo que obtienes"). Esto significa que son editores en los cuales se ve el resultado de lo que se está editando en tiempo real a medida que se va desarrollando el documento. Ahora bien, esto no significa una manera distinta de realizar

sitios web, sino una forma un tanto más simple ya que estos programas, además de tener la opción de trabajar con la vista preliminar, tiene su propia sección HTML que va generando todo el código HTML a medida que se va trabajando.

Combinar estos dos métodos resulta muy interesante, ya que de alguna manera se ayudan entre sí. Por ejemplo; si se edita todo en HTML y de pronto se olvida algún código o etiqueta, simplemente me dirijo al editor visual o WYSIWYG y se continúa ahí la edición, o viceversa, ya que hay casos en que resulta más rápido y fácil escribir directamente el código de alguna característica que queramos adherirle al sitio, que buscar la opción en el programa mismo.



En cualquier caso, es necesario conocer el código HTML porque además ahora es posible añadir objetos de Internet en nuestras páginas web, tales como vídeos o animaciones.



Otra característica y herramienta muy útil de la que disponemos es la opción de poder ver el código HTML de una página web cualquiera desde la mayoría de los navegadores. Esto lo hacemos haciendo clic derecho sobre el fondo de la página que deseamos visualizar, teniendo cuidado de no hacerlo sobre una imagen u objeto similar, y seleccionando la opción "**Ver código fuente**" si usamos Internet Explorer o "**Ver código fuente de la página**" si usamos Firefox/Mozilla. Para los demás navegadores encontraremos una opción similar al hacer clic derecho sobre el fondo.

Generación de aplicaciones para la web (I)

### Opciones de texto en HTML



Con HTML podemos estructurar el texto con un sin fin de formatos diferentes. Aquí vamos ver unos cuantos ejemplo. En el recurso, podemos ver como se ve cada texto y a su lado que código debemos de insertar.

Las principales etiquetas XHTML relacionadas con el formato del texto, son las siguientes:

**Etiquetas de línea:** mantienen el formato de la línea tras su uso.

**<b> ... </b>**

Resalta el contenido en negrita.

|   |   |
|---|---|
| <code>&lt;u&gt; ... &lt;/u&gt;</code>             | Añade el atributo subrayado al contenido.     |
| <code>&lt;i&gt; ... &lt;/i&gt;</code>             | Añade el atributo cursiva al contenido.       |
| <code>&lt;strong&gt; ... &lt;/strong&gt;</code>   | Contenido al que se le quiere dar más fuerza. |
| <code>&lt;em&gt; ... &lt;/em&gt;</code>           | Agrega énfasis al contenido.                  |
| <code>&lt;address&gt; ... &lt;/address&gt;</code> | El texto se representa como una dirección.    |
| <code>&lt;code&gt; ... &lt;/code&gt;</code>       | Destaca el texto como código.                 |
| <code>&lt;sup&gt; ... &lt;/sup&gt;</code>         | Superíndice.                                  |
| <code>&lt;sub&gt; ... &lt;/sub&gt;</code>         | Subíndice.                                    |

**Etiquetas de bloque:** cuando se utilizan cambia de línea.

|   |                            |
|---|----------------------------|
| <code>&lt;p&gt; ... &lt;/p&gt;</code>                   | Definición de párrafo.     |
| <code>&lt;cite&gt; ... &lt;/cite&gt;</code>             | Cita de textos.            |
| <code>&lt;blockquote&gt; ... &lt;/blockquote&gt;</code> | Texto entre comillas.      |
| <code>&lt;pre&gt; ... &lt;/pre&gt;</code>               | Texto preformateado.       |
| <code>&lt;h1&gt; ... &lt;/h1&gt;</code>                 | Títulos, x varía de 1 a 6. |

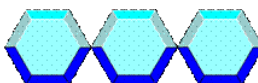
Puedes ver un ejemplo de la variedad de textos y su uso, en el siguiente ejemplo

## Tipos de textos

Generación de aplicaciones para la web (I)

### Añadir enlaces HTML

Los enlaces son una parte vital de HTML y del concepto mismo de la web. Internet es considerada una red debido a cómo los vínculos conectan partes o documentos separados, basados en la idea de una simple referencia sumada a la interactividad. Como resultado, un simple click te lleva desde el documento referente al mencionado en el vínculo.



Los enlaces son los que hacen que naveguemos por la Web, es decir que vayamos saltando por diferentes páginas de un mismo sitio, o de sitios externos. La etiqueta utilizada para crear un vínculo es **a** (**ancla o anclaje**).

No necesariamente un enlace apuntará a una página Web, puede por ejemplo apuntar a una imagen, a cualquier otro tipo de archivo, a un correo electrónico o incluso puede apuntar hacia otros sectores de la misma página (previamente marcando estos sectores).

Veamos un ejemplo sencillo:

```
<a href="http://www.misitio.com">Ejemplo</a>
```

En el navegador lo que veremos será: [Ejemplo](http://www.misitio.com). La dirección **http://www.misitio.com** es el valor del atributo **href**, la palabra **Ejemplo**, encerrada entre **<a>** y **</a>**, es lo que suele llamarse el "**anchor text**" y es el contenido que será visible en el navegador.

Para crear un enlace hacia un correo electrónico debemos apuntar en el valor del atributo **href** hacia una dirección de correo electrónico, pero es importante agregar al principio de la dirección el **mailto:** que vendría a representar el protocolo (**http://**, **ftp://**, etc).

Ejemplo:

```
<a href="mailto:micorreo@misitio.com">Correo de ejemplo</a>
```

Como en el caso anterior, veremos en nuestro navegador: [Correo de ejemplo](mailto:micorreo@misitio.com). Si pruebas el enlace seguramente tu navegador disparará el cliente de correo por defecto.

En el siguiente ejemplo puedes observar cómo el vínculo (o enlace) lo tiene sólo parte del texto, en este caso "tag HTML". En lugar de ese texto también podríamos haber puesto una imagen que en ese caso sería el enlace a la dirección especificada.

#### Código

El `<a href="http://www.juntadeandalucia.es/educacion/"> tag HTML</a>` que nos lleva a una Web.

#### Visualización

El [tag HTML](http://www.juntadeandalucia.es/educacion/) que nos lleva a una Web.

Generación de aplicaciones para la web (I)

### Insertar imágenes en el código

La inserción de imágenes es algo cotidiano en las páginas web. Hacen que nuestra web sea mucho más atractiva y clara, a la hora de explicar algunos conceptos.

Para insertar una imagen en nuestra web debemos de usar la etiqueta **IMG** y en el atributo SRC debemos poner la dirección de la imagen. Esta puede ser una url o una imagen local. Si la imagen se encuentra almacenada en la misma carpeta que la web, basta con añadir el nombre de la misma. Si se encuentra en una subcarpeta debemos nombrar la ruta desde la carpeta en la que se encuentra nuestra web.



#### Código

Ésta es nuestra imagen. `<IMG SRC="boton.jpg">`

#### Visualización



Ésta es nuestra imagen.

Hay una seria discusión acerca de qué formato de imágenes es conveniente utilizar en la Web, que incluye temas variados y complejos cómo:

- algoritmos de compresión de imágenes,
- patentes,
- calidad, y
- compatibilidad.

Existen tres formatos principales de imagen para su utilización en la Web (GIF, JPG y PNG). Para conocer algo más de cada uno de ellos te recomiendo que consultes los siguientes artículos de la Wikipedia:

1. **PNG. (Portable Network Graphics)** formato gráfico basado en un algoritmo de compresión sin pérdida para mapas de bits no sujeto a patentes.
2. **JPEG. (Joint Photographic Experts Group)** es un algoritmo diseñado para comprimir imágenes con 24 bits de profundidad o en escala de grises.
3. **GIF. (Compuserve GIF o Graphics Interchange Format)** es un formato gráfico utilizado ampliamente en la World Wide Web, tanto para imágenes como para animaciones.



Generación de aplicaciones para la web (I)

### Utilización de Listas

Una técnica muy utilizada en páginas Web es la utilización de listas de elementos enumerados, ya que de este modo se aclara la forma de mostrar la información y facilita la visualización de la página al mostrarla más estructurada.

Hay varias formas de representar las listas en XHTML, pueden ser:

- numeradas (ordenadas),
- de viñetas (desordenadas) o
- de definiciones.



#### Listas numeradas

```
<ol> <li> ...
    <ol>

    <li>Manzana</li>

    <li>Plátano</li>

    <li>Naranja</li>

    <li>Melón</li>

</ol>

1. Manzana
2. Plátano
3. Naranja
4. Melón
```

#### Listas de viñetas

```
<ul> <li> ...
    <ul>

    <li>Tomate</li>

    <li>Lechuga</li>

    <li>Aguacate</li>

    <li>Zanahoria</li>

></ul>

■ Tomate
■ Lechuga
■ Aguacate
■ Zanahoria
```

#### Listas de definiciones

```
<dl> <dt> <dd> ...
    <dl>

    <dt>Zumo</dt>

    <dd>Fruta exprimida</dd>

    <dt>Ensalada</dt>

    <dd>Verdura troceada</dd>

</dl>

Zumo
Fruta exprimida
Ensalada
Verdura troceada
```

Como podemos observar en los ejemplos del cuadro, la forma de representar la información es muy clara cuando se utilizan listas, especialmente en las definiciones no es preciso especificar cuál es el término y cuál la definición.

Podemos utilizar el atributo **type** para cambiar el sistema de numeración de las listas ordenadas (los valores soportados por **type** son: **A**, **a**, **I** y **1**, que es el que toma por defecto) y las viñetas en las listas desordenadas (los valores soportados son: **circle**, **disc** y **square**). Sin embargo el cambio de estas propiedades son parte del diseño por lo que no sería correcto su uso en un documento **strict**. Más adelante veremos que con el uso de hojas de estilo **CSS**, podremos usar aún más tipos, y manteniendo la correcta separación del contenido y el diseño del documento.

```
<ol type="a">...</ol>

<ol type="A">...</ol>

<ol type="I">...</ol>

<ol type="1">...</ol>

<ul type="circle">...</ul>

<ul type="disc">...</ul>

<ul type="square">...</ul>
```

Una técnica muy interesante es el anidamiento de listas, en la que una lista puede ser un ítem de otra lista de cualquier otro tipo.

Generación de aplicaciones para la web (I)

### Utilización de tablas



Otra forma de estructurar la información en las páginas es mediante el uso de tablas. Las tablas son una interesante herramienta para el **marcado de información tabular**, otros usos y abusos (como maquetación y diseño) se desaconsejan encarecidamente. De nuevo: **mantener por separado, el diseño del contenido**.

Podemos representar tablas con o sin borde, con una fila, o una columna, o con varias de cada una de ellas. Incluso podemos crear tablas dentro del contenido de una celda de otra tabla, con lo que conseguimos representar tablas multidimensionales, que en determinadas situaciones pueden ser especialmente útiles. En el siguiente documento Web de ejemplo puedes ver todas las posibilidades que acabamos de comentar y acompañamos cada una con el código para construir cada tabla.

## Tablas

Generación de aplicaciones para la web (I)

### La evolución de HTML. XHTML.

HTML se ha visto obligado a una reestructuración debido a la diversificación en la interpretación del código por los diferentes navegadores que existen. Esta reestructuración ha significado una mejora en la estandarización y una mayor rigidez en cuanto a la escritura del código se refiere.

1. Un documento no puede tener etiquetas abiertas, deben ir acompañadas de su correspondiente etiqueta de cierre.
2. Las etiquetas que definen un único elemento deben cerrarse a sí mismas añadiendo al final de la misma un espacio y una barra /. Por ejemplo `<hr />`.
3. Las diferentes etiquetas deben estar debidamente anidados, de forma que se cierre cada etiqueta en orden inverso al de apertura.  
`<table><tr><td>Estamos dentro de una celda</td></tr></table>`
4. Todas las etiquetas y los atributos que las acompañan deben escribirse siempre en minúscula.
5. Los valores de los diferentes atributos de una etiqueta deben aparecer siempre entre comillas.  
`<table width="90%" border="1">`



En el siguiente ejemplo que muestra una unidad didáctica, puedes ver todas estas reglas y un primer ejemplo sobre el uso de hojas de estilo, que puedes desactivar con el enlace que aparece en la parte superior izquierda y comparar el resultado de utilizar las hojas de estilo que veremos más adelante en esta unidad.

#### Ejemplo Unidad Didáctica

##### Para saber más:

En estos enlaces tienes más información sobre html, concepto, uso, ejemplos, así como algún manual de interés.

[Manual de HTML de Webestilo.com](#)

Otro manual de HTML

[Manual de HTML de Desarrolloweb.com](#)

### Autoevaluación

¿Qué tipo de lenguaje es html?

- a) Es un lenguaje de programación imperativo, similar C o Modula
- b) Es un lenguaje de programación orientado a objetos, como Java o C++.
- c) Es un lenguaje de órdenes, para hacer varias tareas sencillas o scripts
- d) Es un lenguaje de marcas, para estructurar texto y darle formato.

Comprobar

¿Cuál de estas afirmaciones es errónea?

- a) HTML es el lenguaje estándar para el desarrollo de páginas web
- b) Las etiquetas o tags nos seleccionan porciones de textos aplicándoles unas propiedades
- c) Con HTML podemos insertar imágenes en nuestras páginas web
- d) Con HTML además de nuestras web, podemos crear pequeños programas, como una calculadora

Comprobar

Generación de aplicaciones para la web (I)

**CASO. Carmen** le comenta a **Víctor** que las páginas Web deben ser interactivas, de modo que el usuario pueda proporcionar datos a la aplicación y ésta mostrarle los resultados. Le explica que en el diseño Web para la introducción de datos se utilizan los formularios, una especie de bloque de código que recoge diferentes modos de entrada de datos, los almacena en variables y posteriormente los envía a la dirección que se especifique, ya sea de correo electrónico o a otra página que tras procesarlos muestra el resultado de su uso.



**Víctor** que está acostumbrado a la interactividad de las aplicaciones de escritorio en las que declara las variables, las inicializa y programa la introducción de los datos, no termina de entender el funcionamiento de los formularios HTML. **Jesús** le aclara que es un modo de programación menos flexible, más cerrada y que sólo tiene que ajustarse a los modos de entrada predeterminados, lo que realmente facilita considerablemente el trabajo.



Los formularios son una característica del estándar HTML que permite recolectar información de los visitantes. Estos formularios pueden ser útiles para obtener información personal, información de contacto, preferencias u opiniones, o cualquier otro tipo de entrada por parte del visitante que el autor pueda necesitar.



Un formulario puede ser insertado en un documento HTML mediante la etiqueta **"form"** la cual actúa como contenedor para todos los elementos de entrada (input).

## La etiqueta Form

Como ya hemos comentado, la etiqueta **"form"** es el elemento con el que introducimos los formularios en nuestra página web.

Los atributos de la etiqueta **"form"** definen información significativa acerca de quién y cómo procesará su información, mientras que los controles se encargan de la entrada de datos.



**id** El atributo **"id"** asigna un identificador al elemento asociado. Este identificador debe ser único en el documento y puede ser usado para referirse a ese elemento

Ejemplo:

```
<p id="parrafo1">
Este es el primer párrafo nombrado parrafo1.
Para cambiar dinámicamente las propiedades del mismo usa este identificador.
</p>
```

**style** Define un estilo visual para el elemento. Es una mejor práctica definir atributos en hojas de estilos externas agrupándolos en clases. Los atributos en el parámetro **"style"** deben preservar este orden **"nombre : valor"** y ser separados por un punto y coma.

Ejemplo:

```
<p style="color: #0000FF; font-size: 12pt">
Este es un párrafo con un estilo definido
</p>
<p>Y este es otro texto sin estilo. </p>
```

**action** Designa la ubicación del archivo que manejará las entradas del formulario. Los formularios son usualmente manejados por scripts del lado servidor.

Ejemplo:

```
<form action="programa.php">
El atributo "method" define cómo la información de un formulario es enviada al agente procesador. Existen dos valores posibles para este atributo (insensibles a mayúsculas/minúsculas):
```

- Method**
1. **get**: Los datos del formulario son agregados a la URL definida en el atributo **"action"**. Por ejemplo: `programa.php?nombre=jhon&apellido=malcovich`
  2. **post**: Los datos del formulario son agregados al cuerpo del formulario.

Ejemplo get:

```
<H1>Ejemplo de procesamiento de formularios</H1>
<FORM ACTION="procesa2.phtml" METHOD="GET">
Introduzca su nombre:<INPUT TYPE="text" NAME="nombre"><BR>
Introduzca sus apellidos:<INPUT TYPE="text" NAME="apellidos"><BR>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
```

Ejemplo post:

```
<H1>Ejemplo de procesamiento de formularios</H1>
<FORM ACTION="procesa2.phtml" METHOD="POST">
Introduzca su nombre:<INPUT TYPE="text" NAME="nombre"><BR>
Introduzca sus apellidos:<INPUT TYPE="text" NAME="apellidos"><BR>
<INPUT TYPE="submit" VALUE="Enviar">
</FORM>
```

**enctype** Especifica el tipo de contenido de la información enviada, cuando el valor del atributo **"method"** es **"post"**. Por ejemplo, el valor **"multipart/form-data"** debería ser usado al enviar archivos mediante un formulario. El valor predeterminado es **"application/x-www-form-urlencoded"**.

Ejemplo:

```
<FORM ACTION="mailto:dirección_de_email" METHOD="POST" ENCTYPE="TEXT/PLAIN">
```

**accept** Define una lista separada por comas de tipos de contenido que el agente procesador manejará correctamente. Esto puede ser utilizado para filtrar archivos a enviar en el lado cliente (por ejemplo, sólo permitir la subida de archivos de imagen comprimidos, como JPG y GIF).

Ejemplo:

```
<form accept="image/gif,image/jpeg">
```

accept-charset Define una lista, separada por espacios o comas, de set de caracteres. Esta lista debería contener los sets de caracteres que el agente procesador puede procesar.

Ejemplo:

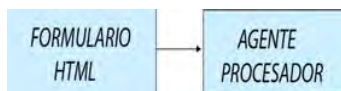
```
<form accept-charset="iso-8859-1,UTF-8">
```

---

Generación de aplicaciones para la web (I)

### Funcionamiento de los formularios

Toda la información recolectada por un formulario puede ser entregada a un agente procesador que es usualmente especificado en el atributo "**action**". Lo que el agente procesador hace con la información y cómo la maneja es un tema que no será tratado en este apartado, dado que no pertenece al estándar HTML. Para manejar datos de un formulario debes usar un scripts del lado servidor.



De esta forma, un formulario simple puede tener la siguiente declaración:

#### Comienzo del código

```
<form method="post" action="programa.php">
```

...Controles...

```
</form>
```

#### Fin del código

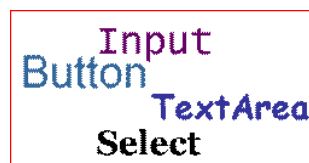
---

Generación de aplicaciones para la web (I)

### Elementos de entrada (input)

La mayoría de los elementos de entrada son visuales y pueden interactuar con el usuario. Su uso depende del tipo de control y también del tipo de información que pueden recolectar. Los elementos de entrada de un formulario pueden ser definidos usando cuatro etiquetas, que son:

- **input**,
- **button**,
- **select** y
- **textarea**.



Observa que el atributo "name" de cada control será el nombre utilizado para identificar los datos para el agente procesador, y el valor dependerá de la naturaleza del control (algunas veces, como en las casillas de verificación o botones radio, será el contenido del atributo "value").

---

Generación de aplicaciones para la web (I)

---

### Entrada de texto

---

Existen tres tipos de entradas de texto que pueden recolectar información textual como nombres, comentarios, etc.

#### 1. Entrada de texto de línea.

Este control recopila información textual en una sola línea, lo que significa que el usuario no podrá utilizar la tecla "enter" para ir a la siguiente línea. Este control es insertado en documentos HTML usando la etiqueta **input** con el valor "**text**" para el atributo "**type**".

#### Código

```
<form method="post" action="programa.php">
<p>Ingresa texto: <input name="textoentrada" type="text" /></p>
</form>
```

#### Visualización

Ingresa texto:



El valor pasado al agente procesador será el texto ingresado por el usuario, o sea el contenido dentro de la caja de texto.

#### 2. Entrada de texto contraseña (password)

Este control actúa exactamente como la entrada de texto de línea, con la excepción de que este control "esconde" los caracteres mostrándolos como puntos o asteriscos para evitar que los usuarios vean el texto ingresado. Es comúnmente usado para el ingreso de contraseñas.

#### Código

```
<form method="post" action="programa.php">
<p>Ingresa contrase&ntilde;a: <input name="contrasena" type="password" /></p>
</form>
```

#### Visualización

Ingresa  
contraseña:

El valor pasado al agente procesador será el texto ingresado por el usuario, o sea el contenido dentro de la caja de texto.

### 3. Entrada de texto de múltiples líneas

Este control permite a los usuarios ingresar texto en una o más líneas. Es insertado utilizando la etiqueta **textarea** y puede ser usado para recolectar párrafos, comentarios, cartas, etc. En esta etiqueta, el contenido será el valor inicial del texto.

#### Código

```
<form method="post" action="programa.php">

  <p>Ingresa tus comentarios:<br />

  <textarea name="comentarios" rows="2" cols="30">

    ...Tus comentarios aquí...</textarea></p>

</form>
```

#### Visualización

Ingresa tus comentarios:

Nota que los atributos **"rows"** y **"cols"** establecen las dimensiones de la caja de texto y son requeridos por algunos **DTDs** (declaración de tipo de documento). El valor pasado al agente procesador será el texto ingresado por el usuario, o sea el contenido dentro de la caja de texto.

Generación de aplicaciones para la web (I)

## Otras Opciones. Marcado y selección.

Los autores también pueden permitir a sus visitantes escoger opciones preestablecidas de una lista. Esto puede lograrse usando uno de los tres controles siguientes, que pueden construir diferentes tipos de listas de opciones.

### 1. Casillas de verificación

La casilla de verificación es una opción simple que puede tomar uno de dos valores: "marcado" o "desmarcado". Las casillas de verificación pueden ser visualmente agrupadas como una lista de opciones, pero cada una de ellas es tratada individualmente. Este control es insertado mediante la etiqueta **input** con el valor **"checkbox"** para el atributo **"type"**.



#### Código

```
<form method="post" action="programa.php">

  <p>Selecciona tus intereses:<br />

  <input name="cbipeliculas" type="checkbox" /> Películas<br />

  <input name="cbilibros" type="checkbox" /> Libros<br />

  <input name="cbiinternet" type="checkbox" /> Internet</p>

</form>
```

#### Visualización

Selecciona intereses:

- ☐ Películas
- ☐ Libros
- ☐ Internet

En este caso, el valor pasado será un valor booleano y representará el estado de verificación de la opción. Dependiendo del agente procesador podría ser **"on/off"**, **"checked/unchecked"**, **"true/false"**, etc.

### 2. Botones de radio

Los botones de radio funcionan en la misma forma que las casillas de verificación con pequeñas diferencias: Los botones radio que comparten un mismo nombre conforman un grupo de opciones donde los usuarios no pueden seleccionar más de una opción a la vez. Esto significa que cuando el usuario selecciona una opción, el resto es automáticamente deseleccionado.

#### Código

```
<form method="post" action="programa.php">

  <p>Selecciona tus intereses:<br />

  <input name="intereses" type="radio" value="rbipeliculas" /> Películas<br />

  <input name="intereses" type="radio" value="rbilibros" /> Libros<br />

  <input name="intereses" type="radio" value="rbiinternet" /> Internet</p>

</form>
```

#### Visualización

Selecciona intereses:

- ☒ Películas
- ☐ Libros
- ☐ Internet

Para los botones radio el valor pasado al agente procesador es el contenido del atributo **"value"**, lo que significa que una lista de opciones con varias opciones sólo pasará un valor.

### 3. Listas desplegables

Estas listas pueden ser insertadas usando tres etiquetas:

- **select** (contenedor principal),
- **option** (declaración de opción) y
- **optgroup** (grupo de opciones).

Este tipo de lista puede ser usado como una lista de botones radio o como una lista de casillas de verificación, dependiendo de la presencia del atributo **"multiple"**.

#### Código

```
<form method="post" action="programa.php">

  <p>Selecciona una sola opción, como en los botones radio:<br />

  <select name="entradasselect">

    <optgroup label="Entradas de texto">

      <option>De una sola línea</option>
```

#### Visualización

```

<option>Contrase&ntilde;a</option>
<option>Multil&iacute;nea</option>
    </optgroup>
    <optgroup label="Opciones">
<option>Casillas de verificaci&on</option>
<option>Botones radio</option>
<option>Listas select</option>
    </optgroup>
</select>
</p>
<p>Selecciona tantas opciones como desees,
    como en las casillas de verificaci&on
    (manteniendo presionada la tecla "Ctrl"):<br />
<select name="entselect[]"multiple="multiple">
    <optgroup label="Entradas de texto">
<option>De una sola l&nea</option>
<option>Contrase&ntilde;a</option>
<option>Multil&iacute;nea</option>
    </optgroup>
    <optgroup label="Opciones">
<option>Casillas de verificaci&on</option>
<option>Botones radio</option>
<option>Listas select</option>
    </optgroup>
</select>
</p>
</form>

```

Selecciona una sola opci&on, como en los botones radio:

☐

Selecciona tantas opciones como desees, como en las casillas de verificaci&on (manteniendo presionada la tecla "Ctrl"):

☐ De una sola l&nea

Para la primera lista sin el atributo **"multiple"**, el valor pasado al agente procesador ser&la opci&on seleccionada, pero para el segundo ejemplo el valor pasado ser& una lista (array) conteniendo las opciones seleccionadas. Es por ello, que para las listas con el atributo **"multiple"** presente, debes agregar al nombre del control **"select"** los corchetes ("[]"). El valor individual pasado en ambos casos es el contenido del atributo **"value"** si es que se encuentra presente, y en su defecto el contenido de la etiqueta.

Generaci&on de aplicaciones para la web (I)

## Botones

Los botones son &tiles en formularios para enviarlos, reestablecerlos o para ejecutar funciones personalizadas. Pueden ser insertados usando la etiqueta **input** (**submit**, **reset** e **image**) o la etiqueta **button** (botones con contenido).

### 1. Botones de env&io

Este tipo de botones env&ia autom&ticamente el formulario al ser presionados. Son insertados usando la etiqueta **input** con el valor **"submit"** para el atributo **"type"**.



#### C&odigo

```

<form method="post" action="programa.php">
<p><INPUT TYPE=submit VALUE="Validar"></p>
</FORM>

```

#### Visualizaci&on

Validar

### 2. Botones de reestablecimiento

Este tipo de botones reestablecen los controles de un formulario a sus valores iniciales al ser presionados. Son insertados usando la etiqueta **input** con el valor **"reset"** para el atributo **"type"**.

#### C&odigo

```

<form method="post" action="programa.php">
<p><INPUT TYPE=reset VALUE="Reiniciar"></p>
</FORM>

```

#### Visualizaci&on

Reiniciar

### 3. Botones de imagen

Los botones de imagen funcionan exactamente como los botones de env&io con la &unica diferencia de que los botones de imagen son

mostrados visualmente con la imagen apuntada en el atributo "src". Estos botones además envían las coordenadas donde el click ha ocurrido (por ejemplo, para un botón de imagen llamado "boton1" las coordenadas serán enviadas con el formulario como "boton1.x" y "boton1.y"). Son insertados usando la etiqueta **input** con el valor "image" para el atributo "type".

#### Código

```
<form method="post" action="programa.php">

<p><input type="image" src="botonOjal.jpg" value="Soy un boton"></p>

</FORM>
```

#### Visualización



#### 4. Botones con contenido

Los botones con contenido pueden ser usados como botones de envío o de reestablecimiento, o pueden no tener ninguna acción preestablecida (dependiendo del valor del atributo "type"), pero sí permiten a los autores insertar contenido dentro de ellos. Esto significa que un trozo de código HTML puede ser mostrado dentro del botón (vínculos, párrafos, texto en negrita, imágenes, etc.).

#### Código

```
<form method="post" action="programa.php">

<p><button type="button">

La <b>etiqueta button</b><br />

permite contenido.<br />

</button></p>

</form>
```

#### Visualización



Generación de aplicaciones para la web (I)

### Entrada de archivos

La entrada de archivos puede ser utilizada para subir archivos al servidor. El control muestra una caja de texto donde el usuario debe especificar la ubicación del archivo (procesada localmente por el navegador) que será enviado hacia el servidor. El control usualmente muestra un botón para elegir el archivo.

Es importante que observes que para formularios con subida de archivos debes especificar el valor "multipart/form-data" en el atributo "enctype" de la etiqueta **form**, tal y como se muestra en el ejemplo.

#### Código

```
<form method="post" action="programa.php" enctype="multipart/form-data">

<p><input name="imagen" type="file" /></p>

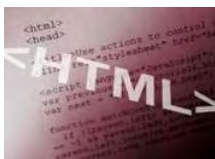
</form>
```

#### Visualización



Generación de aplicaciones para la web (I)

### Etiquetando controles



Las etiquetas de los elementos pueden hacer que tu página se vea mejor y agregar una pequeña funcionalidad para navegadores visuales (cuando un visitante hace click en la etiqueta la acción es transmitida al control asociado). Una etiqueta establece una relación entre un control y un trozo de texto (que se supone ha de dar un "título" para el control).

Las etiquetas pueden ser insertadas usando la etiqueta **label** y son asociadas al control mediante el atributo "for". Para lograr esto, el valor del atributo "for" la etiqueta **label** debe coincidir con el valor del atributo "id" del control.

#### Código

```
<form method="post" action="programa.php">

<div>

<label for="idnombre">Nombre:</label>

<input type="text" id="idnombre" name="nombre" />

<br />

<label for="idapellido">Apellido:</label>

<input type="text" id="idapellido" name="apellido" />

<br />

Género:

<input type="radio" id="idmasculino" name="genero" />

<label for="idmasculino">Masculino</label><br />

<input type="radio" id="idfemenino" name="genero" />
```

#### Visualización

Nombre:

Apellido:

Género:

☐ Masculino

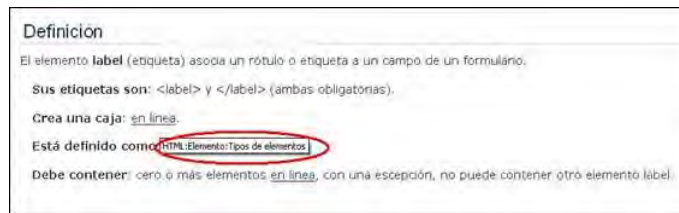
☐ Femenino



```
<label for="idfemenino">Femenino</label>

</div>

</form>
```



## Generación de aplicaciones para la web (I)

### Agrupando elementos

Todos los elementos de un formulario pueden también ser agrupados temáticamente con la etiqueta **fieldset**. Esta etiqueta contiene a un grupo de controles que se relacionan unos con otros de alguna forma (por ejemplo, información personal, información laboral, información financiera, intereses, etc.).

El título para cada grupo de elementos puede establecerse con la etiqueta **legend** que debe ser definido justo a continuación de la etiqueta de apertura del **fieldset**, y proveer un título descriptivo para el grupo de controles.



#### Código

```
<a title=" w3.org " href="http://www.w3.org/TR/REC-html40/">Código HTML</a>
```

#### Visualización

[Código HTML](#)

El uso de estas agrupaciones es amplio y depende de cada formulario específico, pero puede ser muy útil para los visitantes al llenar formularios largos (especialmente para navegadores no visuales).

#### Código

```
<form method="post" action="programa.php">

  <fieldset><legend>Información personal</legend>

    Nombre:

    <input type="text" name="nombre" />

    <br />

    Apellido:

    <input type="text" name="apellido" />

    <br />

    Dirección:

    <input type="text" name="pdireccion" />

    <br />

    Teléfono:

    <input type="text" name="ptelefono" />

  </fieldset>

  <fieldset><legend>Información laboral</legend>

    Direcciónn:

    <input type="text" name="ldireccion" />

    <br />

    Teléfono:

    <input type="text" name="ltelefono" />

  </fieldset>

</form>
```

#### Visualización

Información personal:

Nombre:

Apellido:

Dirección:

Teléfono:

Información laboral:

Dirección:

Teléfono:

#### Para saber más:

En este otro enlace, encontrarás más información sobre el uso de formularios en html.

[Un buen manual sobre formularios de webexperto.com](#)

### Autoevaluación

¿Para qué usarías los formularios en tu página web?

- Para obtener el nombre de usuario y contraseña de acceso a nuestro portal
- Para darle propiedades a una imagen, por ejemplo convertirla en un botón.
- Para obtener un archivo del usuario, por ejemplo una foto.
- Todas las respuestas anteriores son correctas

Comprobar

Generación de aplicaciones para la web (I)

*CASO Pedro habla con Luisa y le comenta que necesita añadir unas opciones a la página web del departamento de ventas. Aunque Pedro es un buen conocedor de Html, advierte que para implementar algunas ideas está teniendo muchos problemas y alguna no ha sido capaz de llevarla a cabo. Sus ideas son para darle más dinamismo a la web y añadirle más interactividad. Luisa le comenta la posibilidad de incorporar **JavaScript** a la web, un lenguaje de órdenes, estándar para la creación de páginas web y soportado por todos los navegadores comerciales. Además de ser muy fácil de usar, **aporta a Html una serie de herramientas que éste no posee haciéndolo más funcional.***



JavaScript, al igual que Java o VRML, es una de las múltiples maneras que han surgido para extender las capacidades del lenguaje HTML. Al ser la más sencilla, es por el momento la más extendida. Antes que nada conviene aclarar un par de cosas:

- JavaScript** no es un lenguaje de programación propiamente dicho. Es un **lenguaje de órdenes** u orientado a documento, como pueden ser los lenguajes de macros que tienen muchos procesadores de texto. Nunca podrás hacer un programa con JavaScript, tan sólo podrás mejorar tu página Web con algunas cosas sencillas (revisión de formularios, efectos en la barra de estado, etc...) y, ahora, no tan sencillas (animaciones usando HTML dinámico, por ejemplo).
- JavaScript** y **Java** son dos cosas distintas. Principalmente porque Java sí que es un lenguaje de programación completo. Lo único que comparten es la misma sintaxis.

**JavaScript** no es un lenguaje orientado a objetos propiamente dicho, ya que no dispone de herencia, es más bien un lenguaje basado en prototipos, ya que las nuevas clases se generan clonando las clases base (prototipos) y extendiendo su funcionalidad.



Todos los navegadores interpretan el código **JavaScript** integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM. El **DOM ( Document Object Model)** es una forma de representar los elementos de un documento estructurado (tal como una página web HTML o un documento XML) como objetos que tienen sus propios métodos y propiedades. El responsable del DOM es el **World Wide Web Consortium**. DOM es una API para acceder, añadir y cambiar dinámicamente contenido estructurado en documentos con lenguajes, como Javascript



Esta imagen es un póster. Pula sobre la imagen para verla a tamaño real y poder leer el texto

Entre las acciones típicas que se pueden realizar en Javascript tenemos dos vertientes.

- Por un lado los **efectos especiales** sobre páginas web, para crear contenidos dinámicos y elementos de la página que tengan movimiento, cambien de color o cualquier otro dinamismo.
- Por el otro, Javascript nos permite **ejecutar instrucciones como respuesta a las acciones del usuario**, con lo que podemos crear páginas interactivas con programas como calculadoras, agendas, o tablas de cálculo.

Javascript es un lenguaje con muchas posibilidades, permite la programación de scripts, pero también de programas más grandes, orientados a objetos, con funciones, complejas estructuras de datos, etc. Toda esta potencia de Javascript se pone a disposición del programador, que se convierte en el verdadero dueño y controlador de cuanto ocurre en la página.

Generación de aplicaciones para la web (I)

### La razón de javascript



Javascript nació para conseguir dotar a html de nuevas funcionalidades que cubrieran la demanda y los requerimientos de los sitios web.

El primer ayudante para cubrir las necesidades que estaban surgiendo fue Java, a través de la tecnología de los Applets, que son pequeños programas que se incrustan en las páginas web y que pueden realizar las acciones asociadas a los programas de propósito general.

Netscape comenzó a desarrollar un lenguaje de programación al que llamó LiveScript que permitiese crear pequeños programas en las páginas y que fuese mucho más sencillo de utilizar que los Applets. De modo que el primer Javascript se llamó LiveScript, pero no duró mucho ese nombre, pues antes de lanzar la primera versión del producto se forjó una alianza con Sun Microsystems, creador de Java, para desarrollar en conjunto ese nuevo lenguaje.



La alianza hizo que Javascript se diseñara como un hermano pequeño de Java, solamente útil dentro de las páginas web y mucho más fácil de utilizar, de modo que cualquier persona, sin conocimientos de programación pudiese adentrarse en el lenguaje y utilizarlo a sus anchas. Además, para programar Javascript no es necesario un kit de

desarrollo, ni compilar los scripts, ni realizarlos en ficheros externos al código HTML, como ocurría con los Applets de Java.

Generación de aplicaciones para la web (I)

### JavaScript se escribe en el documento HTML



Lo más importante y básico que podemos destacar en este momento es que la programación de Javascript se realiza dentro del propio documento HTML. Esto quiere decir que en la página se mezclan los dos lenguajes de programación, y para que estos dos lenguajes se puedan mezclar sin problemas se han de incluir unos delimitadores que separan las etiquetas HTML de las instrucciones Javascript. Estos delimitadores son las etiquetas `<SCRIPT>` y `</SCRIPT>`. Todo el código Javascript que pongamos en la página ha de ser introducido entre estas dos etiquetas.



En una misma página podemos introducir varios scripts, cada uno que podría introducirse dentro de unas etiquetas `<SCRIPT>` distintas. La colocación de estos scripts es indiferente, en un principio nos da igual dónde colocarlos, pero en determinados casos esta colocación si que será muy importante. En cada caso, y llegado el momento se informará de ello convenientemente.

Generación de aplicaciones para la web (I)

### Cómo Ejecutamos nuestros scripts

Existen **dos maneras de ejecutar scripts** en la página. La primera de estas maneras se trata de ejecución directa de scripts, la segunda es una ejecución como respuesta a la acción de un usuario.

#### - Ejecución directa:

Es el método de ejecutar scripts más básico. En este caso se incluyen las instrucciones dentro de la etiqueta `<SCRIPT>`, tal como hemos comentado anteriormente. Cuando el navegador lee la página y encuentra un script va interpretando las líneas de código y las va ejecutando una después de otra. Llamamos a esta manera ejecución directa, pues cuando se lee la página se ejecutan directamente los scripts.



#### - Respuesta a un evento:



Los eventos son acciones que realiza el usuario. Los programas como Javascript están preparados para atrapar determinadas acciones realizadas, en este caso sobre la página, y realizar acciones como respuesta. De este modo se pueden realizar programas interactivos, ya que controlamos los movimientos del usuario y respondemos a ellos. Existen muchos tipos de eventos distintos, por ejemplo la pulsación de un botón, el movimiento del ratón o la selección de texto de la página.

Las acciones que queremos realizar como respuesta a un evento se han de indicar dentro del mismo código HTML, pero en este caso se indican en atributos HTML que se colocan dentro de la etiqueta que queremos que responda a las acciones del usuario.

#### Para saber más

**En los siguientes enlaces podrás encontrar información sobre JavaScript. Muy recomendable si te interesa el tema, dada la pequeña pincelada vista. Además encontrarás manuales de mucha utilidad.**

[Manual de javascript por desarrolloweb.com](#)

[Manual de javascript por webestilo.com](#)

[Más información en la Wikipedia](#)

### Autoevaluación

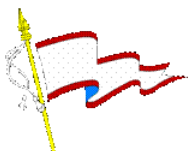
¿Cómo podemos ejecutar nuestros scripts?

- a) A través de la línea de comandos
- b) Sólo como respuesta a un evento, por ejemplo al pulsar un botón
- c) Como respuesta a un evento, por ejemplo al pulsar un botón, y como parte del código, al ir mostrando la página
- d) Como respuesta a un evento, por ejemplo al pulsar un botón, como parte del código, al ir mostrando la página, y ante una llamada al comando `exeScript`

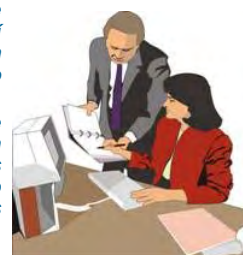
Comprobar

Generación de aplicaciones para la web (I)

**CASO.** *Silvia le comenta a Víctor que ha decidido modificar el estilo de la Web del departamento de marketing. Le explica la gran cantidad de trabajo que le está llevando cambiar los tipos de letras, así como las imágenes corporativas. Además el cambio de estilo de algunos marcos le ha supuesto un gran esfuerzo en cada página. Le explica que aunque no son unos cambios difíciles de llevar a cabo sí es bastante tedioso hacerlo en cada página individual.*



*Víctor, que ya lleva bastante tiempo trabajando en diseño web, le propone a Silvia la posibilidad de incorporar **hojas de estilo en cascada** un mecanismo simple que nos ayuda a definir un tipo de estilo para todas nuestras páginas web, ya que nos asocia trozos de texto con estilos, con lo cual, cambiando el estilo, cambiamos todos los textos que tengamos asociados a éste.*



Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre estilo y formato de sus documentos.

CSS, nos permite crear páginas web de una manera más precisa y homogénea. Gracias a las CSS controlamos más aún los resultados finales de la página, pudiendo hacer muchas cosas que no se podían o eran difíciles de hacer utilizando solamente HTML, como incluir márgenes, tipos de letra, fondos, colores, etc.



Esta imagen es un póster. Pula sobre la imagen para verla a tamaño real y poder leer el texto

Las Hojas de Estilo en Cascada son un estándar muy amplio que actualmente cuenta con tres especificaciones por parte de W3C:

- En 1996 el W3C sacó la **Recomendación de nivel 1** (CSS1-Cascading Style Sheets Specification, level 1. <http://www.w3.org/TR/REC-CSS1-961217.html>)
- En 1998 apareció la **Recomendación de nivel 2** (Cascading Style Sheets, level 2, CSS2 Specification. <http://www.w3.org/TR/REC-CSS2/>)
- En 2004 hay varios trabajos y especificaciones en desarrollo candidatos a **Recomendación de nivel 3**. (<http://www.w3.org/Style/CSS/current-work>).

El lenguaje de hojas de estilo de nivel 1 permite a los autores y usuarios adjuntar estilo (por ejemplo, fuentes, espaciado, etc. a la estructura de los documentos HTML y a las aplicaciones XML. Al separar el estilo de presentación de los documentos del contenido, CSS simplifica la creación de páginas web y su mantenimiento.



CSS2 se construye sobre CSS1 y, con unas pocas excepciones, todo lo que era válido en las hojas de estilo CSS1 es válido para CSS2. CSS2 soporta especificaciones media de estilo así que los autores pueden diseñar la presentación de sus documentos para los navegadores gráficos, impresoras y otros dispositivos. Esta recomendación también soporta posicionamiento del contenido, descarga de fuentes, disposiciones de pantalla, herramientas de internacionalización, contadores automáticos y otras propiedades relacionadas con la interfaz del usuario.

La especificación **CSS3** está todavía en construcción, ya que consta de una serie de módulos como Hyperlink Presentation Module, Text Module, Color Module, Borders Module, List Module, Speech Module, etc.

Generación de aplicaciones para la web (I)

## Ejemplos de etiquetas

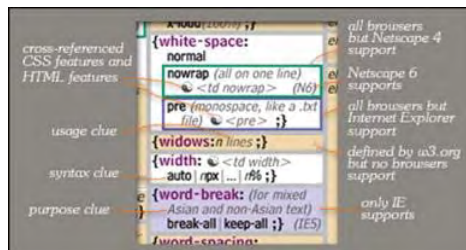


Las Hojas de Estilo en Cascada se pueden escribir dentro del código HTML (o XML) de la página web o en un archivo aparte, en este último caso hay que enlazar la página con ese archivo mediante una declaración en archivos externos.

La manera más directa de aplicar estilos a los elementos de la página se hace a través de la etiqueta HTML **style**.

`<b>&#60;p style="color:blue;font-weight:bold"&#62;El párrafo saldrá con color azul y en negrita&#60;/p&#62; </b>`

Dentro del atributo **style** se deben indicar los atributos de estilos CSS separados por punto y coma (;). En el ejemplo anterior hemos utilizado el atributo **color** (indica el color del contenido) y el atributo **font-weight** (indica el grosor del texto. **Bold** sirve para ponerlo en negrita).



Pero existen otros muchos atributos tales como:

- **Color en los enlaces:** con HTML definimos el color de los enlaces en la etiqueta **<body>**, con lo atributos **link**, **vlink** y **alink**. Esto nos permite cambiar el color de los enlaces para todo el documento. Si queremos cambiar el color de un enlace en concreto, por ejemplo, ponerlo en rojo para que tenga otro color que el definido en la etiqueta **<body>** se utiliza el atributo **style** dentro del enlace:

`<b>&#60;a href="mienlace.html" style="color:red"&#62; </b>`

- **Espaciado entre líneas:** con CSS podemos definir el espacio que hay entre cada línea del documento, utilizando el atributo **line-height**. Por ejemplo, podemos definir que para todo un párrafo el espacio entre cada una de sus líneas



sea 40 píxeles:

```
<b style="line-height: 40px;"></b>
```

Generación de aplicaciones para la web (I)

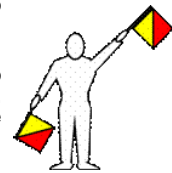
### Ventajas de las hojas de estilo

Las hojas de estilo son muy útiles para grandes proyectos en la World Wide Web como la edición de una enciclopedia en línea compuesta de miles de páginas y referencias, o para la edición de un periódico online que se deba actualizar diariamente. De esta forma se pueden crear muchas páginas con un diseño homogéneo y es más fácil su mantenimiento y actualización.

Así pues, para indicar que en un documento HTML se quiere incluir un rasgo de estilo que está incluido en una hoja de estilo, existen 3 formas diferentes:

1. **Desde el atributo "style" de cada elemento:** el estilo sólo afectará al elemento al que está asociado y tomará el diseño de la hoja de estilo marcada por defecto en el atributo "content" del elemento "meta". Casi todos los elementos pertenecientes al cuerpo (**body**) del documento pueden llevar este atributo. Por ejemplo, para señalar el tamaño de letra y el color del texto de un párrafo, escribiríamos:  
`<P style="font-size: 14 pt; color: blue"> Esta es una web sobre hipertexto.`
2. **Desde la cabecera del documento al que se apliquen las normas de estilo:** se incluye el elemento "style" dentro del elemento "head". Por ejemplo,  
`<STYLE type="text/css" [aquí se incluyen todas las normas de estilo que afectarán al documento]></STYLE>`
3. **Desde un documento aparte al que se hace referencia** desde el elemento de la cabecera "link" y su atributo de relación "rel="stylesheet". Quedaría de la siguiente manera:  
`<LINK rel="stylesheet">`

Esta opción es la más adecuada si se van a utilizar las mismas normas de estilo en varios documentos, puesto que nos evitamos repetir esa información y además, permite dar una uniformidad y coherencia al diseño. Además, un cambio en la hoja de estilo afectará a todos los documentos que la referencien, y así evitaremos tener que modificar los documentos uno a uno.



Pero no sólo los autores pueden crear hojas de estilo, también los usuarios pueden, en parte, elegir cómo se va a presentar la información mediante la utilización de una hoja de estilo predeterminada o seleccionada en el navegador.

### Ejemplo de uso de hojas de estilo

#### Para Saber Más.

En los siguientes enlaces puedes encontrar más información sobre las posibilidades de CSS, así como algunos ejemplos de interés:

[Guía breve de CSS por w3c](#)

[Manual de CSS por webestilo.com](#)

[Manual de CSS por desarroloweb.com](#)

### Autoevaluación

De las siguientes afirmaciones, ¿cuál no es cierta?

- a) Las hojas de estilo son muy útiles para llevar un modelo común en toda nuestra web
- b) Las hojas de estilo aunque aumenta la carga de trabajo para cada página, son muy útiles a la larga ante posibles cambios de estilo
- c) Las hojas de estilo sólo se pueden añadir en un archivo aparte, para que puedan ser importadas desde cualquier lugar
- d) Una modificación en un estilo de un texto, afecta a todos los textos asociados a ese estilo

Comprobar

Generación de aplicaciones para la web (I)

#### CASO.

Andrés está intentando mejorar la aplicación web desarrollada para el departamento de ventas. Hablando con Sara, le explica que la carga de la web es bastante lenta, especialmente para mostrar los productos. Además el tiempo entre página y página se hace un tanto insoportable y se pierde un poco la interactividad. Tras examinar la aplicación a fondo, dice que cree que la única solución es mejorar el servidor. Sara, que le encanta investigar sobre nuevas tecnologías, le comenta la posibilidad de utilizar AJAX, una técnica para el desarrollo de aplicaciones web que fomenta la interactividad reduciendo los tiempos de carga, aprovechando los tiempos en los que se espera una respuesta por parte del usuario. José le comenta que además eso le supondrá un cambio en el diseño de la aplicación pero se ahorrará el gran coste de un servidor nuevo.



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



AJAX es una combinación de tres tecnologías ya existentes:

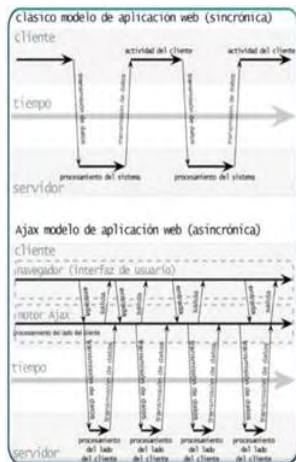
1. **XHTML** (o **HTML**) y hojas de estilos en cascada (**CSS**) para el diseño que acompaña a la información.
2. **Document Object Model (DOM)** accedido con un lenguaje de scripting por parte del usuario, especialmente implementaciones ECMAScript como JavaScript y JScript, para mostrar e interactuar dinámicamente con la información presentada.
3. **XML** es el formato usado comúnmente para la transferencia de vuelta al servidor, aunque cualquier formato puede funcionar, incluyendo HTML preformateado, texto plano, JSON y hasta EBML.



Destacar el uso del objeto **XMLHttpRequest** para intercambiar datos asincrónicamente con el servidor web.

Generación de aplicaciones para la web (I)

### Modelo tradicional para la Web VS Modelo AJAX para la Web



El **modelo clásico** de aplicaciones Web funciona de la siguiente forma: La mayoría de las acciones del usuario en la interfaz disparan un requerimiento HTTP al servidor web. El servidor efectúa un proceso (recopila información, procesa números, hablando con varios sistemas propietarios), y le devuelve una página HTML al cliente.



Una aplicación **AJAX** elimina la naturaleza "arrancar-frenar- arrancar-frenar" de la interacción en la Web introduciendo un intermediario -un motor AJAX- entre el usuario y el servidor.

En vez de cargar un página Web, al inicio de la sesión, el navegador carga al motor AJAX (escrito en JavaScript). Este motor es el responsable de [renderizar](#) la interfaz que el usuario ve y de comunicarse con el servidor en nombre del usuario.

El motor AJAX permite que la interacción del usuario con la aplicación suceda asincrónicamente (independientemente de la comunicación con el servidor). Así el usuario nunca estará mirando una ventana en blanco del navegador y un icono de reloj de arena esperando a que el servidor haga algo.

Cada acción de un usuario que normalmente generaría una petición HTTP toma la forma de una llamada JavaScript al motor AJAX en vez de esa petición. Cualquier respuesta a una acción del usuario que no requiera una viaje de vuelta al servidor es manejado por su cuenta.

Si el motor necesita algo del servidor para responder (sea enviando datos para procesar, cargar código adicional, o recuperando nuevos datos) hace esos pedidos asincrónicamente, usualmente usando XML, sin frenar la interacción del usuario con la aplicación.

Generación de aplicaciones para la web (I)

### Ejemplos

En este apartado vas a ver algunos ejemplos de aplicaciones web que usan AJAX para proporcionar una respuesta rápida al usuario.

- [Google Maps](#). Podemos ver cómo cambiamos la posición del mapa sin recargar la página. Es decir una vez que solicitamos la web, en nuestro navegador se carga el motor AJAX. Si solicitamos, por ejemplo Almería, éste nos cargará primeramente la zona deseada y posteriormente, mientras observamos, nos cargará desde el servidor las zonas colindantes, resto de Andalucía, Murcia, Castilla la Mancha... que son, con alta probabilidad, las zonas que puede que deseemos mirar posteriormente. Esto hace que si desplazamos nuestra selección hacia el norte, Murcia, como ya se encuentra cargada mientras observamos Almería, no debamos de hacer la petición al servidor, haciendo que el movimientos sea más suave y el uso de la aplicación mucho más dinámico, sin tener tiempos de espera mientras solicitamos la nueva zona al servidor. Está claro que si solicitamos una zona ajena, por ejemplo Nueva York, si tendremos que solicitarlo al servidor y esperar la carga de los datos, pero es mucho más probable que desde Almería miremos Murcia o Granada que Nueva York.



Esta imagen es un póster. Pulsa sobre la imagen para verla a tamaño real y poder leer el texto.

- [Google Suggest](#). Los términos sugeridos se van actualizando a medida que uno teclea, casi instantáneamente. Una vez cargamos la web, se carga el motor Ajax. Este a medida que tecleamos lo que deseamos ver, nos va mostrando las posibilidades que dispone, pero sin tener que hacer las peticiones al servidor, es decir, aunque no nos cargue los distintos artículos, si nos carga el índice, el cual podemos ver sin necesidad de hacer nuevas peticiones al servidor.



Esta imagen es un póster. Pulsa sobre la imagen para verla a tamaño real y poder leer el texto

**Para Saber más:**

**Algo más sobre Ajax, muy interesante a la hora de realizar nuestras aplicaciones web más serias. Especial interés tiene el último enlace.**

**Tutorial de Ajax por Cristalab.com**

**[Tutorial de Ajax por Cristalab.com](#)**

**[Breve iniciación a Ajax por Mozilla Developer Center](#)**

## Autoevaluación

De las siguientes afirmaciones, ¿cuál no es cierta?

- a) El uso de la tecnología Ajax nos permite mejorar la visión de nuestras páginas mejorando la interactividad
- b) El uso de la tecnología Ajax puede que haga peticiones al servidor que luego no se utilicen
- c) El uso de la tecnología Ajax requiere una conexión de mayor velocidad que si no la usamos
- d) El uso de la tecnología Ajax nos reduce los tiempos de carga de nuestras aplicaciones web en la mayoría de los casos

Comprobar

Generación de aplicaciones para la web (I)

**CASO.** Antonio está pensando en la posibilidad de poner una aplicación web para realizar la formación del nuevo software de administración. Le comenta a Sara la posibilidad, ya que lleva tiempo dedicada a la educación a distancia. Sara le aconseja la posibilidad de usar un gestor de contenidos, ya que facilita mucho a implementación de la idea, además de aportar una gran cantidad de herramientas extras muy útiles. Da soporte a una gran cantidad de usuarios y permite la interacción de ellos de un modo eficiente. Sara le comenta que incorpora unas medidas de seguridad muy buenas, para evitar el acceso a sitios de acceso protegido.



Un Sistema de Gestión de Contenidos es un sistema de software para ordenador que permite organizar y facilitar la creación de documentos y otros contenidos de un modo cooperativo. Con frecuencia, un CMS es una **aplicación web** usada para **gestionar sitios web y contenidos web**.

Generación de aplicaciones para la web (I)

## Introducción a Moodle

**Moodle** es un sistema de gestión de cursos de libre distribución (**Course Management System - CMS**) que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Moodle fue creado por Martin Dougiamas, quien fue administrador de WebCT en la Universidad Tecnológica de Curtin. Basó su diseño en las ideas del constructivismo en pedagogía que afirman que el conocimiento se construye en la mente del estudiante en lugar de ser transmitido sin cambios a partir de libros o enseñanzas y en el [aprendizaje colaborativo](#). Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que le ayuda a construir ese conocimiento con base en sus habilidades y conocimientos propios en lugar de simplemente publicar y transmitir la información que se considera que los estudiantes deben conocer.



La primera versión de la herramienta apareció el 20 de agosto de 2002 y, a partir de allí han aparecido nuevas versiones de forma regular. Hasta diciembre de 2006, la base de usuarios registrados incluye más de 19.000 sitios en todo el mundo y está traducido a más de 60 idiomas. El sitio más grande dice tener más de 170.000 estudiantes.

La palabra Moodle era al principio un acrónimo de **Modular Object-Oriented Dynamic Learning Environment (Ambiente de Aprendizaje Dinámico Modularmente Orientado a Objetos)**, lo que tiene algún significado para los programadores y teóricos de la educación, pero también se refiere al verbo anglosajón **Moodle**, que describe el proceso de deambular perezosamente a través de algo, y hacer las cosas cuando se antoja hacerlas, una placentera chapuza que a menudo lleva a la comprensión y la creatividad. Las dos acepciones se aplican a la manera en que se desarrolló **Moodle** y a la manera en que un estudiante o profesor podría aproximarse al estudio o enseñanza de un curso en línea.

Generación de aplicaciones para la web (I)

## Características generales de Moodle

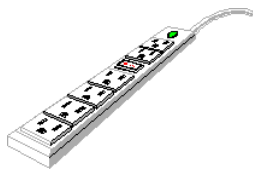
Moodle está desarrollado principalmente en Linux usando Apache, MySQL y PHP (también conocida como plataforma LAMP), aunque es probado regularmente con PostgreSQL y en los sistemas operativos Windows XP, MacOS X y Netware 6.

Los requerimientos de Moodle son los siguientes:

1. **Un servidor web.** La mayoría de los usuarios usan Apache, pero Moodle debe funcionar bien en cualquier servidor web que soporte PHP, como el IIS (Internet Information Server) de las plataformas Windows.
2. **Una instalación de PHP** en funcionamiento (versión 4.3.0 o posterior). PHP 5 está soportado a partir de Moodle 1.4.
3. **Una base de datos: MySQL o PostgreSQL**, que están completamente soportadas y



recomendadas para su uso con Moodle. MySQL es la elección preferida para mucha gente porque es muy popular, pero hay algunos argumentos a favor de PostgreSQL, especialmente si está planificando instalaciones de grandes dimensiones. MySQL 4.1.16 es la versión mínima para trabajar con Moodle 1.6



Se ha puesto énfasis en una seguridad sólida en toda la plataforma. Todos los formularios son revisados, las cookies cifradas, etc. La mayoría de las áreas de introducción de texto (materiales, mensajes de los foros, entradas de los diarios, etc.) pueden ser editadas usando el editor HTML, tan sencillo como cualquier editor de texto de Windows.

Podemos dividir la plataforma en 3 grandes grupos que podemos administrar y configurar:

---

Generación de aplicaciones para la web (I)

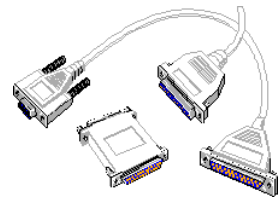
---

### Características de administración del sitio

---

Las características de administración que ofrece **Moodle** son:

1. Administración general por un usuario administrador, definido durante la instalación.
2. Personalización del sitio utilizando "temas" que redefinen los estilos, los colores del sitio, la tipografía, presentación, etc.
3. Pueden añadirse nuevos módulos de actividades a los ya instalados en Moodle.
4. Los paquetes de idiomas permiten una localización completa de cualquier idioma. Estos paquetes pueden editarse usando un editor integrado. Actualmente hay paquetes de idiomas para 35 idiomas.
5. El código está escrito de forma clara en PHP bajo la licencia GPL, fácil de modificar para satisfacer sus necesidades.



---

Generación de aplicaciones para la web (I)

---

### Características de Administración del usuario

---

Moodle soporta un rango de mecanismos de autenticación a través de módulos de autenticación, que permiten una integración sencilla con los sistemas existentes.

Las características principales incluyen:

1. **Método estándar de alta por correo electrónico:** los estudiantes pueden crear sus propias cuentas de acceso. La dirección de correo electrónico se verifica mediante confirmación.
2. **Método LDAP:** las cuentas de acceso pueden verificarse en un servidor LDAP. El administrador puede especificar qué campos usar.
3. **IMAP, POP3, NNTP:** las cuentas de acceso se verifican contra un servidor de correo o de noticias (news). Soporta los certificados SSL y TLS.
4. **Base de datos externa:** Cualquier base de datos que contenga al menos dos campos puede usarse como fuente externa de autenticación.
5. **Cada persona necesita sólo una cuenta para todo el servidor.** Por otra parte, cada cuenta puede tener diferentes tipos de acceso.
6. Con **una cuenta de administrador** que controla la creación de cursos y determina los profesores, asignando usuarios a los cursos.
7. **Seguridad:** los profesores pueden añadir una "clave de acceso" para sus cursos, con el fin de impedir el acceso de quienes no sean sus estudiantes. Pueden transmitir esta clave personalmente o a través del correo electrónico personal, etc. Los profesores pueden dar de baja a los estudiantes manualmente si lo desean, aunque también existe una forma automática de dar de baja a los estudiantes que permanezcan inactivos durante un determinado período de tiempo (establecido por el administrador).



Cada usuario puede especificar su propia zona horaria, y todas las fechas marcadas en Moodle se traducirán a esa zona horaria (las fechas de escritura de mensajes, de entrega de tareas, etc.). También cada usuario puede elegir el idioma que se usará en la interfaz de Moodle (Inglés, Francés, Alemán, Español, Portugués, etc.)

---

Generación de aplicaciones para la web (I)

---

### Características de Administración de cursos

---

El profesor tiene control total sobre todas las opciones de un curso. Se puede elegir entre varios formatos de curso tales como semanal, por temas o el formato social, basado en debates.

En general Moodle ofrece una serie flexible de actividades para los cursos: foros, diarios, cuestionarios, materiales, consultas, encuestas y tareas. En la página principal del curso se pueden presentar los cambios ocurridos desde la última vez que el usuario entró en el curso, lo que ayuda a crear una sensación de comunidad.

La mayoría de las áreas para introducir texto (materiales, envío de mensajes a un foro, entradas en el diario, etc.) pueden editarse usando un editor HTML WYSIWYG integrado.





Todas las calificaciones para los foros, diarios, cuestionarios y tareas pueden verse en una única página (y descargarse como un archivo con formato de hoja de cálculo). Además, se dispone de informes de actividad de cada estudiante, con gráficos y detalles sobre su paso por cada módulo (último mensajes enviados, entradas en el diario, etc. en una sola página).

Pueden enviarse por correo electrónico copias de los mensajes enviados a un foro, los comentarios de los profesores, etc. en formato HTML o de texto.

Generación de aplicaciones para la web (I)

### Ventajas y desventajas.

Naturalmente, Moodle, como cualquier herramienta, tiene sus ventajas y desventajas, y es interesante que las conozcas. Imaginamos que tú, que te encuentras estudiando este curso y posiblemente otros alojados sobre moodle, serás capaz de anticipar algunas de esas características. ¿Te atreves?

#### ■ Ventajas:

Una de las características más atractivas de Moodle, que también aparece en otros gestores de contenido educativo, es la posibilidad de que los alumnos participen en la creación de glosarios, y en todas las lecciones se generan automáticamente enlaces a las palabras incluidas en éstos.

Además debemos reseñar su **alta disponibilidad**, al accederse desde un navegador web, su **escalabilidad**, ya que puede albergar una gran cantidad de cursos tanto como dar soporte a una gran cantidad de usuarios, su **facilidad de uso**, gracias a su sencillez, procesos automáticos y personalizados.

#### ■ Desventajas:

Algunas actividades pueden ser un poco mecánicas, dependiendo mucho del diseño instruccional para Moodle lo que lo hace más dinámico y atractivo. Por estar basado en tecnología PHP la configuración de un servidor con muchos usuarios debe ser cuidadosa para obtener el mejor rendimiento.



#### Para saber más:

*Aquí podemos encontrar información sobre cómo hacer nuestra propia plataforma de aprendizaje con Moodle, especial interés el primer enlace para aclarar conceptos.*

[Un video que nos introduce en el concepto de moodle](#)

[Documentación de moodle.org](#)

[Un buen curso de la Universitat Rovira i Virgili](#)



### Autoevaluación

De las siguientes afirmaciones, ¿cuál no es cierta?

- Moodle está desarrollado sobre Linux, Apache, MySQL y PHP
- Moodle es software propiedad de una empresa, que lo distribuye a muy bajo precio y gratuitamente si es para uso académico
- Moodle al ser una aplicación web, se puede acceder desde cualquier navegador.
- Moodle es un sistema de gestión de cursos.

Comprobar

Generación de aplicaciones para la web (I)

### JOOMLA!

**Joomla!** es un **Sistema de Gestión de Contenidos (CMS)** reconocido mundialmente, de gran ayuda para construir sitios web y otras aplicaciones en línea potentes. Lo mejor de todo, es que Joomla! es una solución de código abierto y está disponible libremente para cualquiera que desee utilizarlo.

La primera versión de Joomla! (Joomla! 1.0.0) fue publicada en septiembre de 2005. Se trataba de una versión mejorada de Mambo. Actualmente se ha publicado Joomla! 1.5b bajo un código completamente reescrito y construido totalmente bajo PHP.



Generación de aplicaciones para la web (I)

### Funcionamiento de Joomla

**Joomla** está programado en lenguaje PHP (Hypertext Pre Processor) y SQL (Structure Query Language). Utiliza bases de datos relacionales, más específicamente MySQL. Tanto PHP como MySQL son programas open-source de libre distribución y uso, y al ser **Joomla** una aplicación WEB, funciona obviamente en servidores de páginas web (HTTP Servers).



Basados en esta misma filosofía open-source podemos afirmar también que **Joomla** correrá mejor en datacenters cuyas plataformas sean Linux/Unix y cuyos servidores HTTP sean Apache. Con esto no vamos a decir que **Joomla** no funciona en plataformas Windows con servidores IIS (Internet Information Server, de Microsoft), pero sí que lo hace con toda su potencia y funcionalidades en plataformas Unix/Apache.

El funcionamiento de **Joomla** se lleva a cabo gracias a sus dos principales elementos:

1. **La base de datos Mysql:** allí es donde se guarda toda la información y la mayor parte de la configuración del sistema, de una forma ordenada y en distintas tablas, cada una de las cuales almacena información específica y determinada.
2. **Los scripts PHP:** son los que ejecutan las acciones de consulta y realizan modificaciones en la base de datos, convirtiendo los datos en simples páginas web interpretables por los navegadores de Internet (Browsers) y perfectamente inteligibles para los usuarios navegantes y administradores.

**BD MySQL**

*Scripts PHP*

Existen también otro tipo de archivos que realizan importantes tareas dentro de **Joomla** (archivos XML, scripts Javascript JS, CSS, etc.), pero el motor fundamental de todo CMS (y de **Joomla** en particular) son los dos elementos enunciados anteriormente.

Generación de aplicaciones para la web (I)

## Características principales

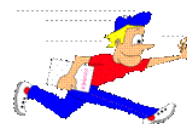
Con Joomla CMS sólo debes ocuparte de la información que desees publicar, ya que el sistema gestionará todos los demás detalles técnicos y administrativos.

1. **Organización del sitio web:** Joomla está preparado para organizar eficientemente los contenidos de un sitio web en secciones y categorías, lo que facilita la navegabilidad para los usuarios y permite crear una estructura sólida, ordenada y sencilla para los administradores. Desde el panel administrador de Joomla se podrá crear, editar y borrar las secciones y categorías del sitio web de la manera en que más convenga.
2. **Publicación de Contenidos:** Con Joomla CMS podrán crearse páginas ilimitadas y editarlas desde un sencillo editor que permite formatear los textos con los estilos e imágenes deseados. Los contenidos son totalmente editables y modificables.
3. **Escalabilidad e implementación de nuevas funcionalidades:** Joomla ofrece la posibilidad de instalar, desinstalar y administrar componentes y módulos, que agregarán servicios de valor a los visitantes del sitio web, por ejemplo: galerías de imágenes, foros, newsletters, clasificados, etc.
4. **Administración de usuarios:** Joomla permite almacenar datos de usuarios registrados y también la posibilidad de enviar e-mails masivos a todos los usuarios. La administración de usuarios es jerárquica, y los distintos grupos de usuarios poseen diferentes niveles de facultades/permisos dentro de la gestión y administración del sitio.
5. **Diseño y aspecto estético del sitio:** Es posible cambiar todo el aspecto del sitio web tan sólo con un par de clicks, gracias al sistema de templates (plantillas) que utiliza Joomla.
6. **Navegación y menú:** Totalmente editables desde el panel administrador de Joomla.
7. **Administrador de Imágenes:** Joomla posee una utilidad para subir imágenes al servidor y usarlas en todo el sitio.
8. **Disposición de módulos modificables:** En un sitio creado con Joomla, la posición de módulos puede acomodarse como se prefiera.
9. **Encuestas:** Joomla posee un sistema de votaciones y encuestas dinámicas con resultados en barras porcentuales.
10. **Feed de Noticias:** Joomla trae incorporado un sistema de sindicación de noticias por RSS/XMS de generación automática.
11. **Publicidad:** es posible hacer publicidad en el sitio usando el Administrador de Banners
12. **Estadísticas de visitas:** con información de navegador, OS, y detalles de los documentos (páginas) más vistos.



## Características de publicación de páginas web en Joomla:

1. **Automatización en la publicación:** Las páginas y documentos de Joomla pueden programarse con fecha de publicación y fecha de caducidad. Es decir un documento puede programarse para que se publique automáticamente al llegar una determinada fecha, y luego despublicarse también de forma automática en otra fecha.
2. **Archivo e historial:** Las páginas viejas o publicaciones que hayan perdido vigencia pueden enviarse a un "archivo" de almacenamiento, sin necesidad de tener que borrarlas. Esto permite también dar la posibilidad a los navegantes de consultar artículos viejos o documentos anteriores en un historial.
3. **Formatos de lectura:** Cada documento es generado automáticamente por Joomla en formato **PDF**, en **versión imprimible**, y en **XML**.
4. **Envío por E-mail:** Los usuarios del sitio Joomla podrán enviar automáticamente a un amigo por email cada documento publicado.
5. **Valoración de contenidos:** Los visitantes del sitio podrán votar la calidad de lo publicado.
6. **Comentarios:** (opcional) Los usuarios podrán comentar sus opiniones o expresar sus inquietudes en la misma página de contenidos.



Generación de aplicaciones para la web (I)

## Extensiones

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



Algunos ejemplos de estas extensiones son:



1. Generadores de Formularios Dinámicos,
2. Directorios de Empresas u Organizaciones,
3. Gestores de Documentos,
4. Galerías de Imágenes Multimedia,
5. Motores de Comercio y Venta Electrónica,
6. Software de Foros y Chats,
7. Calendarios,
8. Software para Blogs,
9. Servicios de Directorio,
10. Boletines de Noticias,
11. Herramientas de Registro de Datos,
12. Sistemas de Publicación de Anuncios,
13. Servicios de Suscripción,

...y muchos, muchos más. Puedes ampliar conocimientos en el siguiente epígrafe ...

**Para saber más**

***Sí Joomla! te ha llamado la atención, visitas los siguientes enlaces donde encontrarás todo lo que deseas sobre este servidor de contenidos:***

[Página oficial del proyecto Joomla.org](#)

[Página de amigos de Joomla en España](#)

### Autoevaluación

De las siguientes afirmaciones, ¿cuál es cierta?

- a) Una de las grandes "armas" de Joomla! es la gran comunidad que tiene detrás, que no para de sacar extensiones dándole más utilidad
- b) Joomla! está escrito en PHP y SQL
- c) Joomla! es un sistema de gestión de contenidos
- d) Todas las anteriores son correctas

Comprobar

Generación de aplicaciones para la web (I)

**CASO.** *Enrique y Andrés comentan cómo añadir a la Web las conclusiones del último congreso al que asistieron. Enrique ha realizado un informe, pero cree que sería muy útil añadir el video de la última mesa redonda. Andrés le comenta que pueden colgar el vídeo para que se descargue. Antonio que se encuentra con ellos, les comenta la posibilidad de usar flash, una tecnología para la creación de animaciones de extendido uso, con la que podrían mostrar del vídeo sin necesidad de que los usuarios deban descargárselo y bajarse reproductores... Además es una tecnología muy usada y soportada por la mayoría de los navegadores.*



Flash es la tecnología desarrollada por Macromedia (ahora Adobe) más comúnmente utilizada en la Web que permite la creación de animaciones vectoriales. El interés en el uso de gráficos vectoriales es que éstos permiten llevar a cabo animaciones de poco peso, es decir, que tardan poco tiempo en ser cargadas por el navegador.

**Los gráficos vectoriales**, son aquellos en los cuales una imagen es representada a partir de líneas (o vectores) que poseen determinadas propiedades (color, grosor...). La calidad de este tipo de gráficos no depende del zoom o del tipo de resolución con el cual se esté mirando el gráfico. Por mucho que nos acerquemos, el gráfico no se pixeliza, es decir no pierde calidad y no se nos generan esos cuadraditos al ampliarlo, ya que el ordenador traza automáticamente las líneas para ese nivel de acercamiento.



Así, Flash se sirve de las posibilidades que ofrece el trabajar con gráficos vectoriales, fácilmente redimensionables y alterables por medio de funciones, así como de un almacenamiento inteligente de las imágenes y sonidos empleados en sus animaciones por medio de bibliotecas, para optimizar el tamaño de los archivos que contienen las animaciones.

Esta optimización del espacio que ocupan las animaciones, combinada con la posibilidad de cargar la animación al mismo tiempo que ésta se muestra en el navegador (técnica denominada **streaming**), permite aportar elementos visuales que dan vida a una web sin que para ello el tiempo de carga de la página se prolongue hasta límites insoportables por el visitante.

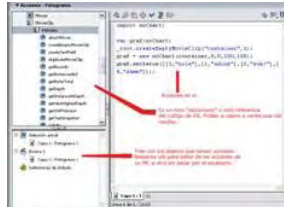
Además de este aspecto meramente estético, Flash introduce en su entorno la posibilidad de interaccionar con el usuario. Para ello, Flash invoca un lenguaje de programación llamado ActionScript. Orientado a objetos, este lenguaje tiene claras influencias del Javascript y permite, entre otras muchas cosas, gestionar el relleno de formularios, ejecutar distintas partes de una animación en función de eventos producidos por el usuario, saltar a otras páginas, etc.

De este modo, Adobe pone a nuestra disposición una tecnología pensada para aportar vistosidad a nuestra web al mismo tiempo que nos permite interaccionar con nuestro visitante. Por supuesto, no se trata de la única alternativa de diseño vectorial aplicada al Web pero, sin duda, se trata de la más popular y más completa de ellas.

Aquí podemos ver algunas imágenes del programa.



Esta imagen es un póster. Pulsa sobre la imagen para verla a tamaño real y poder leer el texto



Esta imagen es un póster. Pulsa sobre la imagen para verla a tamaño real y poder leer el texto



Esta imagen es un póster. Pulsa sobre la imagen para verla a tamaño real y poder leer el texto

Una vez creado un objeto con Flash es preciso incrustarlo en un documento HTML para que pueda ser visto en Internet mediante el navegador. En la siguiente presentación se muestra el código que se incluye al hacer esta inclusión. Hay que decir que se consigue de forma automática con la opción **Publicar** de Flash.



#### Como añadir una animación swf a nuestra página web

##### Para saber más

**Sobre flash podemos encontrar infinidad de información en la web, aquí te destacamos algunos de los enlaces más interesantes.**

[Página oficial de Flash en Adobe](#)

[Manual de Flash por webestilo.com](#)

[Página con gran cantidad de trucos en consejos para Flash](#)

Generación de aplicaciones para la web (I)

#### ActionScript



ActionScript es un lenguaje orientado a objetos, utilizado en especial en aplicaciones web animadas realizadas en el entorno Adobe Flash, la tecnología de Adobe para añadir dinamismo al panorama web. Fue lanzado con la versión 4 de Flash, y desde entonces hasta ahora, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamismo y versatilidad muy altos en la versión 9 (Adobe Flash CS3) de Flash.

ActionScript es un lenguaje de script, esto es, no requiere la creación de un programa completo para que la aplicación alcance los objetivos. El lenguaje está basado en especificaciones de estándar de industria ECMA-262, un estándar para Javascript, de ahí su parecido con Javascript.

La versión más extendida actualmente es ActionScript 3.0, que incluye clases y es utilizada en la última versión de Adobe Flash.



Generación de aplicaciones para la web (I)

#### ¿Qué podemos hacer con ActionScript?

Son muchas las cosas que se pueden hacer con ActionScript, y muchas seguramente las que se pueden conseguir combinándolo con otras herramientas. A continuación enumeramos las más importantes.

##### ■ Controlar la línea de tiempo

Una película de Flash está formada por una serie de fotogramas secuenciales llamada línea de tiempo.

Utilizando ActionScript en ciertos fotogramas clave podemos crear elementos interactivos como:

1. La botonera clásica de navegación, con botones que reaccionan a clics del ratón y envían la cabecera de la película a un



fotograma concreto.

2. Contenido que se anima basándose en movimientos del ratón.
3. Objetos que pueden ser movidos por el ratón o el teclado.
4. Campos de texto que permiten a los usuarios entrar datos a la película como en un formulario.
5. Controlar el contenido audiovisual.

#### ■ Controlar las propiedades de los objetos

ActionScript puede ser utilizado para examinar o modificar las propiedades de los elementos de una película.

Por ejemplo, podemos:

1. Cambiar el color y la localización de un objeto.
2. Reducir el volumen de un sonido.
3. Especificar la tipografía de un bloque de texto.
4. Campos de texto que permiten a los usuarios entrar datos a la película como en un formulario.
5. Modificar las propiedades repetidamente produciendo comportamientos únicos como son los movimientos basados en la física y la detección de colisiones.



#### ■ Generación de contenido programado

Con ActionScript podemos generar contenido directamente desde la biblioteca de la película o duplicar contenido existente en el Escenario. El contenido generado en forma de programa puede servir como:

1. Un elemento estático.
2. Una plantilla visual aleatoria.
3. Un elemento interactivo, por ejemplo una nave en un juego espacial.
4. Una opción de un menú que se abre cuando la presionan.(pull-down menú).

#### ■ Comunicación con el servidor

ActionScript provee de una amplia variedad de herramientas para enviar y recibir información del servidor. Ejemplos de comunicación con el servidor son:

1. Enlace a una página web.
2. Libro de visitas.
3. Aplicación de chat.
4. Juego multijugadores a través de la red.
5. Transacción de e-comercio.
6. Sitio personalizado con nombre de usuario y contraseña.



---

Generación de aplicaciones para la web (I)

## SWF

SWF es un formato de archivo de gráficos vectoriales creado por la empresa Macromedia (ahora absorbida por Adobe). Los archivos SWF pueden ser creados por el programa Flash, aunque hay otras aplicaciones que también lo permiten. Básicamente es un formato vectorial, pero también admite bitmaps y, necesita para ser ejecutado el plugin Flash, el cual permite mostrar las animaciones vectoriales que contienen los ficheros.



Los archivos SWF suelen ser suficientemente pequeños para ser publicados en la Web en forma de animaciones o applets con diversas funciones y grados de interactividad. También son usados frecuentemente para crear animaciones y gráficos en otros medios, como menús para películas en DVD y anuncios de televisión.



El objetivo principal del formato SWF es crear archivos pequeños pero que permitan la interactividad y que funcionen en cualquier plataforma, aún sobre un ancho de banda reducido, cómo un navegador web. El plugin que permite reproducir ficheros SWF está disponible para diferentes navegadores y diferentes sistemas operativos.

Finalmente a continuación mostramos un ejemplo con muchas de las posibilidades de Flash. Podemos destacar el menú lateral desplegable al que podemos añadir opciones sin más que editar el archivo "menudesp.xml" que lo acompaña y siguiendo el formato establecido incorporar nuevas opciones. También hay que destacar las animaciones interactivas como el pinball y la presentación de tipos de ordenadores.



### Portafolio

#### Para saber más

**Para ver una magnífica demostración de las posibilidades que presenta Flash, no sólo en el terreno de la programación y diseño Web, te recomendamos dos enlaces interesantes con noticias de actualidad mediante presentaciones en Flash. Son continuamente actualizadas, por lo que deberías visitarlas de forma asidua si quieres estar a la última.**

[Gráficos Multimedia El País](#)

[Gráficos Multimedia El Mundo](#)

## Autoevaluación

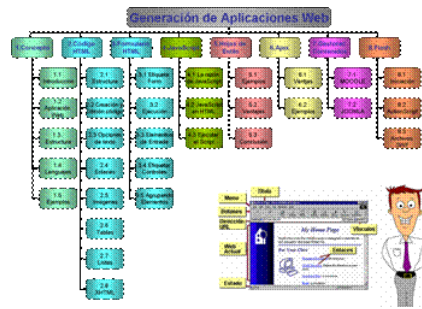
De las siguientes afirmaciones, ¿cuál no es cierta?

- a) Gracias a ActionScript podemos controlar las propiedades de los objetos
- b) La tecnología Flash es una tecnología propietaria, pero de amplio uso en el mercado
- c) Las imágenes que creamos con esta tecnología son de buena calidad, pero al hacer zoom se pixelizan, es decir se ven los cuadraditos
- d) La tecnología Flash permite la creación de animaciones vectoriales

Comprobar

---

Generación de aplicaciones para la web (I)



Generación de aplicaciones para la web (I)