

Unidad Didáctica 8
**Audio, vídeo y
otros objetos multimedia**

Contenido

1. Introducción
2. Los objetos multimedia
3. Multimedia en HTML 5
4. Marcos flotantes o *iframes*
5. Elementos interactivos en HTML 5

1. Introducción

Además de las imágenes, existen multitud de objetos multimedia que pueden ser incrustados en los documentos Web, todo ello debido al incremento de las capacidades multimedia de los ordenadores en los últimos años.

En este tema se verán algunos de estos elementos multimedia así como otros conceptos y aplicaciones relacionados con los mismos (*plugin*, tipos MIME etc.). Se hará hincapié en la gran cantidad de novedades que aporta, en este ámbito, la aparición de HTML 5.



2. Los objetos multimedia

2.1. Plugins

Los navegadores no son capaces de manejar todos los formatos multimedia que existen de forma nativa. Para solventar esto, se utilizan programas auxiliares llamados *plugins*, los cuales actúan “integrándose” en el navegador.

Básicamente un *plugin* consiste en un bloque de código que interpreta el contenido de un fichero determinado, mostrando los resultados en el navegador. Algunos ejemplos de *plugins* pueden ser ActiveX de Microsoft o Flash de Adobe, etc.

Definición



La tecnología conocida como *streaming* goza de gran protagonismo en la actualidad Web. Su principal característica es permitir a los usuarios recibir sonido y video a través de Internet sin necesidad de esperar la descarga completa del recurso, por lo que la reproducción comienza automáticamente (youtube).

2.2. El tipo MIME

Para que un *plugin* funcione, el navegador tiene que disponer de un mecanismo que identifique el formato del fichero a ejecutar. Existe un estándar denominado norma MIME (*Multipurpose Internet Mail Extensión*), creado inicialmente para ayudar a la transferencia de ficheros por correo electrónico.

Cuando un navegador recibe un fichero a ejecutar, también recibe una información que es el tipo MIME del propio fichero. Esto facilita en gran medida el tratamiento del archivo. En el caso de que el navegador no reconozca un tipo MIME concreto, por lo general se pregunta al usuario que indique el programa a utilizar para abrirlo.

Algunos ejemplos de tipo MIME son: *image/gif*, *video/quicktime*, *text/html* etc.

En definitiva, gracias al tipo MIME, el navegador sabe qué *plugin* utilizar cuando detecte un fichero determinado.

2.3. Directivas de inserción de objetos

Existen varias formas de incorporar objetos multimedia a un documento HTML, aunque la más usual es a través de directivas específicas.

HTML dispone de etiquetas determinadas que son capaces de definir con gran precisión la integración de elementos multimedia dentro de un documento Web. Estas directivas son fundamentalmente dos: `<EMBED>` y `<OBJECT>`.

2.3.1 `<EMBED>`

La directiva `<EMBED>` se localiza siempre en el cuerpo del documento y permite especificar la localización de los ficheros multimedia y los parámetros de visualización y ejecución. Esta directiva es un poco antigua, pero sigue conservándose en HTML 5. Algunos atributos han sido eliminados (aunque todavía sean soportados por algunos navegadores), pero otros se conservan. Estos son:

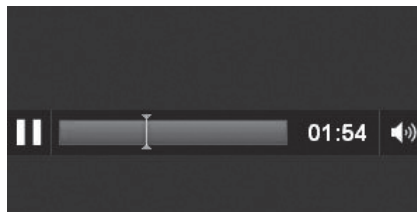
- *src*: Lugar donde se localiza el fichero a reproducir con el *plugin*.
- *type*: Tipo MIME del fichero. Si no se especifica, el navegador tomará la extensión del fichero (indicado *src*) para elegir el *plugin* adecuado.
- *name*: Nombre del objeto contenido en el *plugin*.
- *height*, *width*: Son los valores de altura y anchura, en píxeles, de la ventana de visualización del objeto.

Esta directiva también permite incluir atributos específicos para cada *plugin*. Esto se utiliza para sacar el máximo partido de estas aplicaciones sin necesidad de complicaciones en el código.

A continuación se muestra un ejemplo de reproducción de un archivo de audio del tipo *mp3*:

```
...
<embed src="sonidos/rise.mp3" type="audio/mp3"></embed>
...
```

El aspecto en el navegador (Chrome) corresponde a un barra de reproducción típica de cualquier software de reproducción de audio:



2.3.2 <OBJECT>

<OBJECT> es un contenedor para archivos multimedia. El uso de <OBJECT> ya está generalizado y es la opción recomendada en HTML 5, ya que originalmente se creó para ofrecer una solución universal en la inclusión de objetos genéricos.. Entre la etiqueta <OBJECT> y su cierre se pueden incluir

directivas `<PARAM>` que definen parámetros adicionales para la reproducción del contenido multimedia. Los atributos más importantes de `<OBJECT>` son:

- *data*: Indica dónde se encuentra el archivo del objeto multimedia a insertar. Este mecanismo funciona en todos los navegadores, aunque algunos *plugins* necesitan que esta información se indique con `<PARAM>`
- *form*: asocia el objeto con un formulario.
- *name*: nombre del objeto, para identificarlo.
- *type*: Indica el tipo MIME del objeto declarado en *data*.
- *usemap*: Se usa en el caso de que una imagen se comporte como un mapa interactivo, este atributo se usa para definirlos y contiene el identificador de un elemento denominado `<MAP>`. El valor de *usemap* debe ser igual al valor del atributo *name* del elemento `<MAP>` asociado. La gran ventaja de este método con relación al ya explicado con `` (ver Capítulo 9) es que, al colocar la definición de `<MAP>` (que es la lista de enlaces) entre `<OBJECT>` y su cierre, cabe la posibilidad de que (por alguna razón) al no poder cargar la imagen, se vean los enlaces del mapa. A continuación se muestra un ejemplo de uso de este atributo:

...

```
<OBJECT data="map.png" usemap="#mapa" type="/image/png" width=500,
height=500, TITLE="Paisaje">
  <MAP name="mapa">
    <A href="mar.html" coords= "200,59,350,140">Mar</A>
    <A href="arena.html" coords= "400, 200,450,450">Arena</A>
  </MAP>
</OBJECT>
```

...

- *height*, *width*: Representan, respectivamente, la altura y anchura del objeto en píxeles (aunque también se puede expresar en porcentaje).

Una vez vistos estos atributos, se va a mostrar un ejemplo de reproducción de un objeto `.swf` (shockwave flash) en el navegador. En el caso de que no se pueda reproducir el elemento, se mostrará una imagen `.gif`.

```

...
<OBJECT type="application/x-shockwave-flash" data="fm_01010503.swf"
width="200" height="150" border="2">
<OBJECT data="imagenes/img2.jpeg">
</OBJECT>
</OBJECT>
...

```

En el caso de que se pueda reproducir, se verá un vídeo flash (.swf).

El siguiente código contiene un video de Youtube, en el caso no poder reproducirlo, se mostrará un mensaje de error:

```

...
<OBJECT width="425" height="350">
<PARAM name="movie" value="http://www.youtube.com/
watch?v=MyEHpXjd41E">
</PARAM>
<PARAM name="wmode" value="transparent">
</PARAM>
Lo siento pero su navegador no permite visualizar este objeto.
</OBJECT>
...

```

3. Multimedia en HTML 5

HTML 5 se preocupa en gran medida por la semántica de la páginas Web. Esto es fácil deducirlo a la vista de las dos nuevas etiquetas contenedoras: <VIDEO> y <AUDIO>.

3.1. <VIDEO>

Se encarga de dar soporte a contenido audiovisual para que se pueda reproducir directamente en el navegador. Los principales atributos de esta etiqueta son:

- *src*: Contiene la ubicación del archivo a reproducir.
- *autoplay*: Si se incluye este atributo, el video se reproducirá automáticamente una vez cargada la página.
- *loop*: Este atributo sirve para que el video se vuelva a reproducir una vez finalizado.
- *height, width*: Tamaño en pantalla (la altura y anchura respectivamente en píxels o bien en porcentaje) del vídeo.
- *controls*: Indica si hay controles embebidos del vídeo.
- *poster*: Se utiliza cuando se desee colocar una imagen en lugar del video en el caso de que este último no se reproduzca.

No todos los navegadores soportan los mismos formatos de video. Por esta razón es muy útil especificar varios formatos por cada video a reproducir y así aumentar las posibilidades de que un navegador interprete correctamente uno de estos formatos. Cada uno de estos formatos es declarado con la directiva `<SOURCE>` (dentro de `<VIDEO>`), la cual posee algunos atributos fundamentales:

- *src*: sirve para indicar la dirección de cada fichero.
- *type*: Es el tipo MIME del fichero.
- *codecs*: Contiene información adicional sobre la forma de interpretar el video.



Definición

Cuando el organismo responsable (W3C) de crear la especificación HTML5 hizo el borrador, especificó que el formato de los vídeos debería ir en Ogg / Theora (extensión .ogv y .ogg), un codec de vídeo libre sin patentes. Algunas empresas que componen el W3C se quejaron fuertemente ya que tenían intereses comerciales en usar sus propios codecs, y al final no se especificó un codec concreto para usar con la etiqueta `<video>`. No obstante, Ogg/Theora es el formato que mejor soportan los navegadores actuales (que implementen HTML 5).

En el siguiente ejemplo, el navegador intentará reproducir un video en *mp4* en caso de no poderlo hacer con el *ogg* (no tener instalados los codecs etc.).

```
...
<VIDEO width="320n height="240" autoplay>
  <SOURCE src="Video.ogg" type="video/ogg"/>
  <SOURCE src="Video.mp4" type="video/mp4"/>
</VIDEO>
...
```



3.2. <AUDIO>

Con esta etiqueta se podrán reproducir archivos de audio a través de los navegadores. Los atributos más importantes de esta directiva son *src*, *autoplay*, *loop* y *controls*; los cuales tienen la misma función que en la directiva <VIDEO>. A continuación se muestra un ejemplo:

```
...
<AUDIO autoplay controls>
  <SOURCE src="Rise.wav"/>
```

```
<SOURCE src="Rise.mp3"/>  
</AUDIO>
```

...

3.3. <CANVAS>

Con el uso de la directiva <CANVAS> se pueden definir áreas de presentación independientes del contenido de la página. Cada *canvas* o “lienzo” definido es un área bidimensional (aunque se está trabajando para diseñar un soporte 3D) en el que se pueden dibujar figuras, reproducir imágenes, etc.

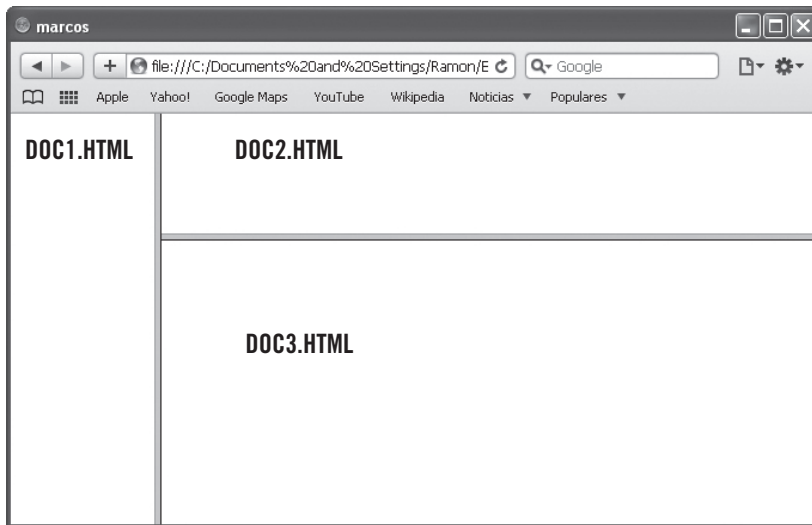
Con HTML puede definirse varios *canvas* dentro de un documento con parámetros que permitan identificarlos (*name*) y asignarles unas dimensiones (*width* y *height*), pero la forma de mostrarlos requiere programación (por ejemplo en Javascript), lo cual va más allá del conocimiento de HTML y CSS.

4. Marcos flotantes o *iframes*

Desde su aparición, los marcos o *frames*, han sido unos componentes polémicos y muy discutidos. Permitían convertir un documento en un conjunto de contenedores que podían albergar varios documentos HTML. Estos elementos fueron muy usados los comienzos del desarrollo Web.

No obstante, siempre han existido muchos detractores que nunca han sido partidarios de usar este tipo de elementos, lo cual ha dado lugar a que desaparezcan con HTML 5. Por el contrario, ha “sobrevivido” una única clase de este tipo de elementos: los llamados *frames* flotantes o *iframes*, hasta ahora una directiva exclusiva de los navegadores de Microsoft.

Un *iframe* permite definir un área “flotante” dentro de una página Web en la que puede cargarse un documento. Al contrario de un *canvas* que puede manipularse para ocultarse o mostrarse en primer plano y que requiere programación; un *iframe* se muestra siempre y su contenido se puede definir y gestionar únicamente con HTML (sin necesidad de conocer otro lenguaje).



La directiva encargada de la definición *de frames* flotantes es `<IFRAME>`. Estos son sus atributos:

- *name*: es el nombre que recibirá el *iframe*.
- *src*: es la URL del documento que se cargará en el *iframe* (puede ser un documento local u otra dirección Web).
- *height*: la altura del *iframe* en píxeles.
- *width*: anchura del *iframe* en pixels.
- *sandbox*: permite especificar restricciones de seguridad en el acceso a los contenidos de la página que contiene el *iframe*.
- *scrolling*: Indica si se desea incluir barra de scroll en el *iframe*. Puede tener dos valores: "Yes" y "No".

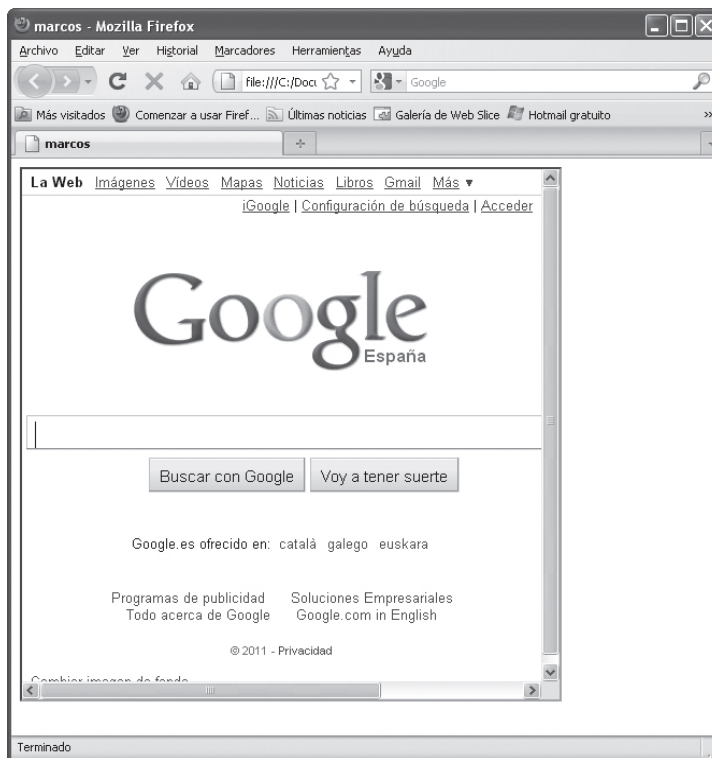


Aplicación práctica

Imagine que se desea cargar una página (www.google.es) dentro de un marco. Para esto decide recurrir al uso de un `iframe` (llamado contenedor). Escriba el código que defina a este marco flotante si se desea que sus dimensiones sean 500 x 500 píxeles.

SOLUCIÓN

```
...  
<IFRAME src="http://www.google.es" scrolling="yes" name="contenedor" height="500" width="500"></IFRAME>  
...
```



5. Elementos interactivos en HTML 5

Uno de los grandes cambios de HTML 5 con respecto a versiones anteriores es la enorme simplificación respecto a la forma de introducir contenidos multimedia que antes requerían extensiones concretas como Flash. Esto explica la aparición de algunas directivas nuevas que mejoran el dinamismo de las páginas Web. A continuación se explican brevemente (debido a la escasa o nula compatibilidad que presentan con los navegadores actuales) algunas de ellas.

<DETAILS>: Este elemento recoge información adicional que puede ser mostrada u ocultada. Esta ampliación/reducción de detalles se puede efectuar a partir de la selección de un símbolo que el navegador es capaz de incorporar.

<MENU> y <COMMAND>: <MENU> define un menú con sus respectivas opciones, donde cada elemento se define con la directiva con <COMMAND> que contiene atributos como: *OnClick*: acción que debe desencadenarse al pulsar sobre el elemento del menú, y *label* e *icon* con la etiqueta y la imagen asociados, respectivamente, a ese elemento.

<MENU> cuenta con el atributo *label*, el cual da nombre al menú, y *type* que permite indicar el tipo de menú de comandos. Existen tres valores para este atributo “*context menu*” para un menú contextual; “*toolbar*” para una barra de herramientas; y “*list*” para una lista de comandos.