

Introducción

HTML5, la última versión del *Hypertext Markup Language* (HTML), es la versión oficial más actual del lenguaje. Presenta muchos recursos nuevos en una variedad de áreas. Algunas de las novedades más destacadas son:

Etiquetas integradas multimedia para audio y vídeo

Un bastidor para dibujar contenido en el navegador

Formatos más inteligentes que permiten hacer cosas como validación mediante el uso de un atributo requerido

Aprender a programar en HTML5 resulta extremadamente sencillo para quienes ya tienen conocimientos previos en programación HTML, aunque tampoco implica demasiada dificultad para quienes se inician desde cero. Todo lo que hay que hacer es tener interés, disponer de un curso o tutorial y practicar todo lo que se aprende.

Elemento audio de HTML5

Algunas etiquetas nuevas sirven para integrar contenido multimedia, pues es habitual que estos nuevos tipos de información estén presentes en la web.

audio: Para insertar sonido dentro de una web.

video: Para insertar clips de vídeo.

embed: Para embeber contenido externo de otro tipo, como el traído de diversos *plugins* que se comercializan actualmente o se comercializarán en el futuro.

source: Permite especificar varias fuentes diferentes cuando se insertan elementos en AUDIO y VÍDEO.

track: Permite especificar varias pistas de sonido o vídeo para los elementos AUDIO y VÍDEO.

Estas nuevas etiquetas audio forman parte de la especificación de HTML5 y permiten embeber (o empotrar) archivos de audio en desarrollos de sitios en internet, sin necesidad de utilizar *plugins* adicionales como el tan incordiante Flash.

El nuevo elemento permite el uso de diferentes formatos de archivo puesto que los que soportan los diversos navegadores no son parte del estándar sino que dependen de la implementación de cada fabricante. Esta tabla sirve para comprobar que formatos soportan los navegadores más usados de forma nativa:

Códec	Tipo	IE >=9	Firefox	Chrome	Safari	Opera
Ogg Vorbis	Libre (BSD)	no	si	si	no	si
WAV PCM	Privativo (Microsoft, IBM)	no	si	<i>si</i>	si	si
MP3	Privativo (En disputa)	si	no	si	si	si en Linux y FreeBSD
AAC	Bajo patente AAC	si	no	<i>si</i>	si	si en Linux y FreeBSD
Speex	Libre (BSD)	no	no	<i>si</i>	no	no

Tanto en el navegador Safari como en Chrome, algunos formatos dependen de Quicktime y *ffmpeg*, respectivamente. Chrome realmente no soporta todos los formatos de forma nativa, tan solo Vorbis y [MP3](#), por el contrario, si se compila Chromium (versión libre de Chrome) con soporte para *ffmpeg*, si estarán disponibles todos esos formatos de forma nativa a través de *ffmpeg*, de ahí la cursiva, del *si*, en la tabla.

¿Código de inserción?

Incorporar un archivo de audio en HTML5 es muy simple:

```
<audio src="archivo.mp3">
</audio>
```

Como sucede con las etiquetas de HTML5 lo que se encuentre entre las etiquetas audio solo será tenido en cuenta por navegadores que soporten la nueva etiqueta. Obviamente, el ejemplo anterior es extremadamente simple, la etiqueta audio ofrece más posibilidades. Por ejemplo, se puede lograr que una páginas sea detestada por la mayoría de personas que acceden a ella si se embebe un archivo de audio que se autorreproduzca por sí mismo, mediante el parámetro del ejemplo mostrado seguidamente:

```
<audio src="archivo.mp3" autoplay>
</audio>
```

El atributo `autoplay` es un atributo booleano y por lo tanto no necesita de un valor, sin embargo se ha de escribir código XHTML, se puede usar `autoplay="true"`. Si la gente que visita el sitio web aún no ha decidido salir corriendo por el *autoplay* siempre se puede hacer "más divertido" utilizando además el atributo `loop` para que se reproduzca en bucle:

```
<audio src="archivo.mp3" autoplay loop>
</audio>
```

Controles

Si no se aspira a desacreditar más el sitio web en cuestión, con los bucles de audio autorreproducidos sin posibilidad de pausa, siempre se puede ofrecer al usuario que navega por el sitio web unos controles básicos para el elemento audio:

```
<audio src="archivo.mp3" controls>
</audio>
```

El atributo `controls` hace que el navegador proporcione una interfaz con controles para la reproducción y el control del volumen de forma nativa. Los controles del navegador pueden ser suplantados por controles propios usando JavaScript a través de una API que proporcione métodos para controlar todos los aspectos de la reproducción del archivo de audio:

```
<audio id="player" src="archivo.mp3">
</audio>
<div>
<button onclick="document.getElementById('player').play();">Reproducir</button>
<button onclick="document.getElementById('player').pause();">Pausa</button>
<button onclick="document.getElementById('player').volume += 0.1;">Subir Volumen</button>
<button onclick="document.getElementById('player').volume -= 0.1;">Bajar Volumen</button>
</div>
```

Buffer

Si se requiere que el archivo de audio sea precargado en segundo plano por el navegador, se puede usar el atributo `preload` que puede tomar tres posibles valores: `none`, `auto` y `metadata`. El navegador Safari precarga los archivos de audio por defecto, usando `preload="none"` se garantiza que eso no ocurra en aquellos contextos en los que no es necesario que el navegador precargue los archivos, por ejemplo, cuando existen muchos archivos en una misma página:

```
<audio src="archivo.mp3" preload="none">
</audio>
```

Si solo se dispone de un archivo de audio en la página, es adecuado usar `preload="auto"`.

Especificar más de un formato de archivo a la vez

Aunque el formato MP3 está ampliamente extendido desde hace muchos años, no es un formato abierto y está ligado a patentes. Las tecnologías que decodifican archivos MP3 deben pagar una tasa.

Eso es posible para grandes empresas como Apple, Microsoft o Google, pero no lo es para los grupos de código abierto y software libre o las empresas pequeñas. Safari, IE y Chrome puede reproducir archivos MP3 sin problemas, sin embargo ese no es el caso de Firefox y de Opera (aunque Opera puede a través de `ffmpeg`).

Existen otros formatos abiertos como Vorbis que no requieren de patentes para ser utilizados. Firefox, Opera y Chrome lo soportan de forma nativa, IE y Safari no. Además, existe una forma de definir más de un archivo de audio en diferentes formatos utilizando únicamente una etiqueta `audio` para ello. En lugar de usar el atributo `src` en la etiqueta de apertura, se pone la etiqueta `source` para poder definir múltiples archivos:

```
<audio controls>
<source src="archivo.ogg" type="audio/ogg" />
<source src="archivo.mp3" type="audio/mpeg" />
```

</audio>

Los navegadores que puedan reproducir archivos Ogg Vorbis no buscarán más allá de la primera etiqueta `source` mientras que los que no lo soporten, avanzarán hasta la siguiente etiqueta. De esta manera se pueden considerar las necesidades de todos los usuarios sin discriminar a ningún navegador. Aunque el atributo `type` no es obligado, siempre es conveniente utilizarlo y ayudar a los navegadores. El elemento `source` es un elemento *standalone* así que si se está usando sintaxis XHTML se puede utilizar la etiqueta `source` / tal y como se comprueba en el ejemplo.

¿Y para quienes usen un navegador que no cumpla con los estándares?

Dejando a un lado la obviedad de que usarlos no tiene ningún sentido, existe una forma de ofrecer esos archivos de audio utilizando Flash. Como bien he apuntado antes, cualquier elemento que se encuentre entre la apertura y cierre de las etiquetas audio será tenido en cuenta solo por navegadores que no soporten aún la nueva especificación por lo que se puede jugar con esto último para seguir utilizando Flash para estos menesteres, e incluso se puede ofrecer un enlace directo para aquellos navegadores que no soporten ni la nueva especificación, ni flash utilizando la etiqueta `object`:

```
<pre class="prettyprint lang-html"><code><audio controls>
<source src="archivo.ogg" type="audio/ogg" />
<source src="archivo.mp3" type="audio/mpeg" />
<object type="application/x-shockwave-flash" data="player.swf?soundFile=archivo.mp3">
<param name="movie" value="player.swf?soundFile=archivo.mp3" />
<a href="archivo.mp3">Descarga el archivo de audio</a>
</object>
</audio>
```

Elemento vídeo de HTML5

Las especificaciones del estándar HTML5 incorporan la nueva etiqueta `video` con la que se pueden embeber archivos de vídeo de forma nativa en los desarrollos web sin necesidad de *plugins* adicionales.

Con el aumento del ancho de banda, los contenidos de vídeo han ido aumentando de forma vertiginosa hasta convertirse en una de las mayores necesidades de capacidad en internet. Hasta ahora la forma por excelencia de reproducir vídeo a través de un navegador ha sido utilizar el *plugin* de Flash. Mucho ha tenido que ver la industria de la pornografía primero y YouTube después en esa decisión. Usar el plugin de flash no es la única vía, se puede usar también Silverlight ya que no tiene complejidad alguna utilizar dicha plataforma para generar contenido multimedia. Por supuesto, todo está cambiando con la llegada de HTML5.

Como siempre y para no perder la costumbre, existe una guerra entre fabricantes de navegadores sobre que formato de vídeo debe de ser el estándar para la nueva especificación y por supuesto no todos reproducen los mismos formatos de forma nativa.

Códec	Tipo	IE >=9	Firefox	Chrome	Safari	Opera
Ogg Theora	Libre	no ^[1]	si	si	no ^[2]	si
H.264	Propietario	si	no ^[3]	no	si	no
VP8	Libre	no ^[4]	si	si	no	si

Fuente: [Genbeta:dev](#)

¿Cómo funciona?

El elemento `video` es muy parecido al elemento `audio`, también dispone de los atributos `autoplay`, `loop` y `preload`. También se puede especificar la fuente de un archivo bien mediante el atributo `src` en la etiqueta de apertura o usando el elemento `source` entre las etiquetas de apertura y cierre. Asimismo se pueden utilizar los controles que ofrece el navegador de forma nativa a través del atributo `controls` o bien puedes ofrecer tus propios controles en JavaScript.

Obviamente, el elemento `video` ocupa espacio en la ventana, por lo tanto será conveniente definir un tamaño para el mismo:

```
<video src="archivo.mp4" controls width="360" height="240">
</video>
```

Portada del vídeo

También podemos definir una imagen representativa para el vídeo para que sea mostrada al navegador como portada del elemento antes de la reproducción usando el atributo `poster`:

```
<video src="archivo.mp4" controls width="360" height="240" poster="poster.jpg">
</video>
```

Multiformato

Como siempre y para no marginar a ningún navegador, se pueden especificar diferentes archivos en diferentes formatos para que todos ellos puedan reproducir el contenido:

```
<video controls width="360" height="240" poster="poster.jpg">
<source src="archivo.ogv" type="video/ogg" />
<source src="archivo.mp4" type="video/mp4" />
</video>
```

Como ocurre con el elemento `audio` no es necesario especificar el atributo `type` pero es siempre recomendable.

¿Y que pasa con quienes usan navegadores que no son compatibles con los estándares?

No hay problema, se puede usar el mismo truco que se empleó con el elemento `audio` para seguir dando soporte a lo que pronto serán navegadores obsoletos y añadir el *plugin* de Flash a través de la etiqueta

object:

```
<video controls width="360" height="240" poster="poster.jpg">
<source src="archivo.ogv" type="video/ogg" />
<source src="archivo.mp4" type="video/mp4" />
<object type="application/x-shockwave-flash" width="360" height="240"
data="player.swf?file=archivo.mp4">
<param name="movie" value="player.swf?file=archivo.mp4" />
</object>
</video>
```

Sobre Adobe Flash

[Adobe Flash Player](#) es una aplicación en forma de reproductor multimedia creado inicialmente por Macromedia y actualmente distribuido por Adobe Systems. Permite reproducir archivos con formato SWF que pueden ser creados con la herramienta de autoría Adobe Flash, 2 con Adobe Flex o con otras herramientas de Adobe y de terceros. Estos archivos se reproducen en un entorno determinado. En un sistema operativo tiene el formato de aplicación del sistema, mientras que si el entorno es un navegador, su formato es el de un *Plugin* u objeto *ActiveX*.

Originalmente creado para mostrar animaciones vectoriales en dos dimensiones, pasó a convertirse, hasta hace poco tiempo, en la opción preferida a la hora de crear aplicaciones Web que incluyen flujo de audio y vídeo e interactividad. La utilización de gráficos vectoriales le permite disminuir el ancho de banda necesario para la transmisión y, por ende, el tiempo de carga de la aplicación.

Sin embargo una de las cosas que traen más problemas en un ordenador es Flash, muchas veces falla y hay que cerrar el navegador para que vuelva a funcionar, pues si se desea evitar este problema existe [HTML5](#), una extensión para el navegador Chrome que fuerza la carga de los videos en HTML5.

La extensión es ideal para quienes tienen un ordenador antiguo y desean una mejor experiencia en Youtube, las ventajas de reproducir un vídeo en HTML5 en vez de Flash son muchas y la principal es que consume muchos menos recursos por lo que funciona mejor. La extensión es muy sencilla de usar, pues ni siquiera tiene opciones que cambiar, simplemente se instala y se accede a sitios como Youtube, Amazon o Vimeo y automáticamente cargará los vídeos en HTML5, también funciona con videos incrustados en otras páginas haciéndolo así más completa.

Algunas razones para no utilizar Flash:

Los buscadores dan menos importancia al contenido en Flash.

Cada web implica una interfaz distinta.

Crea dependencia de un formato no estándar.

Mayor dependencia de los creadores de la web.

Tendencia a centrarse en los efectos gráficos espectaculares.

Pérdida de control por parte del usuario.