

HTML 5 - MULTIMEDIA

2º DAW

Curso 2022/2023

Mari Cruz Gómez García

IES La Encantá (Rojales)

INTRODUCCIÓN

Hoy en día es de lo más común descargar y ver vídeo dentro del navegador web.

Conforme el ancho de banda comenzó a hacerse realmente ancho, la popularización del video a través de una conexión de datos se hizo posible; la información dejó de ser sólo texto e imágenes pequeñas o ultra comprimidas y la web se tornó realmente multimedia.

Y hablando de multimedia, una tecnología ascendió velozmente al trono de la contención de vídeo incrustado en un sitio web: Flash de Adobe.

Hasta hace pocos años esto no era un gran problema, ya que la mayoría del contenido de vídeo de la web se encontraba en FLV o F4V1

Sin embargo, varios factores hicieron que el panorama cambiara:

- La llegada de dispositivos móviles, ligeros, de bajo consumo de energía y gran poder gráfico.
- La necesidad de contar con un lenguaje web mejor y más expresivo.
- Conflictos de interés de grandes jugadores de la industria tecnológica.

INTRODUCCIÓN

En HTML5 se pretende que el contenido multimedia, (audios y vídeos) formen parte de la página y dejen de ser unos elementos incrustados.

Se pretende por tanto que los nuevos navegadores se adapten a ello e incorporen recursos para poder reproducir este contenido sin tener que usar plugins externos.

De momento pocos son los navegadores que permiten reproducir audios o vídeos en los formatos más habituales sin utilizar plugins externos.

INTRODUCCIÓN

Flash tenía problemas de rendimiento, así como alta demanda de recursos computacionales.

Este fue el pretexto ideal para que Apple decidiera no dar soporte a la tecnología de Adobe en el sistema operativo (iOS) de sus nuevos dispositivos móviles, comenzando en 2007 con el iPhone y en 2010 con el iPad.

Ahora bien, si Apple no iba a incluir a Flash en sus dispositivos móviles, tenía que contar con algo que lo reemplazara. Este sustituto sería HTML5.

Una de las características más mencionadas de HTML5 fue la capacidad de procesar vídeo.

VIDEO DIGITAL

Un video digital está compuesto de diversas pistas o tracks, de las cuales una es de vídeo y una o más de audio. Las pistas de audio y vídeo se encuentran encapsuladas en contenedores que definen el formato del vídeo y son codificadas mediante algoritmos llamados codecs.

La palabra códec proviene de la contracción de las palabras coder y decoder (codificador y decodificador), aunque también se denomina códec al software o hardware encargado de realizar la codificación y decodificación de la pista.

Los archivos de video de tipo AVI, MOV, MP4, M4V u otro similar son contenedores de video.

Existen diversos reproductores de video que dan soporte a uno o más de estos formatos (por ejemplo, Windows Media Player da soporte, principalmente, a AVI y QuickTime de Apple es capaz de reproducir videos en MOV, MP4 y otros), sin embargo eso no es suficiente para poder reproducir un vídeo, y para esto, es necesario que la aplicación cuente también con el códec apropiado, es decir, con el mismo con el cual fue codificado.

5

CONTENEDORES

Como se mencionó antes, un contenedor define el formato y contiene las pistas de vídeo y audio. Los contenedores soportados por HTML5 son los siguientes:

MP4 (MPEG-4)

Especificado por el MPEG (Moving Picture Experts Group), este estándar se encuentra presente en diversos medios: desde sistemas de TV por cable, reproductores de discos Blu-ray, smartphones, películas y series de TV de compra y alquiler en la tienda iTunes de Apple, etc.

HTML5 soporta este contenedor en varios de los navegadores principales, generalmente, con una extensión .mp4 o .m4v.

6

CONTENEDORES

OGG

Este estándar abierto está soportado por HTML5 en varios navegadores bajo la extensión .ogg. OGG soporta vídeo codificado con Theora y audio con Vorbis.

WebM

La especificación de este contenedor, apoyada por Google, es abierta y utiliza también Vorbis para codificar sus pistas de audio y el códec VP8 para video. HTML5 soporta este contenedor utilizando la extensión .webm

7

CÓDECS DE VIDEO PARA HTML5

Los siguientes codecs de vídeo son válidos en HTML5, aunque no todos ellos son soportados por cada uno de los principales navegadores.

H.264 (MPEG-4 Part 10 o MPEG-4 AVC)

El estándar de codificación de video H.264 fue desarrollado por el MPEG (Moving Picture Experts Group) y es uno de los formatos más utilizados actualmente en diversas aplicaciones, desde video en sitios web, consolas de videojuegos, sistemas de TV por cable, etc.

La utilización de este códec requiere licencia del MPEG LA Group.

8

CÓDECS DE VIDEO PARA HTML5

Theora

Basado en VP3, este estándar fue cedido en 2002 a Xiph.org por parte de On2 con licencia de uso libre (royalty-free). Theora está soportado por HTML5 en Mozilla Firefox, Google Chrome y Opera en un contenedor OGG.

VP8

Al igual que VP3 (después Theora), VP8 fue desarrollado por On2 Technologies, empresa adquirida por Google en 2010. VP8 es un códec de estándar abierto utilizado para codificar vídeo, normalmente, contenido en el formato WebM.

CÓDECS DE AUDIO PARA HTML5

Los siguientes codecs de audio son válidos en HTML5, aunque, como ocurre con los codecs de vídeo, no todos ellos son soportados por todos los navegadores.

AAC (Advanced Audio Coding)

AAC es un estándar de codificación de audio digital utilizado en una gran variedad de dispositivos: reproductores de música, teléfonos, consolas de videojuegos, etc.

Puede ser utilizado como pista de audio en un vídeo en conjunto con el códec H.264 en un contenedor MP4.

AAC requiere licencia para su uso.

Vorbis

Vorbis es un códec de estándar libre que se utiliza comúnmente para codificar sonido dentro de vídeos de formato OGG y WebM, aunque es más habitual encontrarlo en el primero.

CÓDECS DE AUDIO PARA HTML5

MP3 (MPEG-1 Audio Layer 3)

El estándar de codificación de audio MP3 es uno de los más populares dentro del mundo de la música digital. Este códec puede ser utilizado también en conjunto con el H.264 en un contenedor MP4 para reproducir vídeo mediante HTML5 en ciertos navegadores.

El estándar requiere licencia de uso.

SOPORTE NAVEGADORES

| Contenedor | Extensión | | Licencia de Contenedor | Codec | Tipo | Licencia de Codec | Soporte Nativo | |
|------------|-----------|--------------|------------------------|--------|-------|-------------------|---------------------------------|---------------------------------|
| | HTML5 | Otros | | | | | HTML5 | Otros |
| MP4 | .mp4 | .mov .m4v | Requiere licencia | AAC | Audio | Requiere licencia | Chrome (3.0+) Safari (3.1+) | Reproductores de discos Blu-ray |
| | | | | H.264 | Video | Requiere licencia | IE (9.0) Android | iTunes QuickTime Player |
| | | | | MP3 | AUDIO | Requiere licencia | iOS | TV por cable |
| | | | | H.264 | Video | Requiere licencia | Safari (3.1+) IE (9.0) | Apple TV PS3 Xbox 360 |
| OGG | .ogv | -- | Gratuita | Vorbis | Audio | Gratuita | Firefox (3.5+) Opera (10.5+) | VLC XMMS (Linux) |
| | | | | Theora | Video | Gratuita | Chrome (3.0+) | |
| WebM | .webm | -- | Gratuita | Vorbis | Audio | Gratuita | Firefox (4.0+) Opera (10.6+) | VLC |
| | | | | VP8 | Video | Gratuita | Chrome (3.0+) Android (2.3+) | |

SOPORTE NAVEGADORES

Es importante hacer notar que no existe una sola combinación formato/códec de vídeo/códec de audio, a la cual, todos los navegadores modernos le den soporte, por este motivo, es necesario realizar la codificación en varios formatos.

Existen numerosas herramientas que nos permiten recodificar un archivo de vídeo, dos ejemplos de aplicaciones gratuitas serían:

HandBrake (<http://handbrake.fr/>): aplicación de escritorio.

Firefogg (<http://www.firefogg.org/>): complemento de Firefox

ELEMENTO VIDEO

El elemento <video> ofrece varios atributos para establecer su comportamiento y configuración:

Los atributos width y height, al igual que en otros elementos HTML ya conocidos, declaran las dimensiones para el elemento o ventana del reproductor.

El tamaño del vídeo será automáticamente ajustado para entrar dentro de estos valores, pero no fueron considerados para redimensionar el vídeo, sino para limitar el área ocupada por el mismo y mantener la consistencia en el diseño.

ELEMENTO VIDEO

El atributo `src` indica la fuente del vídeo. Este atributo puede ser reemplazado por el elemento `<source>` y su propio atributo `src` para declarar varias fuentes con diferentes formatos, como en el siguiente ejemplo:

```
<video id="medio" width="720" height="400" controls>
```

```
<source src="video.mp4">
```

```
<source src="video.ogg">
```

```
</video>
```

En el ejemplo anterior, el elemento `<video>` fue expandido. Ahora, dentro de las etiquetas del elemento hay dos elementos `<source>`. Estos nuevos elementos proveen diferentes fuentes de vídeo para que los navegadores puedan elegir. El navegador leerá la etiqueta `<source>` y decidirá qué archivo reproducir de acuerdo a los formatos que soporte.

15

ATRIBUTOS PARA VIDEO

Podemos ver en los ejemplos anteriores que hemos añadido un atributo `controls`, este atributo muestra controles de vídeo provistos por el navegador. Cuando el atributo está presente cada navegador activará su propia interface de control del vídeo, permitiendo al usuario comenzar a reproducir el vídeo, pausarlo o saltar hacia un cuadro específico, entre otras funciones.

Junto con `controls`, también podemos usar los siguientes atributos:

- `autoplay`: cuando este atributo está presente, el navegador comenzará a reproducir el vídeo automáticamente, tan pronto como pueda.
- `loop`: si este atributo es especificado, el navegador comenzará a reproducir el vídeo nuevamente cuando llegue al final.

16

ATRIBUTOS PARA VIDEO

- poster: este atributo es utilizado para proveer una imagen que será mostrada mientras esperamos que el vídeo comience a ser reproducido.
- preload: este atributo puede recibir tres valores distintos:
- none: indica que el vídeo no debería ser cacheado, por lo general, con el propósito de minimizar tráfico innecesario.
- metadata: recomendará al navegador que trate de capturar información acerca de la fuente (por ejemplo, dimensiones, duración, primer cuadro, etc.).
- auto: es el valor configurado por defecto, que le sugerirá al navegador descargar el archivo tan pronto como sea posible.

17

ATRIBUTOS PARA VIDEO

```
<video id="reproductor" width="720" height="400" preload controls loop
poster="poster.jpg">
<source src="video.mp4">
<source src="video.ogg">
</video>
```

18

EL ELEMENTO AUDIO

HTML5 provee un nuevo elemento para reproducir audio en un documento HTML.

El elemento, por supuesto, es `<audio>` y comparte casi las mismas características del elemento `<video>`.

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

ATRIBUTOS AUDIO

- **src**: Este atributo especifica la URL del archivo a ser reproducido. Al igual que en el elemento `<video>`, normalmente, será reemplazado por el elemento `<source>` para ofrecer diferentes formatos de audio entre los que el navegador pueda elegir.
- **controls**: Este atributo activa la interface que cada navegador provee por defecto para controlar la reproducción del audio.
- **autoplay**: Cuando este atributo está presente, el audio comenzará a reproducirse automáticamente tan pronto como sea posible.
- **loop**: Si se indica este atributo, el navegador reproducirá el audio una y otra vez de forma automática.

ATRIBUTOS AUDIO

- preload: Este atributo puede tomar tres valores diferentes:
 - *none*: indica que el audio no debería ser cacheado, normalmente, con el propósito de minimizar tráfico innecesario.
 - *metadata*: recomendará al navegador obtener información sobre el medio (por ejemplo, la duración).
 - *auto*: es el valor configurado por defecto y le aconseja al navegador descargar el archivo tan pronto como sea posible.

21

CODIFICADORES AUDIO

Una vez más, debemos hablar acerca de codificadores.

MP3 está bajo licencia comercial, por lo que, no es soportado por navegadores como Firefox u Opera.

Vorbis (el codificador de audio del contenedor OGG) es soportado por estos navegadores, pero no por Safari e Internet Explorer. Por esta razón, nuevamente, debemos aprovechar el elemento `<source>` para proveer al menos dos formatos, entre los cuales el navegador pueda elegir.

22

EJEMPLO

```
<audio id="medio" controls>
  <source src="cancion.mp3">
  <source src="cancion.ogg">
</audio>
```

El código anterior reproducirá música en todos los navegadores utilizando los controles por defecto. Aquellos que no puedan reproducir MP3 reproducirán OGG y viceversa. Debemos recordar que MP3, al igual que MP4 para vídeo, tienen uso restringido por licencias comerciales, por lo que sólo podemos usarlos en circunstancias especiales, de acuerdo con lo determinado por cada licencia.

•

23

TIPOS MIME

Cuando definimos fuentes en nuestro código, podemos ayudar a nuestro navegador en la identificación del contenido especificando el tipo MIME utilizado.

```
<audio>
  <source src="elvis.mp3" type='audio/mpeg; codecs="mp3"'>
  <source src="elvis.ogg" type='audio/ogg; codecs="vorbis"'>
</audio>

<video poster="star.png" autoplay loop controls tabindex="0">
  <source src="movie.webm" type='video/webm; codecs="vp8, vorbis"' ,
  <source src="movie.ogv" type='video/ogg; codecs="theora, vorbis"' />
</video>
```

| Valor MIME | Audio/Video |
|------------|---------------|
| video/ogg | Video |
| video/mp4 | Video |
| video/webm | Video |
| audio/ogg | Audio (*.ogg) |
| audio/mpeg | Audio (*.mp3) |

Aquí definimos el elemento y las fuentes. El navegador elegirá solamente una de ellas. Al especificar el atributo type (no es obligatorio), permitimos que el navegador conozca el tipo MIME y los tipos de codecs que debe utilizar antes de descargar la canción. Si no indicamos dicho atributo, el navegador intentará averiguar, mediante prueba y error, cuál es el tipo adecuado.

24

RESUMEN FORMATOS AUDIO-VIDEO

| Navegador | MP3 | Wav | Ogg |
|-------------------|-----|-----|-----|
| Internet Explorer | SI | NO | NO |
| Chrome | SI | SI | SI |
| Firefox | SI | SI | SI |
| Safari | SI | SI | NO |
| Opera | NO | SI | SI |

| Navegador | MP4 | WebM | Ogg |
|-------------------|-----|------|-----|
| Internet Explorer | SI | NO | NO |
| Chrome | SI | SI | SI |
| Firefox | SI | SI | SI |
| Safari | SI | NO | NO |
| Opera | NO | SI | SI |