

SERVICIO PÚBLICO
DE EMPLEO ESTATAL
SEPE

1. Introducción.....	3
2. El problema de la incompatibilidad de formatos.	3
3. Vídeos en las páginas web.	4
4. Audio en las páginas web.	7
5. Elemento Canvas.	8
7. Inserción directa de imágenes SVG.	9
7. Otra forma de incorporar elementos.	9
8. Mapas de google.	10
9. Bibliografía.	10

1. Introducción.

Indudablemente no se puede concebir una página web hoy en día sin incluir en ella elementos visuales atractivos. En este sentido se conoce como multimedia el hecho de incorporar medios procedentes de diversos formatos.

Esencialmente la base es la inclusión de imágenes, pero también es cada vez más común la inclusión de vídeos, sonido. Todo aquello que no sea texto, está metido respecto a este capítulo, en el mismo carro.

El hecho de que podamos incluir una imagen en la página requiere que tengamos en cuenta cómo funcionan los navegadores.

Como hemos explicado en los primeros capítulos, una página web es un documento de texto que utiliza el lenguaje etiquetado HTML, el navegador interpreta las etiquetas HTML y así da formato de forma apropiada a los distintos elementos de la página. Evidentemente cualquier contenido que no sea texto queda fuera de este planteamiento.

2. El problema de la incompatibilidad de formatos.

Dejando de lado los contenidos especiales (como Flash, Silverlight u otros formatos complejos), cualquier aplicación web actual muestra imágenes y, cada vez más, vídeo y audio.

El problema es el formato de estos elementos, ya vimos en el tema anterior la cantidad de formatos diferentes que poseen las imágenes. En el caso del vídeo y el audio el problema es mayor.

El vídeo es un tipo de contenido que ocupa mucho espacio por lo que la lucha ha sido siempre comprimir al máximo su tamaño. Para ello se utiliza software especial capaz de decodificar y codificar vídeo haciendo que ocupe muy poco espacio. La potencia de los actuales procesadores permite realizar esa ardua tarea incluso a tiempo real (mientras el vídeo se reproduce).

El problema es que el navegador tiene que tener incorporado ese software, llamado popularmente **codec**. No solo eso: tiene que ser el mismo codec.

En el vídeo (también en el audio), los archivos se graban eligiendo un tipo concreto de vídeo, como por ejemplo puede ser **mp4**. Sin embargo el hecho es que un archivo de vídeo es un contenedor de elementos multimedia que incluye: las imágenes de vídeo, el audio, metadatos y otras informaciones. Lo normal es que se usen al menos dos codecs: uno para el audio y otro para el vídeo. Incluso el mismo formato puede usar diferentes codecs para cada cosa.

En definitiva las posibilidades son enormes porque hay decenas de codecs (aunque algunos más populares que otros). Por lo tanto poder reproducir un elemento concreto dependerá de si el navegador incorpora los codecs concretos.

De esta forma si incorporamos vídeo a nuestra página a través de un enlace normal:

```
<a href="video1.mpeg">Ver vídeo</a>
```

El navegador buscará si disponemos del plugin apropiado para ver el vídeo. Si no es así, simplemente lo descargará en nuestro ordenador al no poder mostrar el contenido directamente.

En otros tipos de elementos multimedia pasará parecido. Como ya se ha comentado los navegadores deben ser capaces de traducir el tipo de contenido concreto, bien porque de base incorporan esta posibilidad o porque se añaden plugins para ello.

3. Vídeos en las páginas web.

FORMATOS de VÍDEO

Como ya hemos comentado, el vídeo posee numerosos formatos relacionados con los diversos codecs y tipos de contenedores del mismo. Como cada navegador ha apostado por unos codecs concretos de vídeo, lo normal es que los creadores de contenido graben los vídeos usando varios codecs y formatos.

En la siguiente dirección disponemos de una lista de compatibilidad en audio y vídeo:

https://developer.mozilla.org/en-US/docs/Web/HTML/Supported_media_formats#Browser_compatibility

En HTML 5 se ha simplificado, en parte, el problema ya que se habla solo de tres formatos de vídeo:

- **MP4.** Probablemente el más popular ya que es muy compatible con los navegadores de los dispositivos móviles. Se basa en el formato **MPEG-4 parte 14** de vídeo (creado por el grupo de trabajo especificación del MPEG (*Moving Pictures Experts Group*) pensada para la reproducción de vídeo en línea.

Normalmente usa el codec **H264** para el vídeo, pero también se puede usar **FLAC** y otros codecs. Para el audio se usan, normalmente, los codecs **MP3** o el **AAC**. Este contenido tiene asociado el tipo MIME (indicación oficial de contenido) **video/mp4**.

- **WebM.** Popularizado por Google, pero ya muy compatible con casi todos los navegadores se ha creado para optimizar la visualización de vídeo a través de los navegadores. Suele usar como códec **VP8**, pero ya se empieza a usar **VP9** (supera al anterior). Para el audio lo normal es que use el códec **Vorbis**. Su tipo MIME es **video/webm**.
- **OGG Vídeo.** Formato creado por la fundación **Xiph.org** bajo la premisa de ser un formato abierto y sin patentes. Suele usar el codec de vídeo **Theora** y para el audio **Vorbis** u **Opus**. La extensión de sus archivos suele ser **ogv**, pero su tipo MIME es **video/ogg**.

INSERTAR VÍDEO


```
<video src="ejemplo.mp4"></video>
```

```
<video src="carrera.mp4" controls autoplay  
poster="f1.jpg">  
</video>
```

src	URL al vídeo que se desea mostrar
width	Anchura del vídeo en nuestra página
height	Altura del vídeo en nuestra página
autoplay	Usa el valor fijo <i>autoplay</i> (o no se indica valor alguno) para indicar que el vídeo se inicia automáticamente en cuanto se descargue
loop	Usa el valor fijo <i>loop</i> (o no se indica valor alguno) para indicar que el vídeo se ejecuta automáticamente una y otra vez
controls	Con valor <i>controls</i> (o no indicando valor alguno) indica que el navegador mostrará controles para que el usuario pueda controlar la reproducción del vídeo de reproducción (pausa, play,...)
preload	Indica al navegador una recomendación sobre cómo debemos realizar la descarga. Posibilidades: <ul style="list-style-type: none"> ▪ auto. El vídeo se descarga en cuanto se carga la página ▪ none. El vídeo no se descarga. Cuando el usuario pulse play, se descargará. ▪ metadata. Descarga los metadatos del vídeo, no el vídeo en sí
poster	Permite indicar la dirección URL a una imagen que se mostrará mientras el vídeo no se está reproduciendo. Si no se usa este atributo, se usa el primer fotograma del vídeo como póster.
muted	Atributo sin valor que marca que el vídeo se debe reproducir sin audio.

Crea un documento HTML en el que insertes un vídeo. Debes indicar que no se descargue hasta que el usuario no pulse play y que añada una imagen hasta que el vídeo se reproduzca.

USO DE DIFERENTES FORMATOS DE VÍDEO

Dentro del elemento **video**, podemos incorporar elementos **source** que hagan referencia a distintas fuentes de vídeo. **source** dispone de estos atributos:

atributo	uso
src	URL a la fuente de vídeo
type	Tipo de vídeo que queremos mostrar. Se usa un tipo MIME para especificar el tipo

```
<video autoplay="autoplay" controls="controls"
poster="foto1.jpg" >
  <source src="video.mp4"
  type="video/mp4;codecs='avc1.42E01E, mp40a.40.2'" >
  <source src="video.ogv"
  type="video/ogg;codecs='theora, vorbis' " >
No se puede mostrar el vídeo en este navegador
</video>
```

Dentro de **type**, el uso de codecs es opcional, ya que si el navegador no reconoce el formato no suele hacer caso a los **codecs** que se indiquen (aunque a veces les descarga). La idea es que si el primer formato no se reconoce (primer elemento **source**), se intenta el segundo y así sucesivamente. Si ninguno es reproducible por el navegador actual, éste mostrará la frase final tras el último **source**.

SUBTÍTULOS

Se realiza mediante elementos de tipo **track**, los cuales se colocan dentro de la etiqueta **video**. Desde hace unos años, casi todos los navegadores manejan los subtítulos, aunque todavía hay algunas carencias al respecto.

Los subtítulos son textos que permiten traducir el contenido del vídeo. Por ejemplo un vídeo podría estar grabado en inglés y los subtítulos nos enseñarías el texto traducido en español a tiempo real. Hay varios formatos de subtítulos pero el estándar en la web es **WebVTT** (*Web Video Text Tracks*). Normalmente se utilizan archivos de texto cuya extensión es vtt basados en este estándar.

Ejemplo de archivo vtt:

```
WEBVTT
1
00:00:00.150 --> 00:00:01.300
¿Habéis visto a Sara?
2
00:00:01.540 --> 00:00:03.800
-No
-Yo tampoco la he visto
```

En este archivo la primera frase se muestra en el primer segundo del vídeo. En los dos segundos siguientes se mostrará el otro texto. En el navegador Chrome, y en algunos otros, se permite dar formato CSS al texto de los subtítulos.

El elemento track que permite asignar subtítulos a un vídeo posee los siguientes atributos:

atributo	uso
src	URL al archivo VTT o SRT que contiene los subtítulos
kind	<p>Tipo de subtítulos. Posibilidades:</p> <ul style="list-style-type: none"> ■ subtitles. Subtítulos normales ■ captions. Igual que el anterior pero con posibilidad de añadir sonidos, música y otros elementos más allá de la traducción del diálogo. El caso más utilizado este tipo de "tracks" son los audios especiales para personas con problemas de visión o textos para personas con problemas de audición. ■ descriptions. Texto alternativo e independiente de la película. También muy utilizado para ayudar a interpretar el audio del vídeo a las personas sordas. ■ chapters. Contiene los capítulos que permiten navegar de forma ágil por la película ■ metadata. Metadatos de la película. No visibles para los usuarios, se usan con JavaScript.
label	Nombre de los subtítulos para reconocerlos en caso de haber indicado varios. Es el texto con el que aparece el nombre del idioma en el navegador al elegir un subtítulo u otro.
srclang	Código de idioma (oficial según la norma ISO BCP 47) de los subtítulos: por ejemplo es para español, en para inglés, fr francés, etc.
default	Indica los subtítulos que se deben de cargar por defecto si no es posible saber el idioma de preferencia del usuario.

```

<video autoplay="autoplay" controls="controls" poster="foto1.jpg" >
  <source src="video.mp4" type="video/mp4" >
  <source src="video.ogv" type="video/ogg" >
  <track src="spa.vtt" kind="subtitles" srclang="es"
    label="Español" >
  <track src="deu.vtt" kind="subtitles" srclang="de"
    label="Alemán" >
  El navegador no puede mostrar el vídeo
</video>

```

INSERCIÓN DE VÍDEO YOUTUBE

0_video.html

4. Audio en las páginas web.

Insertar audio en una páginas web es muy similar a insertar vídeo. El elemento que lo hace posible, se llama precisamente audio. Sus atributos (que funcionan igual que con el elemento **video**) son:

atributo	valor
<code>src</code>	URL a la fuente de audio que estamos insertando
<code>autoplay</code>	El audio se reproducirá inmediatamente no se esperará a que el usuario pulse el botón Play
<code>loop</code>	El audio se repetirá continuamente
<code>controls</code>	Se muestran los controles de ese audio (<code>play</code> , <code>rewind</code> , <code>pause</code> , etc.)
<code>preload</code>	Recomendación sobre cómo debemos realizar la descarga. Posibilidades: <ul style="list-style-type: none"> ■ <code>auto</code>. El audio se descarga en cuanto se carga la página ■ <code>none</code>. El audio no se descarga. Cuando el usuario pulse play, se descargará. ■ <code>metadata</code>. Descarga los metadatos del audio, no el audio en sí

Con el audio hay el mismo problema con la cuestión de los codecs y los formatos, por lo que también es habitual convertir el audio a distintos formatos y dar opciones dentro de la etiqueta audio gracias a la etiqueta **source**. Ejemplo:

```
<audio controls="controls" autoplay>
  <source src="audio1.ogg" type="audio/ogg">
  <source src="audio2.mp3" type="audio/mpeg">
  No se puede reproducir el archivo de audio
</audio>
```

5. Elemento Canvas.

En inglés **canvas** significa *lienzo*, y define muy bien para que sirve este elemento creado en la norma HTML5. Es uno de los componentes de HTML5 más famosos por el gran aporte que ha supuesto al dinamismo de las páginas web.

Mediante este elemento dispondremos de un área que podremos utilizar donde queramos para dibujar elementos gráficos mediante las instrucciones gráficas del lenguaje JavaScript. Eso ha permitido (gracias a la potencia de JavaScript) crear juegos, animaciones y elementos visuales atractivos en las páginas web.

atributos de canvas

- **id**. Es el atributo que utilizan todos los elementos HTML para identificarlos. En el caso de canvas es casi obligatorio su uso para poder hacer referencia al mismo desde el lenguaje JavaScript.
- **width**. Anchura del lienzo (funciona igual que en el caso de **img**)
- **height**. Altura del lienzo.

1_canvas.html

2_canvas.html

Mediante la etiqueta script podemos colocar código en lenguaje JavaScript, desde ese código podemos utilizar el lienzo para dibujar.

Los canvas pueden ser animados, interactivos, usados en juegos, etc.

https://www.w3schools.com/graphics/canvas_intro.asp

7. Inserción directa de imágenes SVG.

En el tema anterior hemos visto que es posible insertar imágenes SVG (imágenes de tipo vectorial) a través del elemento **img** al igual que las imágenes JPEG o PNG por ejemplo. Sin embargo las imágenes SVG están construidas en un dialecto de XML que cualquier navegador de hoy en día reconoce.

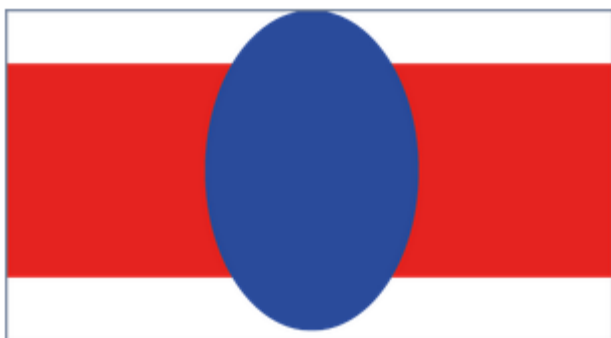
Es decir, podemos insertar el código SVG directamente en una página web. Ejemplo:

```
<svg xmlns="http://www.w3.org/2000/svg">
  ..etiquetas svg
</svg>
```

Dentro del elemento **svg** se colocan las etiquetas que permiten dibujar en el área del svg. En principio si no indicamos tamaño alguno, todo el área de la página web se utiliza para dibujar en SVG, pero podemos indicar atributos **width** y **height** para indicar tamaños concretos.

Las etiquetas que se pueden utilizar dentro de **svg** pertenecen al lenguaje estándar SVG y rebasan el propósito de este manual, pero un ejemplo de ellas sería:

```
<h1>Dibujo SVG</h1>
<svg xmlns="http://www.w3c.org/2000/svg" height="400">
  <rect id="rect1" x="50" y="50" width="300"
    height="100" fill="red" >
  <ellipse id="elips1" cx="200" cy="100" rx="50"
    ry="75" fill="blue" >
</svg>
```



3_svg.html

7. Otra forma de incorporar elementos.

Permite incorporar cualquier tipo de contenido en una página web.

```
<object data="Wildlife.wmv" type="video/x-ms-wmv" width="500" height="300">  
  <param name="autoplay" value="true" />  
</object>
```

8. Mapas de google.

4_maps.html

9. Bibliografía.

- <https://www.w3schools.com/>
- <https://desarrolloweb.com/>
- <http://jorgesanchez.net>
- <https://www.campusmvp.es>
- https://oscarmaestre.github.io/lenguajes_marcas/index.html