

## IMPLANTACIÓN DE CONTENIDO MULTIMEDIA

1. INTRODUCCIÓN .....	2
2. DERECHOS DE LA PROPIEDAD INTELECTUAL. LICENCIAS. LEY DE LA PROPIEDAD INTELECTUAL. DERECHOS DE AUTOR.....	2
2.1. DERECHOS DE LA PROPIEDAD INTELECTUAL .....	2
2.2. LICENCIAS .....	3
2.3. LA PROPIEDAD INTELECTUAL .....	4
2.4. DERECHOS DE AUTOR .....	5
3. LAS IMÁGENES EN LA WEB.....	5
3.1. TIPOS DE IMÁGENES.....	5
3.2. FORMATOS MÁS COMUNES.....	6
3.3. COMPARATIVA DE FORMATOS SEGÚN NAVEGADOR.....	7
3.4. IMÁGENES EN LA WEB: NUEVOS ATRIBUTOS DE IMG .....	8
3.5. LA ETIQUETA PICTURE.....	9
4. INSERTAR AUDIO EN UNA WEB.....	12
4.1. FORMATOS MÁS COMUNES.....	13
4.2. COMPARATIVA Y CONVERSIÓN ENTRE FORMATOS.....	14
4.3. LA ETIQUETA AUDIO.....	15
4.3.1. ALTERNATIVAS OBSOLETAS.....	15
4.3.2. ESTRUCTURA Y PARÁMETROS.....	16
5. INSERTAR VIDEO EN UNA WEB .....	18
5.1. FORMATOS MÁS COMUNES.....	19
5.2. COMPARATIVA Y CONVERSIÓN ENTRE FORMATOS.....	21
5.3. LA ETIQUETA VIDEO .....	22

## 1. INTRODUCCIÓN

En la Web abunda el material interactivo, gráfico y multimedia, pero su fácil acceso y su disponibilidad no significan que todo ese material sea gratuito ni que se pueda utilizar como si fuera propio.

En este tema se introducen formatos y herramientas ligados a la manipulación y conversión de material multimedia y su inclusión en sitios web (imágenes, audio, vídeo y animaciones).

Además, se identificará la legislación relacionada con la salvaguarda y el respeto a los derechos de autor. Dicha legislación regula, a nivel nacional e internacional, los derechos que tiene el material que se ofrece y que ofrecemos en la Web, y no supone una cortapisa a su uso, sino un reconocimiento de sus legítimos propietarios y de sus derechos.

## 2. DERECHOS DE LA PROPIEDAD INTELECTUAL. LICENCIAS. LEY DE LA PROPIEDAD INTELECTUAL. DERECHOS DE AUTOR

Los **derechos de propiedad intelectual y de autor son esenciales** para **proteger la creatividad humana** al ofrecer a los autores incentivos en forma de reconocimiento y recompensas económicas equitativas.

Este sistema de derechos garantiza a los creadores la divulgación de sus obras sin temor a que se realicen copias no autorizadas o actos de piratería.

La *propiedad intelectual* es un concepto amplio, que abarca tanto a los derechos de autor como los llamados derechos conexos a los derechos de autor (que son otras facultades previstas a favor de otros agentes que intervienen en la creación de una obra, como los artistas o intérpretes, los productores de fonogramas y de grabaciones audiovisuales, las entidades de radiodifusión, etc.).

### 2.1. DERECHOS DE LA PROPIEDAD INTELECTUAL

Los derechos que conforman la propiedad intelectual se dividen en varios grupos:

- **Derechos morales:** la legislación española es claramente defensora de los derechos morales, reconocidos para los autores y para los artistas-intérpretes. Estos derechos son *irrenunciables e inalienables*, acompañando al autor toda su vida y a sus herederos al fallecimiento de aquellos. Entre ellos destacan el derecho de reconocimiento de autoría y el derecho de integridad de la obra, entre otros.
- **Derechos de carácter económico:** Hay que distinguir entre derechos relacionados con la explotación de la obra o prestación protegida, que a su vez se subdividen en derechos exclusivos y en los derechos de remuneración.
  - ❑ Los derechos exclusivos son aquellos que permiten a su titular autorizar o prohibir los actos de explotación de su obra o prestación protegida por el usuario, y a exigir de este una retribución a cambio de la autorización que le conceda.
  - ❑ Los derechos de remuneración, a diferencia de los derechos exclusivos, no facultan a su titular a autorizar o prohibir los actos de explotación de su obra o prestación

protegida por el usuario, aunque si obligan a este al pago de una cantidad dineraria por los actos de explotación que realice, cantidad esta que es determinada, bien por la ley o en su defecto por las tarifas generales de las entidades de gestión.

- **Derechos compensatorios:** como el derecho por copia privada que compensa los derechos de propiedad intelectual dejados de percibir por razón de las reproducciones de las obras o prestaciones protegidas para uso exclusivamente privado del copista.

Las preguntas que surgen en función de los derechos de propiedad intelectual y de tus necesidades como creador y autor de sitios web pasarían por ser capaz de responder a preguntas como las siguientes: ¿Dónde encuentro recursos digitales para enriquecer mi trabajo? ¿Puedo utilizarlos libremente? ¿Tengo obligación de citar la fuente u origen de los recursos que utilice? ¿Cómo sé qué tipo de uso puedo hacer de los recursos que he encontrado en la Web? ¿Hay alguna forma de enriquecer con mi trabajo a la comunidad web? ¿Qué derechos tengo sobre los materiales que produzco? Muchas de estas preguntas tienen su respuesta en la licencia con la que se ofrezcan los recursos en la Web.

## 2.2. LICENCIAS


Una *licencia de software* es un contrato entre el autor/titular de los derechos de explotación/distribuidor y el usuario consumidor /usuario profesional o empresa, para utilizar el software cumpliendo una serie de términos y condiciones establecidas dentro de sus cláusulas




- **Copyright:** es la fórmula anglosajona para designar únicamente los derechos de explotación de una obra. El símbolo © asociado a un nombre indica titularidad de derechos de explotación va seguido de "todos los derechos reservados"
- **Copyleft:** consiste en permitir la libre distribución de copias y versiones modificadas de una obra u otro trabajo, exigiendo que los mismos derechos sean preservados en las versiones modificadas. La efectividad de ejercerlo puede depender de la legislación particular de cada país, pero en principio se puede utilizar para programas informáticos, obras de arte, cultura, ciencia, o cualquier tipo de obra o trabajo creativo que sea regido por el derecho de autor.

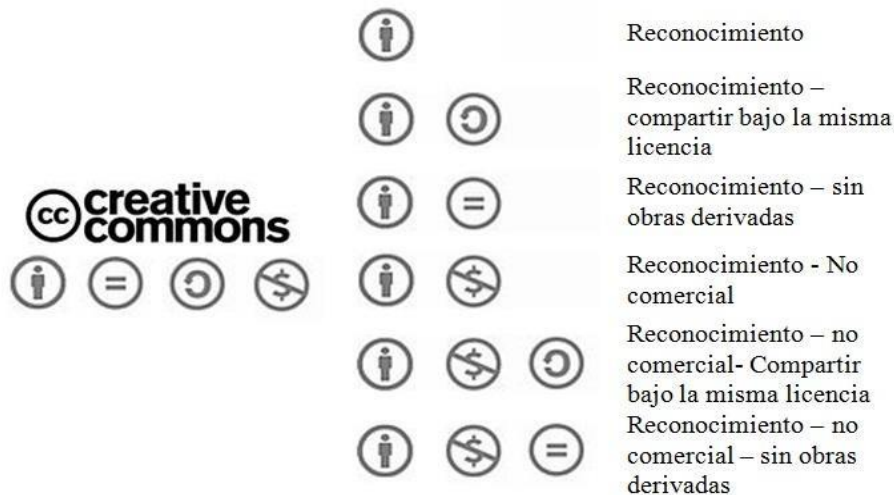


**Creative Commons** plantea un paso intermedio entre el férreo sistema de copyright y el "libre total" (Copyleft). Creative Commons pone a disposición de los autores licencias "a la carta", cuya redacción se incorpora en el sitio web o soporte donde esté la obra y donde se regulan los usos autorizados por el autor con respecto a la referida obra. Esto, evidentemente, también permite que los autores puedan decidir que su obra pase automáticamente a dominio público.

Más concretamente, en las licencias Creative Commons el autor otorga a la comunidad una mayor libertad de uso sobre su obra, aunque bajo determinadas condiciones. Estas condiciones son escogidas por el propio autor, de modo que frente una obra con todos los derechos reservados las licencias Creative Commons proponen "algunos derechos reservados". Tras un proceso de adaptación tenemos en España desde el año 2004 diferentes modelos posibles de licencias Creative Commons a elegir, en función de lo que se pretenda

- **Reconocimiento:**  El creador permite copiar, distribuir y comunicar públicamente la obra mientras se reconozca y cite adecuadamente al autor original.

- **No comercial:**  Se permite copiar, distribuir y comunicar públicamente la obra mientras no sea utilice con fines comerciales.
- **Prohibición de obras derivadas:**  El creador permite copiar, distribuir y comunicar públicamente copias inalteradas de la obra, pero no realizar trabajos derivados de ellas.
- **Redistribución bajo la misma licencia:**  Se permite distribuir obras derivadas solo bajo una licencia idéntica a la que regula la obra original.



## 2.3. LA PROPIEDAD INTELECTUAL

La *propiedad intelectual* es el conjunto de derechos de autor, personales (morales) y patrimoniales (económicos) que corresponden a los autores sobre las obras de su creación.

La norma nacional principal en lo que se refiere a Propiedad Intelectual es el Texto Refundido de la Ley de Propiedad Intelectual, aprobado por Real Decreto Legislativo 1/1996 de 12 de abril.

También de consideración son las Directivas Comunitarias como la Directiva 2001/29/CE sobre derechos de autor y derechos afines en la sociedad de la información.

Otras normas internacionales importantes son: Tratado de la Unión de Berna, firmado en 1886 y revisado en varias ocasiones, y los Tratados de la OMPI de 20 de diciembre de 1996, uno sobre derechos de autor y otro sobre derechos afines.

Las obras que son objeto de propiedad intelectual son todas las creaciones originales literarias, artísticas o científicas expresadas por cualquier medio o soporte, tangible o intangible, actualmente conocido o que se invente en el futuro. Lo que no puede incluirse en la definición anterior son las ideas, la información, y todo conocimiento que es patrimonio común y no es susceptible de apropiación.

## 2.4. DERECHOS DE AUTOR

Se **considera autor** a la persona natural que aparece como tal en la obra. En algunos casos previstos por la ley las personas jurídicas pueden tener algunos derechos económicos de propiedad intelectual.

En las **obras en colaboración**, es decir, en aquellas en las que hay varios autores, los derechos pertenecen a todos los autores, sin embargo, en las obras colectivas **corresponden a la persona bajo cuya iniciativa y coordinación se edita y divulga la obra**.

Los derechos de autor, al igual que se ha comentado para los derechos de propiedad intelectual, son de dos clases:

- *Derechos morales, irrenunciables e inalienables*, como el derecho de reconocimiento de autoría y el derecho de integridad de la obra, entre otros
- *Derechos económicos, transferibles y de duración limitada en el tiempo*, básicamente los derechos de explotación, aunque hay otros (i.e. derecho a remuneración por copia privada)

## 3. LAS IMÁGENES EN LA WEB

La elección y el diseño de las imágenes es una de las decisiones más importantes en el diseño web, deben ser de calidad, deben pesar poco, el conjunto debe ser homogéneo, la información que aportan debe ser accesible, deben ser responsivas, agradables, funcionales, entendibles, deben estar en consonancia con los contenidos de la web, no debemos vulnerar los derechos de autor. En este apartado se estudiarán las características más importantes para lograr estos objetivos.

### 3.1. TIPOS DE IMÁGENES

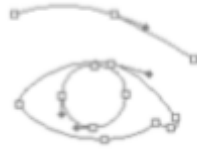
Una de las principales decisiones a la hora de incluir imágenes en una web es elegir el formato correcto para cada tipo de imagen de manera que se logre una correcta relación entre la calidad visual de la misma y su peso en bytes u otras unidades de medida derivadas.

Básicamente, existen distintos tipos de imágenes: las que son mapas de bits, las vectoriales y las imágenes animadas. Estas últimas las trataremos más adelante en la sección dedicada a las animaciones. Esta sección se centra en los otros dos grupos de imágenes (mapas de bits y vectoriales).

- ✓ **Mapas de bits:** Los píxeles son la unidad mínima de las imágenes de mapas de bits, que también son llamadas imágenes raster o bitmaps. Un mapa de bits es una matriz cartesiana (bidimensional) de píxeles, con coordenadas verticales y horizontales que determinan la posición de un píxel en la imagen.



- ✓ **Vectores:** Son la descripción geométrica (matemática) de una imagen. Por ejemplo, para describir todos los puntos del perímetro de un círculo sólo es necesaria su fórmula ( $x^2 + y^2 = R$ ). Modificando la variable R, se obtienen círculos de todos los radios posibles.



Un ejemplo ilustrativo de la diferencia entre imágenes de píxeles y de vectores es el siguiente: Hay dos maneras de enviar una tarta de cumpleaños, o bien ya cocinada en una caja, o enviar la receta, de manera que el destinatario la pueda cocinar siguiendo los pasos contenidos en ella.

Cuando yo envío un archivo de mapa de píxeles estoy enviando la tarta entera. Cuando envío un archivo de vectores, estoy enviando la receta, pues los vectores siempre se pueden "cocinar" para obtener un mapa de píxeles.

Además, un píxel puede requerir mayor o menor cantidad de memoria para ser almacenado, y de acuerdo con este valor (llamado la profundidad de un píxel) la imagen podrá desplegar una mayor o menor cantidad de colores.

Por otro lado, existen diferentes tipos de vectores o, lo que es igual, diferentes métodos matemáticos de describir una imagen. Por ejemplo, una curva es un primitivo importante de la información vectorial.

### 3.2. FORMATOS MÁS COMUNES

#### MAPAS DE BITS

**JPEG (Joint Photographic Experts Group) o JPG:** Fue diseñado para la compresión de imágenes fotográficas, basándose en el hecho de que el ojo humano no es perfecto y no es capaz de captar toda la información que se puede almacenar una imagen de 24 bits. Por regla general, es el más indicado para aquellas imágenes que son fotografías. Las características de este formato son:

- ✓ Número de colores: 24 bits color o 8 bits B/N.
- ✓ Elevado grado de posibilidad de compresión.
- ✓ Formato de compresión con pérdida.
- ✓ No permite transparencias.
- ✓ No permite la animación.



**PNG (Portable Network Graphics)** es el más adecuado para imágenes renderizadas, es decir que provienen de un modelo, ya que se logran unos degradados más suaves y una buena definición de las líneas. Permite compresión sin pérdida. Las características de este formato son:

- ✓ Color indexado hasta 256 colores y TrueColor hasta 48 bits por píxel.
- ✓ Mayor compresión que el formato GIF (+10%).
- ✓ Compresión sin pérdida a diferencia del formato JPG.
- ✓ Canal alfa (permite definir transparencia variable).
- ✓ No permite animación, a diferencia del formato GIF.



**GIF (Graphic Image File Format)** hacen uso de un formato más adecuado para aquellas imágenes sencillas, de formas simples y en las que no existe un elevado número de colores. Sus características son:

- ✓ Número de colores: de 2 a 256 de una paleta de 24 bits.
- ✓ Formato de compresión sin pérdida basado en el algoritmo LZW.
- ✓ Carga progresiva en el navegador.
- ✓ Máscara de transparencia de 1 bit.
- ✓ Permite la animación simple.



**WebP** es un formato gráfico en forma de contenedor, que sustenta tanto compresión con pérdida como sin ella. El formato pretende ser un nuevo estándar abierto para gráficos en color verdadero con compresión con pérdida, y por tanto siendo presentado como competidor directo del esquema JPEG, frente al que se espera superar en la producción de archivos de menor tamaño con una calidad de imagen comparable.



## VECTORES

**SVG (Scalable Vector Graphics)** es un formato vectorial poco conocido pero muy útil para su uso online por su flexibilidad y por la capacidad de ofrecer gráficos con calidad.

El formato SVG es para muchos un total desconocido. Cuando queremos colocar algún gráfico en la web, en la mayoría de las ocasiones optamos por JPG o alguna vez por un PNG o GIF, en caso de necesitar transparencia o animación. Lo que muchos no saben es que se pueden usar archivos vectoriales para su uso en navegadores.

Los archivos SVG se caracterizan por:

- ✓ Ocupar poco, igual que lo haría cualquier archivo vectorial.
- ✓ Soportan estilos CSS, con lo que si cambiamos el estilo en nuestra web también cambiará dentro del archivo.
- ✓ Puede incluir scripts que permiten caminos dinámicos.
- ✓ Puede admitir acciones como los rollovers o cambios al hacer click.



Otros estándares más populares de formatos de imagen vectorial son: **.eps** (archivo postscript), **.ps**, **.pdf**, **.fla** y **.swf** (definidos por Adobe), **.wmf** (definido por Microsoft Windows).

## 3.3. COMPARATIVA DE FORMATOS SEGÚN NAVEGADOR

A continuación se muestra el soporte de cada tipo de imagen en las versiones actuales de los navegadores más empleados, aunque esta lista varía continuamente y se recomienda consultar sitios como “Can I Use” para comprobar el soporte actual. Esto será también aplicable a las comparativas de formatos de audio y vídeo.

Abbreviation	File format	MIME type	File extension(s)	Browser compatibility
<b>APNG</b>	Animated Portable Network Graphics	image/apng	.apng	Chrome, Edge, Firefox, Opera, Safari
<b>BMP</b>	Bitmap file	image/bmp	.bmp	Chrome, Edge, Firefox, Internet Explorer, Opera, Safari
<b>GIF</b>	Graphics Interchange Format	image/gif	.gif	Chrome, Edge, Firefox, Internet Explorer, Opera, Safari
<b>ICO</b>	Microsoft Icon	image/x-icon	.ico, .cur	Chrome, Edge, Firefox, Internet Explorer, Opera, Safari
<b>JPEG</b>	Joint Photographic Expert Group image	image/jpeg	.jpg, .jpeg, .jfif, .pjpeg, .pjp	Chrome, Edge, Firefox, Internet Explorer, Opera, Safari
<b>PNG</b>	Portable Network Graphics	image/png	.png	Chrome, Edge, Firefox, Internet Explorer, Opera, Safari
<b>SVG</b>	Scalable Vector Graphics	image/svg+xml	.svg	Chrome, Edge, Firefox, Internet Explorer, Opera, Safari
<b>TIFF</b>	Tagged Image File Format	image/tiff	.tif, .tiff	None built-in; add-ons required
<b>WebP</b>	Web Picture format	image/webp	.webp	Chrome, Edge, Firefox, Opera

### 3.4. IMÁGENES EN LA WEB: NUEVOS ATRIBUTOS DE IMG

Cabe recordar que las imágenes se incluyen en las páginas web con la etiqueta **<img>**, independientemente de su tipo, y otra serie de elementos cuyo soporte depende de cada navegador, pero que el propio HTML5 no especifica, ya que queda en manos de cada uno de ellos. Los principales atributos ligados a la etiqueta **<img>** son los siguientes:

- ✓ **src:** este atributo es obligatorio al añadir una imagen e indica la dirección URL donde se encuentra el elemento a mostrar.
- ✓ **alt:** este elemento es el "texto alternativo" en una imagen. Es un atributo importante ya que está muy relacionado con el soporte a la característica de accesibilidad en la Web. Dependiendo de una serie de combinaciones hay que ponerlo o no.
- ✓ **width /height:** Si no se especifican estas propiedades, las imágenes ocuparán el mismo tamaño que en el fichero original. No obstante, por norma general es interesante indicarla ya que de esta forma el navegador no tendrá que esperar a finalizar la carga de la página para poder acabar de renderizarla correctamente. Hoy en día estas propiedades se incluyen en el CSS.

La especificación de HTML5 ha incluido dos nuevos atributos para la etiqueta **<img>**:

- ✓ **srcset:** Al igual que src indica la ruta de la imagen a la que se hace referencia. Por ejemplo:

```
srcset="kitten.png"
```

También se puede indicar una lista de rutas separadas por comas y que incluyan el sufijo que indica la densidad de píxeles. Por ejemplo:

```
srcset="kitten.png, kitten@2X.png 2x"
```

Otra opción más eficiente aún si cabe es la unidad **w** que se refiere a la anchura automática que una imagen ocupará en el navegador teniendo en cuenta el atributo



**sizes** y la **densidad de píxeles**. Por cada resolución, se cogerá la primera imagen cuyo ancho resultante sea menor o igual la unidad especificada. Por ejemplo, en resoluciones de 1000px de ancho se selecciona aquella correspondiente con el valor 1280w, pero a partir de 1281 se seleccionará aquella con el valor 2560w.

```
srcset="lighthouse-landscape-640.jpg 640w, lighthouse-landscape-1280.jpg 1280w, lighthouse-landscape-2560.jpg 2560w"
```

- ✓ **sizes**: Acepta cualquier valor que describa la anchura de la imagen en px, em, rem o vw (no en porcentajes, que solo se podría definir por CSS en la propia etiqueta im). Por ejemplo:

```
sizes="100vw"
```

Otra opción es especificar una lista de media queries separadas por comas. Por defecto se utilizaría el último de los valores definidos.

```
sizes="(max-width: 30em) 100vw, (max-width: 50em) 50vw, calc(33vw - 100px)"
```

### 3.5. LA ETIQUETA PICTURE

El nuevo elemento **<picture>** de HTML5 permite describir con todo detalle cómo deben cargarse las imágenes de tu sitio web. Ya no serán necesarios los hacks de CSS o JavaScript para gestionar las imágenes responsivas de los diseños web. Además, los usuarios se aprovecharán de las ventajas de cargar solamente las imágenes optimizadas para el dispositivo que están utilizando, lo que es especialmente útil para usuarios con móviles y conexiones lentas a Internet.

Al margen de los nuevos atributos **srcset** y **sizes** definidos recientemente para los elementos **<img>**, el nuevo elemento **<picture>** permite una mayor flexibilidad al especificar qué imágenes utiliza el sitio. Gracias a este elemento, será posible escribir código HTML limpio y semántico, dejando que el navegador haga todo el trabajo de seleccionar la mejor imagen para cada situación.

[En este enlace](#) se demuestra el funcionamiento de esta etiqueta. El siguiente código HTML y CSS muestra todo lo necesario para crear la anterior demo:

```
<picture>
  <source media="(min-width: 650px)" srcset="images/kitten-md.png">
  <source media="(min-width: 465px)" srcset="images/kitten-sm.png">
  
</picture>
```

#### SINTAXIS BÁSICA

Cuando el navegador soporta el elemento **<picture>**, tu único trabajo consiste en definir todas las imágenes responsivas que tienes disponibles y es el navegador el que se encarga de seleccionar la mejor alternativa.

A continuación, se explican con más detalle las propiedades explicadas en el ejemplo anterior.

## Elemento source

El elemento `<picture>` no define ningún atributo propio, pero puedes conseguir comportamientos muy avanzados cuando utilizas `<picture>` para encerrar a varios elementos `<source>`.

El elemento **<source>**, que se utiliza para cargar elementos multimedia como audios y vídeos, se ha actualizado para que también soporte la carga de imágenes. Para ello, se le han añadido los siguientes atributos:

- ✓ **srcset y sizes:** Son los mismos atributos que en la etiqueta `<img>`. Sirven para seleccionar la ruta y el tamaño de las imágenes, como se explicaba anteriormente. El atributo srcset es obligatorio en este caso y sustituye a `src` en caso de solo querer incluir una imagen. El atributo sizes es opcional
- ✓ **media:** Opcional. Permite indicar cualquier media query que sea válida en el selector `@media` de CSS (ejemplo `media="(max-width: 30em)"`). De esta manera, la etiqueta `source` solo aparecería cuando se cumple la condición del atributo `media`.
- ✓ **type:** Opcional. Acepta como valor cualquier tipo MIME estándar (ejemplo `type="image/gif"` o `type="image/png"`). No suele ser necesario y normalmente el navegador detecta el formato de cada imagen. No obstante, [en este enlace](#) puedes consultar los tipos de imágenes más comunes.

## Ejemplo en función de la densidad de píxeles

Utiliza los descriptores **1x**, **1.5x**, **2x** y **3x** para añadir soporte para pantallas de alta densidad de píxeles, como por ejemplo las de los smartphones. El atributo `srcset` que permite indicar estos descriptores ahora se soporta tanto en el elemento `<img>` como en los elementos `<source>`

```
<picture>
  <source media="(min-width: 650px)"
    srcset="images/kitten-stretching.png,
    images/kitten-stretching@1.5x.png 1.5x,
    images/kitten-stretching@2x.png 2x">
  <source media="(min-width: 465px)"
    srcset="images/kitten-sitting.png,
    images/kitten-sitting@1.5x.png 1.5x,
    images/kitten-sitting@2x.png 2x">

  
</picture>
```

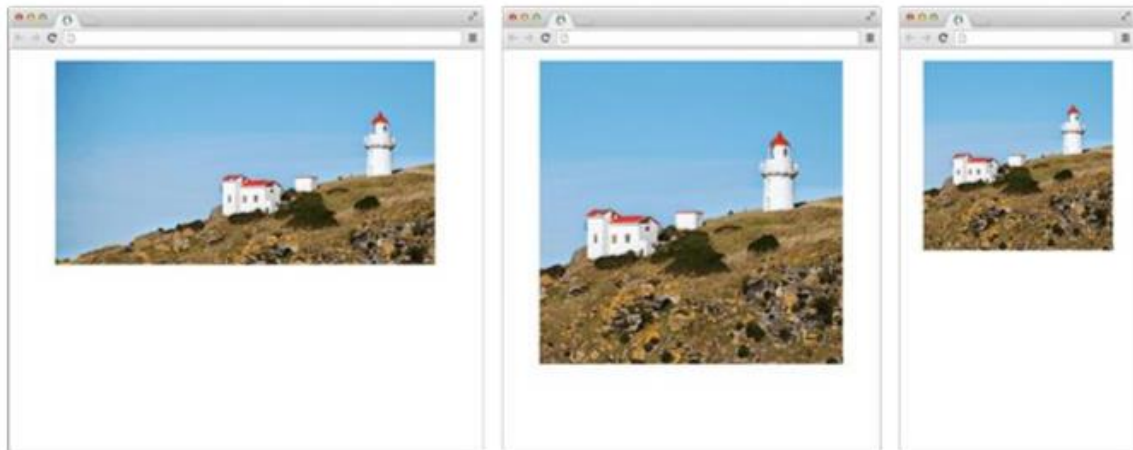
## Ejemplo en función de la anchura

Anteriormente se explicaba que cuando se desconoce el tamaño definitivo de una imagen, no es posible indicar el descriptor relacionado con la densidad de píxeles mencionado en la sección anterior. Así que, en vez de definir imágenes de anchura fija, se puede añadir un descriptor para que el navegador calcule automáticamente la densidad de píxeles y así descargue la mejor imagen en cada caso.

Esto se puede combinar con la etiqueta `<picture>` tal y como se muestra a continuación.

```
<picture>
  <source
    media="(min-width: 600px)"
    sizes="(min-width: 800px) 65vw, 80vw"
    srcset="images/lighthouse-landscape-400.jpg 400w,
           images/lighthouse-landscape-800.jpg 800w,
           images/lighthouse-landscape-1200.jpg 1200w">
  
</picture>
```

Siguiendo con este mismo ejemplo, cuando el navegador tiene una anchura de 600px o superior, se carga la imagen panorámica del faro y si no se carga la imagen con orientación vertical, de acuerdo con el atributo `media`.



Además, por cada resolución tenemos tres versiones de la misma imagen. En resoluciones mayores de 600px, tenemos a su vez dos tamaños: de 65vw a partir de 800px y con 80vw en el resto (es decir de 600px a 800px). En resoluciones menores de 600px la imagen siempre ocuparía 80vw. A continuación se exponen una serie de ejemplos para aclarar el funcionamiento del código anterior:

- ✓ Resolución de 700px y DPI 1x. La imagen ocupa 560px (son 80vw por tanto el 80% de 700px). El navegador selecciona la imagen `images/lighthouse-landscape-800.jpg` de 800w.
- ✓ Resolución de 1000px y DPI 2x. La imagen ocupa 1300px (son 1000px multiplicado por 2 al ser DPI 2x y aplicando el ancho de pantalla de 65vw). El navegador selecciona la imagen `images/lighthouse-landscape-1200.jpg` de 1200w.
- ✓ Resolución de 500px y DPI 1x. La imagen ocupa 400px (son 80vw por tanto el 80% de 600px). El navegador selecciona la imagen `images/lighthouse-400.jpg` de 400w.

- ✓ Resolución de 550px y DPI 3x. La imagen ocupa 1650px (son 550px multiplicado por 3 al ser DPI 3x y aplicando el ancho de pantalla de 65vw serían 1072,5px). Por tanto, el navegador selecciona la imagen `images/lighthouse-1200.jpg` de 1200w.

### Ejemplo en función del contenedor

Imaginemos el ejemplo anterior, pero resulta que se definen los siguientes estilos:

```
source, img {
    /* Se fuerza el 50%, pero se coge la imagen que corresponda en
    picture */
    width: 50%;
}
```

Por un lado, habíamos definido los tamaños en sizes en la unidad vw. Además se comentaba en apartados anteriores que sizes no admite porcentajes.

Por tanto, de esta manera se selecciona la imagen que corresponda según los atributos srcset y sizes y con CSS se puede forzar a que luego realmente ocupe una unidad relativa al contenedor como ocurre en la imagen de debajo (que ocupa el 50% aplicando porcentajes).



### Ejemplo en función del formato

El atributo **type** del elemento `<source>` se puede utilizar para cargar formatos de imagen alternativos que pueden no estar soportados por el navegador del usuario. Si por ejemplo quieres servir imágenes en formato WebP para los navegadores que lo soportan, pero al mismo tiempo mantener las imágenes JPEG para el resto de navegadores, debes utilizar lo siguiente:

```
<picture>
  <source type="image/webp" srcset="images/butterfly.webp">
  
</picture>
```

## 4. INSERTAR AUDIO EN UNA WEB

Antes de estudiar con más profundidad como gestionar el audio en una página web, se expondrán una serie de conceptos relacionados con su representación como sonido digital.

- Frecuencia: Es el número de vibraciones por segundo que da origen al sonido analógico. El espectro de un sonido se caracteriza por su rango de frecuencias. Ésta se mide en **Hertzios (Hz)**.
- Tasa de muestreo (sample rate): Un audio digital es una secuencia de ceros y unos que se obtiene del muestreo de la señal analógica. La tasa de muestreo o sample rate define cada

cuánto tiempo se tomará el valor de la señal analógica para generar el audio digital. Esta tasa se mide en **Hertzios (Hz)**. Por ejemplo: 44100 Hz. nos indica que en un segundo se tomaron 44100 muestras de la señal analógica de audio para crear el audio digital correspondiente. Un audio tendrá más calidad cuanto mayor sea su tasa de muestreo. Algunas frecuencias estándares son 44100 Hz., 22050 Hz., y 11025 Hz.

- **Resolución (bit resolution):** Es el número de bits utilizados para almacenar cada muestra de la señal analógica. Una resolución de 8 bits proporciona 256 ( $2^8$ ) niveles de amplitud, mientras que una resolución de 16 bits alcanza 65536 ( $2^{16}$ ). Un audio digital tendrá más calidad cuanto mayor sea su resolución. Ejemplo: El audio de calidad CD suele ser un sonido de 44.100 Hz – 16 bits – estéreo.
- **Velocidad de transmisión (bitrate):** El bitrate define la cantidad de espacio físico (en bits) que ocupa un segundo de duración de ese audio. Por ejemplo, 3 minutos de audio MP3 a 128 Kbit/s, ocupa 2,81 MB de espacio físico ( $3\text{min} \times 60\text{seg/min} \times 128\text{Kbit/seg} = 23040\text{Kbits}$  ->  $(23040\text{Kbits} \times 1024\text{bits/Kbit} / 8\text{bits/bytes}) \times 1024\text{bytes/KBytes} \times 1024\text{KBytes/MBytes} = 2,81\text{MBytes}$  ó MB). En los audios en formato MP3 se suele trabajar con bitrates expresados en Kbps (kilobits por segundo). El audio tendrá más calidad cuanto mayor sea su bitrate y el archivo que lo contiene tendrá mayor peso.
- **Códec:** Acrónimo de "codificación/decodificación". Un códec es un algoritmo que reduce el número de bytes que ocupa un archivo de audio. Los archivos codificados con un códec específico requieren el mismo códec para ser decodificados y reproducidos. El códec más utilizado en audio es el MP3.

### 4.1. FORMATOS MÁS COMUNES

Al hablar del audio en Internet, el formato más conocido es **mp3**. Éste es un formato de compresión de audio digital patentado que usa un algoritmo con pérdida para conseguir un menor tamaño de archivo. Es un formato de audio común usado para música tanto en ordenadores como en reproductores de audio portátil.

Es importante destacar que, en lo que respecta al desarrollo web y al uso del audio para ser interpretado por los diferentes navegadores, mp3 no es único y universal, es decir, hay más formatos iguales o mejores y cada navegador soporta una serie de formatos. Por lo tanto, hay otras posibilidades de formatos que hay que conocer, sobre todo para, como luego veremos, permitir que la reproducción de audio sea lo universal posible entre todos los navegadores actuales.

**MP3 (MPEG 1 Layer 3):** Fue creado por el Instituto Fraunhofer y por su extraordinario grado de compresión y alta calidad está prácticamente monopolizando el mundo del audio digital.

- ✓ Es ideal para publicar audios en la web. Se puede escuchar desde la mayoría de los reproductores.
- ✓ La transformación de WAV a MP3 o la publicación directa de una grabación en formato MP3 es un proceso fácil y al alcance de los principales editores de audio.
- ✓ Tiene un enorme nivel de compresión respecto al WAV. En igualdad del resto de condiciones reduciría el tamaño del archivo de un fragmento musical con un factor entre 1/10 y 1/12.
- ✓ Presentan una mínima pérdida de calidad.

**Ogg:** Es un formato válido para audio y vídeo, desarrollado por la Fundación Xiph.org. Su principal ventaja con respecto a mp3 es que es un formato libre de patentes y abierto diseñado para dar un alto grado de eficiencia en el streaming y la compresión de archivos. De manera incorrecta, muchas veces los archivos ogg se les llama Vorbis, ya que éste fue el primer códec desarrollado para leer este formato. Sin embargo, hay muchos otros códec además de Vorbis que interpretan ogg. Algunas características de este formato son:

- ✓ Muestra un grado de compresión similar al MP3 pero según los expertos en música la calidad de reproducción es ligeramente superior.
- ✓ No todos los reproductores multimedia son capaces de leer por defecto este formato. En algunos casos es necesario instalar los códec o filtros oportunos.
- ✓ El formato OGG puede contener audio y vídeo.

**Real Audio:** La gran mayoría de las páginas que usan streaming utilizan el formato Real Audio para difundir sus contenidos. RealNetworks es la compañía que ha creado Real Audio y también ha creado un reproductor para su formato, el RealPlayer (del que existe una versión gratuita y otra comercial).

**Windows Media Audio (wma):** Es un formato desarrollado por Microsoft. Ofrece gran calidad de sonido en un tamaño reducido. Sus características más importantes son:

- ✓ Como ventaja para las compañías y los músicos, el formato de Microsoft incluye un sistema de gestión de los derechos de autor para evitar la piratería y la creación de copias no autorizadas de las canciones.
- ✓ Resulta muy adecuado para la reproducción de archivos en streaming.
- ✓ Microsoft ha desarrollado un reproductor propio para escuchar su formato de audio, el Windows Media Player que se encuentra disponible en su web.
- ✓ La desventaja de wma es que, al ser de Microsoft, los navegadores Internet Explorer o Edge lo soportan, pero no se puede asegurar en el resto.

**Wav o wave (WAVEform audio file format):** Es un formato de audio que se suele utilizar sin compresión de datos. Es propiedad de Microsoft y de IBM. Este formato es muy conocido en entornos Windows, pero no se suele usar en Internet por ocupar mucho espacio al ser usado sin compresión. Es más propio para guardar audios originales a partir de los cuales se puede comprimir y guardar en distintos tamaños de muestreo para publicar en la web. Los archivos WAV se pueden guardar con distintos tipos de compresión (por ejemplo, PCM o ADPCM), no obstante, incluso definiendo un sistema de compresión, con un audio de cierta duración se genera un archivo excesivamente pesado.

## 4.2. COMPARATIVA Y CONVERSIÓN ENTRE FORMATOS

Un hándicap importante a la hora de trabajar con audio en páginas web es saber cómo implementa cada navegador los reproductores. No todos los navegadores tienen reproductores para todos los formatos, sino que hay preferencias y lo que un navegador puede reproducir otro no. Esta situación lo único que hace es complicar el trabajo del desarrollador, ya que éste debe intentar garantizar que los audios que inserte en un sitio web puedan ser reproducidos por todos los navegadores, y si eso no es posible, por aquellos más utilizados por el usuario objetivo del portal.

La siguiente tabla muestra una comparativa de qué formatos son reproducidos por los 5 navegadores más destacados actualmente (bajo Windows). Esta lista no es definitiva, ya que la vertiginosa actualización de los navegadores hace que las prestaciones que ofrecen en una versión sean mejoradas o reducidas en las siguientes versiones.

Navegador	MP3	Wav	Ogg
IE	SI	NO	NO
Chrome	SI	SI	SI
Firefox	SI	SI	SI
Safari	SI	SI	SI (parcialmente)
Opera	SI	SI	SI

Recuerda que en direcciones como [www.caniuse.com](http://www.caniuse.com) puedes comprobar la compatibilidad los diferentes formatos en cada navegador.

Por otro lado, herramientas de conversión hay muchas disponibles en Internet. Algunas gratuitas son:

- **Free Studio:** Es una potente herramienta para convertir todo tipo de archivos (vídeo, audio, imagen). Es fácil de usar y muy adecuada para un diseñador que trabaja con audio en la web. Solo versión para Windows.
- **Audacity:** Es una herramienta avanzada para grabar y editar. Es libre y de código abierto. Su funcionalidad es mucha, y entre ella está la de convertir archivos. Es muy adecuada si el diseñador desea crear sus propios sonidos. Para Linux y Windows.

También hay conversores online, servicios web a los que se le sube el archivo de audio, se selecciona el formato al que se quiere convertir y el resultado se envía por email. Un ejemplo de este tipo de herramientas es AConvert, sin embargo, hay muchas más disponibles en Internet.

### 4.3. LA ETIQUETA AUDIO

Vistos los diferentes formatos de audio adecuados para Internet, cuáles son soportados por los navegadores más adecuados y herramientas de conversión de formatos, en este punto se muestra la alternativa actual para introducir audio en una página web.

#### 4.3.1. ALTERNATIVAS OBSOLETAS

**<bgsound>:** Es una alternativa aparentemente adecuada, pero con el inconveniente de que es solo entendible por Internet Explorer.

```
<bgsound src="audios/Beethoven.wma" loop="2" volume="20" />
```

**<embed>:** Es una alternativa para insertar complementos (plugins) de audio y vídeo que es compatible para todos los navegadores. Sin embargo, no está incluida dentro del estándar W3C por lo que su implementación depende de los navegadores, y no todos la interpretan igual. Además, dependiendo del formato del archivo, así lo interpretarán los navegadores.

```
<embed src="audios/Aretha.mp3" height="0" type="audio/mpeg"
loop="false" controller="false">
```

**Descargar/ejecutar audio por el usuario (en HTML4):** La manera más sencilla de incluir sonidos es dejando al usuario la decisión de escucharlos o no en local. En realidad, es más una opción descargar que reproducir. Para ello se puede colocar el nombre del archivo de sonido en un atributo href de la etiqueta <a>. El siguiente código muestra un ejemplo.

```
<a href="audios/Beethoven.wma">Audio de Beethoven</a>
```

#### 4.3.2. ESTRUCTURA Y PARÁMETROS

En HTML5 la inclusión de sonidos en páginas web intenta paliar muchos de los inconvenientes vistos con las etiquetas **<bgsound>** o **<embed>**. De alguna manera, HTML5 ofrece una alternativa más sencilla y eficaz para que el desarrollador pueda insertar audio.

La propuesta de HTML5 utiliza la etiqueta **<audio>**. Esta etiqueta es nueva en HTML5 y no existe para versiones de HTML4. Evidentemente, eso obliga a que los navegadores que la interpreten deban implementar HTML5 en general (y esta etiqueta en particular). Los atributos de esta etiqueta son los siguientes.

- ✓ **autoplay:** Especifica que el audio debería empezar automáticamente tan pronto como esté listo para reproducirse (el único valor de este atributo es autoplay y se puede omitir indicando únicamente "autoplay"). IMPORTANTE: El atributo autoplay ya no funciona en la mayoría de navegadores. Desde algunos se puede configurar, pero en otros se puede solucionar este problema añadiendo el atributo **muted** que permite reproducir el audio automáticamente, pero desactivando el sonido.
- ✓ **controls:** Indica que los controles de reproducción serán visibles (el único valor posible de este atributo es controls y se puede omitir indicando únicamente "controls").
- ✓ **loop:** Indica que el audio se repetirá automáticamente cuando termine (el único valor posible de este atributo es loop y se puede omitir indicando únicamente "loop").
- ✓ **preload:** Especifica si el audio debería ser cargado cuando la página se carga o no, es decir, se empieza a cargar cuando el usuario lo quiera reproducir (el valor auto indica que se cargará todo el archivo cuando se cargue la página, el valor meta indica que solo se cargarán los metadatos del archivo y none que no habrá precarga).
- ✓ **src:** Especifica la URL de archivo de audio, ya sea con rutas absolutas o relativas.

Un ejemplo del uso de esta etiqueta para obtener un reproductor con controles para un audio que se inicia solo sería:

```
<audio src="audios/Aretha.mp3" controls autoplay></audio>
```

Otro ejemplo para reproducir audio automáticamente y mientras la página siga abierta sin mostrar controles es el siguiente:

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

Nótese, que por defecto no se muestran los controles y si queremos tener la opción de iniciar o parar manualmente el audio es necesario emplear el atributo **controls**.



Una mejora para hacer más versátil esta opción es usar la etiqueta **<source>** dentro de **<audio>** como se muestra en el código de debajo. De esta manera, el navegador intenta cargar la primera línea **<source>**. Si falla o no es soportada esa reproducción, entonces pasa a la siguiente. Por ejemplo, si Internet Explorer, Safari y Google Chrome leen archivos mp3 y las primeras versiones de Firefox leen archivos ogg (no soporta mp3), el siguiente código ofrece una solución para las cuatro versiones de estos navegadores. Cuando el navegador Firefox llegue al primer **<source>** y no pueda interpretar mp3 entonces pasará al segundo con ogg. Sin embargo, Internet Explorer, Safari y Google Chrome se quedarán con el primer **<source>**.

Aquellos navegadores que no soporten la etiqueta audio mostrarán el mensaje que se muestra al final de la etiqueta.

```
<audio controls="controls" preload="auto">
  <source src="music.mp3" type="audio/mpeg"/>
  <source src="music.ogg" type="audio/ogg"/>
  El navegador no soporta el audio
</audio>
```

Otra opción para mejorar más aun la compatibilidad entre navegadores, es combinar la etiqueta **<source>** con **<object>**. Combinando **<object>** con **<embed>** se puede, por ejemplo, insertar un reproductor (Flash) para aquellos navegadores que no soporten el código mostrado arriba (por ejemplo versiones anteriores a Internet Explorer 8).

```
<audio controls="controls" preload="auto">
  <source src="Aretha.mp3" type="audio/mpeg"/>
  <source src="Aretha.ogg" type="audio/ogg" />
  <object type="application/x-shockwave-flash"
    data="player_mp3_mini.swf" width="200" height="20">
    <param name="movie" value="player_mp3_mini.swf" />
    <param name="bgcolor" value="#085c68" />
    <param name="FlashVars" value="mp3=Aretha.mp3" />
    <embed href="player_mp3_mini.swf" bgcolor="#085c68"
      width="200" height="20" name="movie" align=""
      type="application/x-shockwave-flash"
      flashvars="mp3=Aretha.mp3">
    </embed>
  </object>
</audio>
```

Para que este código funcione es necesario que el archivo **player\_mp3\_mini.swf** esté disponible en la misma carpeta (local o en web) que la página HTML. Este archivo es el que carga **<object>** cuando ninguna de las otras opciones son soportadas. La etiqueta **<object>** tiene varios parámetros que definen el tamaño del elemento que quiere embeber, en este caso el reproductor Flash. Sin embargo, es con la etiqueta **<embed>** con la que realmente se está incluyendo el archivo mp3 que se quiere cargar en ese reproductor. De alguna manera, con este código se pretende insertar un audio mp3 con la etiqueta **<embed>** vista anteriormente, pero añadiendo directamente el reproductor, y no suponer que el navegador lo tiene.

A la etiqueta **<audio>** también se le puede aplicar estilos CSS. Por ejemplo, el siguiente código define un estilo para hacer que una etiqueta audio que use la clase **audio-fondo** aparezca con un color de fondo amarillo y un tamaño determinado.

```
<style>
    .audio-fondo {
        width: 160px;
        height: 36px;
        background: #FFFF00;
    }
</style>
```

A modo de conclusión de lo visto, aunque no todos los navegadores actuales soportan HTML5 con la madurez deseada, y no todos los usuarios usan los navegadores debidamente actualizados para garantizar que soportan HTML5, la etiqueta audio ofrece muchas ventajas que aconsejan su uso futuro a la hora de insertar audio en sitios web. HTML5 permite que los navegadores ya no necesiten complementos (plugins) del estilo Adobe Flash, Microsoft Silverlight, etc. para reproducir audio y eso le facilita al desarrollador no tener que preocuparse si un navegador reproduce o no un audio determinado.

Los dispositivos móviles actuales integran navegadores que soportan HTML5 por lo que es cuestión de poco tiempo que todos los usuarios tengan navegadores HTML5 y no sea necesario buscar alternativas con <object> o JavaScript para detectar que audio puede reproducir un complemento determinado de un navegador concreto. Aunque será inevitable que en la etiqueta <audio> se den opciones variadas respecto al tipo de archivo (mp3, ogg, etc.).

Por último, como opción de diseño avanzada, se puede usar **JavaScript y HTML5 para personalizar el reproductor**. El siguiente código utiliza botones propios para reproducir, en vez de aquellos que vienen por defecto (para ello no se define el atributo controls). Esta opción da mucho juego en el diseño del reproductor al poder cambiar los botones por imágenes acordes con el estilo del sitio.

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

## 5. INSERTAR VIDEO EN UNA WEB

Antes de estudiar con más profundidad como gestionar el vídeo en una página web, se expondrán una serie de conceptos relacionados con su representación digital.

- **Dimensiones:** Es el tamaño del video (ancho x alto) expresado en píxeles cuando se visualiza al 100%, sin agrandar ni reducir. Los reproductores pueden mostrar un video a pantalla completa o con una ampliación del 200%, 300%, etc. En estos casos el video pierde calidad de imagen y esta pérdida depende del formato de archivo. Un video AVI

puede tener cualquier ancho y alto mientras que los estándares de VCD son 352 x 288 y de DVD 720 x 576.

- **Códec:** Acrónimo de "codificación/decodificación". Un códec es un algoritmo especial que reduce el número de bytes que ocupa un archivo de video. Los archivos codificados con un códec específico requieren el mismo códec para ser decodificados y reproducidos. Ejemplos: Xvid, Theora, RealVideo, WMV, MPEG-4, H264, etc.
- **Velocidad de transmisión (bitrate):** El bitrate define la cantidad de espacio físico (en bits) que ocupa un segundo de duración de ese video. El video tendrá más calidad cuanto mayor sea su bitrate y el archivo que lo contiene tendrá mayor peso. El bitrate puede ser fijo o variable. El bitrate variable consigue mayor calidad de imagen porque recoge más calidad en escenas muy cargadas o con mucho movimiento y ahorra en aquellas más estáticas.
- **Fotogramas por segundo:** Un video resulta de la exposición imágenes o fotogramas uno detrás de otro. Un parámetro de la calidad del video es el número de fotogramas por segundo que muestra durante su reproducción. Este valor oscila entre 15 y 30. Por ejemplo los vídeos en DVD en Europa exhiben 25 fotogramas por segundo (25 fps).
- **Proporción o ratio de aspecto:** Es la proporción entre la anchura y altura de un video. Cuando se reproduce un video se suele mantener por defecto esta proporción para evitar deformación de las imágenes. Por este motivo cuando se elige la visualización a pantalla completa, aparecen franjas negras arriba y abajo. Es habitual una relación 4:3 para los videos domésticos (352x288 píxeles, por ejemplo) mientras que en DVD se suele trabajar con ratios de 16:9.

## 5.1. FORMATOS MÁS COMUNES

Los videos digitales se pueden guardar en archivos de distintos formatos. Cada uno se corresponde con una extensión específica del archivo que lo contiene. Existen muchos tipos de formatos de video. Aquí se citan algunos de los más utilizados.

Asimismo, cada tipo de archivo admite en cada momento un códec de compresión distinto. Estos formatos son en realidad formatos contenedores, es decir, un tipo de formato de archivo que almacena información de vídeo, audio, subtítulos, capítulos, metadatos e información de sincronización siguiendo un formato preestablecido en su especificación.

**AVI (Audio Video Interleaved = Audio y Video Intercalado):** Es el formato estándar para almacenar video digital. Cuando se captura video desde una cámara digital al ordenador, se suele almacenar en este formato con el códec DV (Digital Video). Algunas de sus características más representativas son:

- ✓ Admite distintos códecs de compresión como AC3/DivX o MP3/Xvid, entre otros, etc. Los códecs con más capacidad de compresión y una calidad aceptable son DivX y XviD.
- ✓ El formato AVI puede ser visualizado con la mayoría de los reproductores: Windows Media, QuickTime, etc. siempre y cuando se encuentren instalados en el equipo los adecuados códecs para cada tipo de reproductor.
- ✓ Es ideal para guardar videos originales que han sido capturados de la cámara digital (codificados con DV).
- ✓ El archivo AVI puede contener video con una calidad excelente. Sin embargo, el peso del archivo resulta siempre muy elevado y no es recomendable para publicarlo en internet por este motivo.

**MP4 o MPEG (Moving Pictures Expert Group = Grupo de Expertos de Películas):** Es un formato estándar para la compresión de video digital:

- ✓ Son archivos de extensión \*.MPG, \*.MPEG y actualmente sobre todo \*.MP4.
- ✓ Admite distintos tipos de códecs de compresión: MPEG-1 (calidad CD), MPEG-2 (calidad DVD), MPEG-3 (orientado al audio MP3) y MPEG-4 o MP4 (más orientado a la web).
- ✓ Se reproducen normalmente con Windows Media Player y QuickTime.

**MOV:** Es el formato de video y audio desarrollado por Apple.

- ✓ Utiliza un códec propio que evoluciona en versiones con bastante rapidez.
- ✓ Este tipo de archivos también pueden tener extensión \*.QT
- ✓ Se recomienda utilizar el reproductor de QuickTime.
- ✓ Es ideal para publicar videos en Internet por su razonable calidad/peso.
- ✓ Admite streaming.

**WMV:** Es un formato que ha sido desarrollado por Microsoft.

- ✓ Utiliza el códec MPEG-4 para la compresión de video.
- ✓ En principio sólo se puede visualizar con una versión actualizada de Windows Media 7 o superior.
- ✓ Es ideal para publicar videos en Internet por razonable calidad/peso.
- ✓ Admite streaming.

**FLV:** Es un formato que utiliza el reproductor Adobe Flash para visualizar vídeo en Internet.

- ✓ Son archivos de extensión \*.FLV. SWF y .F4V
- ✓ Se pueden reproducir desde distintos reproductores locales: MPlayer, VLC media player, Riva, Xine, etc.
- ✓ Opción recomendada para la web por su interoperabilidad. Al visualizarse a través del reproductor de Flash es accesible desde la mayoría de los sistemas operativos y navegadores web.
- ✓ Los repositorios de vídeo más conocidos en Internet aceptan este formato para la difusión de vídeos: YouTube, Google Video, iFilm, etc.
- ✓ Permite configurar distintos parámetros del vídeo para conseguir una aceptable calidad/peso. Admite streaming.
- ✓ Sin embargo, debido a la reciente apuesta por el HTML5, incluyendo el nuevo tag <video>, los vídeos codificados en FLV se han reducido hasta casi la extinción completa en Internet.

**OGG y OGV:** ogv es el correspondiente contenedor Open Source de la Fundación Xiph.Org. Apropiado para contener el códec Theora.

**MKV (Matroska):** Es un formato Open Source que puede contener casi cualquier tipo de formato de vídeo. Muy usado originalmente para comprimir películas que se han de compartir por Internet.

**WEBM (WebM):** Es un contenedor de vídeo Open Source desarrollado por Google orientado para usarse con HTML5. Está compuesto por el códec VP8 y el códec de audio Vorbis (ogg) dentro de un contenedor multimedia Matroska.

## 5.2. COMPARATIVA Y CONVERSIÓN ENTRE FORMATOS

Como ocurre con el audio, existen muchas aplicaciones que permiten convertir de un formato de vídeo a otro. En el desarrollo web, estas aplicaciones ofrecen una alternativa sencilla para poder dar soluciones flexibles para la mayor cantidad posible de navegadores.

Al igual que con el audio, existe una "batalla" entre fabricantes de navegadores sobre que formato de vídeo debe de ser el estándar y por supuesto no todos reproducen los mismos formatos de forma nativa. Al usar HTML5 ocurre igual que con los ficheros de audio, unos navegadores contemplan de manera nativa unos códec y otros apuestan por otros, y las diferencias son irreconciliables. La siguiente tabla muestra los códec en los principales navegadores.

En los ejemplos siguientes se emplearán tres de los formatos más usados con HTML5, que emplean los siguientes códec para vídeo y audio:

- mp4 = H.264 (vídeo) + AAC (audio)
- ogg / ogv = Theora (vídeo) + Vorbis (audio)
- webm = VP8 (vídeo) + Vorbis (audio)

Navegadores	MP4	WebM	Ogv
Internet Explorer	SI	NO	NO
Chrome	SI	SI	SI
Firefox	SI	SI	SI
Safari	SI	Parcialmente	NO
Opera	SI (desde Opera 25)	SI	SI

Por otro lado, herramientas de conversión hay muchas disponibles en Internet y online.

Algunas herramientas de escritorio gratuitas son:

- **Miro Video Converter:** Es una utilidad muy sencilla para convertir a cualquier formato de vídeo.
- **Free Studio:** Es una potente herramienta para convertir todo tipo de archivos (video, audio, imagen). Es fácil de usar y muy adecuada para un diseñador que trabaja con audio y vídeo en la web. Solo versión para Windows. La versión 5.2.1 y anteriores no soporta conversión a ogv.
- **AtubeCatcher:** Es una utilidad muy interesante. Además de convertir archivos a una gran cantidad de formatos, permite descargar (en varios formatos) vídeos de los repositorios más conocidos (Youtube, Google Videos o Vimeo).

### 5.3. LA ETIQUETA VIDEO

Para aprender a insertar vídeo en páginas web nos centraremos básicamente en las posibilidades de HTML5. Tal y como se vio en la Sección anterior, HTML5 ofrece soluciones mucho más ventajosas para insertar audio que su predecesor HTML4. Y para el caso del vídeo ocurre lo mismo. En HTML5 hay una etiqueta **<video>** que permite embeber archivos de vídeo de forma nativa sin necesidad de complementos (plugins) adicionales.

La etiqueta **<video>** es muy parecida a **<audio>**: dispone de los atributos **autoplay**, **loop** y **preload**, con la misma sintaxis y semántica que en **<audio>**. También se puede especificar la fuente de un archivo, bien usando el atributo **src** o bien usando la etiqueta **<source>**.

Además, se puede utilizar los controles que ofrece el navegador de forma nativa utilizando el atributo **controls** o bien puedes ofrecer tus propios controles en JavaScript. De la misma forma, los controles del navegador solo aparecen cuando se especifica el atributo **controls**.

Sin embargo, a diferencia del audio, el vídeo necesita un tamaño para reproducirse, tal y como ocurre con las imágenes. Para ello se pueden usar los atributos **height** (alto) y **width** (ancho) con más sentido que en **<audio>**. Estos atributos se pueden sustituir por las propiedades análogas en CSS. Además, como ocurre en audio, también a la etiqueta **<video>** se le pueden aplicar estilos con CSS. Un ejemplo de uso de esta etiqueta con un archivo mp4 es el siguiente:

```
<video controls loop poster="images/portada.png"
src="videos/recetapollo.mp4" width="360" height="240"></video>
```

Al margen de los atributos vistos hasta ahora, la etiqueta **<video>** anterior también tiene un atributo **poster** en el que se indica una imagen que se usará como portada antes de que el vídeo empiece a reproducirse.

El ejemplo anterior usa un archivo mp4 lo que hace que su reproducción no funcione en todos los navegadores. Entonces, para hacer la reproducción adecuada para la mayor cantidad de navegadores posible, se puede utilizar la etiqueta **<source>** igual que se hizo en **<audio>**.

```
<video autoplay width="360" height="240">
  <source src="videos/recetapollo.mp4" type="video/mp4">
  <source src="videos/recetapollo.ogv" type="video/ogg">
  <source src="videos/recetapollo.webm" type="video/webm">
  No está soportado el vídeo
</video>
```

Además, el ejemplo anterior no tiene el atributo **controls** y por tanto no da al usuario la opción de iniciar o parar el vídeo. Por tanto, la única alternativa es que se reproduzca automáticamente con **autoplay**. A tener en cuenta que ocurre lo mismo que en la etiqueta **<audio>** y en la mayoría de navegadores no se permite. De la misma manera se puede solucionar con el atributo **muted** para desactivar el sonido.

En cualquier caso, la etiqueta **<video>** es solo válida para navegadores que soportan HTML5. Si se desea tener en cuenta a versiones anteriores, por ejemplo, a Internet Explorer 8, se puede utilizar etiquetas **<object>** (o usar JavaScript). Esta solución es similar a la vista para audio.

```
<video autoplay width="360" height="240">
  <source src="videos/recetapollo.mp4" type="video/mp4">
  <source src="videos/recetapollo.ogv" type="video/ogg">
  <source src="videos/recetapollo.webm" type="video/webm">
  <object type="application/x-shockwave-flash"
    data="player_flv_maxi.swf" height="240" width="360">
    <param name="movie" value="player_flv_maxi.swf">
    <param name="FlashVars" value="flv=videos/recetapollo.flv">
    <embed type="application/x-shockwave-flash" width="360"
      height="240" src="player_flv_maxi.swf"
      flashvars="flv=videos/recetapollo.flv">
    </embed>
  </object>
  No está soportado el vídeo
</video>
```

Para que este código funcione es necesario que el archivo `player_flv_maxi.swf` esté disponible en la misma carpeta (local o en web) que la página html. Este archivo es el que carga `<object>` cuando ninguna de las otras opciones es soportada. La etiqueta `<object>` tiene varios parámetros que definen el tamaño del elemento que quiere embeber, en este caso el reproductor Flash.

Por último, también es posible utilizar vídeos disponibles en cualquier repositorio de vídeo como, por ejemplo, Youtube. Estos repositorios ofrecen código para incrustar un vídeo determinado en una página web. Actualmente, el código proporcionado utiliza la etiqueta `<iframe>` en sustitución a `<object>` con el fin de dar soporte más optimizado a páginas web para dispositivos móviles tipos iPad.