

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DE TELECOMUNICACIÓN  
UNIVERSIDAD POLITÉCNICA DE CARTAGENA



**Proyecto Fin de Carrera**

Desarrollo de un entorno multilingüe para la plataforma wifitickets



AUTOR: Ilham Nouari Ben Attou  
DIRECTOR: Francesc Burrull i Mestres  
02/ 2015



<b>Autor</b>	Ilham Nouari Ben Attou
<b>E-mail del Autor</b>	Ilham_n_60@hotmail.com
<b>Director(es)</b>	Francesc Burrull i Mestres
<b>E-mail del Director</b>	francesc.burrull@upct.es
<b>Codirector(es)</b>	
<b>Título del PFC</b>	Desarrollo de un entorno multilingüe para la plataforma wifitickets
<b>Descriptores</b>	HTML5, Css3
<b>Resumen</b>	<p>La plataforma wifitickets ofrece acceso condicional a Internet, al estilo de un tarificador clásico. Se pretende dotar a la plataforma diseñarla en la nueva tecnología, es decir HTML5 usando estilos CSS3, Javascript, JQuery en lo cual de integrar en la página animaciones, en su vez hemos usado esta tecnología para que se viera en los dispositivos tanto móviles como tablets</p>
<b>Titulación</b>	Ingeniería de Telecomunicación especialidades Telemática

# Agradecimientos

A mis padres y mis hermanos por el apoyo que me han dado durante todos estos años de carrera, sin olvidarme por supuesto de mi Marido que siempre ha estado conmigo en los momentos difíciles.

A mi director del proyecto, Francesc Burrull i Mestres por su ayuda y predisposición.

A todos vosotros.

## Índice general

1.	Introducción y objetivos.....	1
1.1.	Introducción .....	1
1.2.	Objetivos .....	2
2.	Introducción a HTML5 .....	5
2.1.	<i>HyperText Markup Language 5</i> (HTML5).....	6
2.2.	Estándares de la Web. HTML4.X vs HTML5.....	6
2.3.	Algunas características que aporta HTML5 .....	8
2.4.	Nivel de implementación en los navegadores de HTML5 .....	9
2.5.	Especificación borrador HTML5.0.....	12
2.6.	HTML5 y elementos relacionados con el contenido multimedia.....	21
3.	Accesibilidad Web .....	31
3.1.	Legislación y normativa relativa a la accesibilidad .....	31
3.2.	Iniciativa de Accesibilidad Web (WAI) .....	33
3.3.	Accesibilidad al contenido multimedia en la Web .....	43
4.	Accesibilidad en elementos relativos al contenido multimedia en HTML5.....	55
4.1.	Elementos semánticos .....	56
4.2.	Elemento Track .....	60
5.	Diseño y desarrollo de interfaz basado en HTML5 .....	61
5.1.	Requisitos de accesibilidad de un contenido multimedia en la Web .....	62
5.2.	Diseño de Interfaz que reproduzca contenido multimedia accesible .....	62
5.3.	Implementación en HTML5 de la Interfaz .....	64

# Capítulo 1

## 1. Introducción y objetivos

### 1.1. Introducción

En la última década, hemos sido testigos del gran cambio que ha provocado Internet en nuestras vidas. No teníamos ni idea de cómo un invento que se creó para unos pocos, ha sido una revolución para muchos y que ha hecho que Internet sea un elemento imprescindible para el día a día, ya sea en el ámbito personal como en el profesional.

En un principio, el origen de Internet fue la investigación de computadores más avanzados por parte del Departamento de Defensa de los EE.UU, pero con el paso del tiempo se fue popularizando y extendiendo a lo largo del mundo y se hizo público con el fin de darse a conocer y hacerse accesible para cualquier ciudadano.

Poco a poco se fueron investigando las características que ofrecía y fueron surgiendo lenguajes para dar soporte a esta nueva forma de comunicarse, y el principal fue el lenguaje de Marcado de Hipertexto o *HyperText Markup Language* (HTML). Éste nuevo lenguaje dio la posibilidad de compartir información con el resto del mundo, principalmente por texto, ya que éste lenguaje está basado en texto (o hipertexto como se denomina técnicamente), pero con el paso del tiempo surgieron nuevos lenguajes destinados a complementar al HTML, como son JavaScript, Flash, etc. que permitieron compartir diversos tipos de información, ya no sólo de texto sino imágenes, fotos, videos, animaciones, juegos, es decir, contenido multimedia que dotaba a los sitios web de un contenido más vistoso y agradable de compartir que el mero hecho de leer sin más.

Por tanto, todos estos nuevos lenguajes necesitaban unas normas que los programadores siguiesen para que la Web no sea un caos. Para garantizar la accesibilidad en la Web existe la *Web Accessibility Initiative* (WAI), iniciativa del *World Wide Web* (W3C) que ofrece unas pautas y reglas para permitir el acceso a contenido web a todo tipo de usuarios además de proponer normas a seguir para la estructura de una página web, y proporciona estándares de

distintos lenguajes como HTML, CSS (*Cascading Style Sheets*), etc., fundamentales para una buena construcción de sitios web. El Consorcio W3C es una comunidad internacional donde las organizaciones miembro, personal y el público en general, trabajan conjuntamente para desarrollar estándares web [W3C, 2011 c]. Desde entonces, la accesibilidad se convirtió en algo indispensable, y además de que una página web tenga un diseño de interfaz atractivo, su contenido debe ser accesible para cualquier tipo de persona.

El objetivo de este proyecto es diseñar en esta nueva tecnología la plataforma de wifitickets para relacionar con contenido multimedia de tipo animación de fotos por ejemplo, y que se observa bien en los dispositivos esta nueva versión de lenguaje de marcado aun en desarrollo y en camino de estandarizarse como es el nuevo HTML5.

## 1.2 Objetivo

La plataforma wifitickets ofrece acceso condicional a Internet, al estilo de un tarificador clásico. Se pretende dotar a la plataforma diseñarla en la nueva tecnología, es decir HTML5 usando estilos CSS3, Javascript, JQuery en lo cual de integrar en la página animaciones, en su vez hemos usado esta tecnología para que se viera en los dispositivos tanto móviles como tablets.

# Capítulo 2

## 2. Introducción a HTML5

En este capítulo se presenta una introducción al estado de la cuestión en el que se encuentran la tecnología base que se va a tratar en este proyecto principalmente, el HTML, siendo su versión más reciente, y actualmente en borrado camino de convertirse en estándar, HTML5 [W3C, 2011 a]. Se ofrecerá una introducción de lo que fue la versión anterior HTML4.01 y sus principales diferencias con HTML5, que son varias y significativas.

Una de las diferencias que hay con este nuevo estándar es la creación de nuevas etiquetas que van a dar más significado semántico a la Web, como etiquetas específicas para diferentes zonas de un sitio web, como por ejemplo para el encabezado, menús laterales, pies de página, y cuerpo principal, todas ellas con una etiqueta especialmente creada para la ocasión. También se debe comentar las posibles futuras aplicaciones que tendrá HTML5, como la Web semántica, que gracias a estas nuevas etiquetas se dará un mayor significado a la Web y dotará a los motores de búsqueda una mayor facilidad a la hora de encontrar resultados más ajustados a lo que el usuario demande.

Una de las características más esperadas y que más expectación ha dado es la estandarización en la forma de incluir vídeo y audio en la Web, que permita la inclusión de estos elementos multimedia sin necesidad de ninguna aplicación externa, lo que facilita enormemente un acceso más accesible y usable a estos contenidos en la Web, y que más adelante se verá con más detalle.

Otro importante factor a considerar es cómo se está implementando el HTML5 en los distintos agentes de usuario web (navegadores), ya que este software es una pieza fundamental para un correcto acceso de un sitio web. Se podrá observar mediante un estudio que se incluye, el nivel de aceptación de los navegadores con el nuevo estándar HTML5.

Otra de las características más interesantes que incorpora HTML5 según objetivos del W3C es la posibilidad de incluir subtítulos a nivel de código, para facilitar al desarrollador la introducción de estos recursos. Actualmente esta funcionalidad no ha sido implementada, pero se espera que un futuro se pueda contar con este esperado recurso. Esta funcionalidad es uno de los pilares fundamentales para otorgar de accesibilidad al contenido multimedia, por lo que mientras tanto, se puede aportar esta funcionalidad mediante otras tecnologías como JavaScript

o Flash, tal como se ha tenido que hacer en este proyecto con la utilización de tecnología JavaScript.

## 2.1. *HyperText Markup Language 5 (HTML5)*

HTML5 es la nueva versión del lenguaje de marcado que se usa para estructurar páginas web, que actualmente está en desarrollo y no está terminado y con el paso del tiempo contaremos gracias a él con características nuevas y modificaciones que mejorará significativamente este nuevo estándar.

En esta sección se presenta una breve introducción de las diferencias entre la anterior versión HTML4.01 y la nueva de HTML5, para a continuación explicar en profundidad este nuevo estándar.

## 2.2. Estándares de la Web. HTML4.X vs HTML5

HTML4 [W3C, 1998] y las diferentes versiones de la misma familia, ha sido un lenguaje muy longevo que ha tenido que adaptarse y renovarse, para mostrar el contenido de las páginas web tal y como lo conocemos ahora. Está claro que todo lo que podemos encontrar en una página web hoy en día, no estaba hace 10 o 15 años, cuando Internet no estaba tan extendido y no se sabía cómo podría afectar a nuestras vidas hoy en día, pero lo más importante que ha dado la Web es la expansión y la posibilidad a cualquier ciudadano, de crear su propia página web para plasmar sus ideas y sus opiniones. Para ello ha tenido que adaptarse para ofrecer lo nuevo que las nuevas tecnologías son capaces de incluir en la web.

El lenguaje HTML5 [W3C, 2011 a] es la nueva versión del lenguaje con el que estará escrita la Web, que es más que una renovación del HTML4.01 [W3C, 1999 a] que con el paso del tiempo se ha quedado obsoleto.

Con este nuevo borrador se pretende crear un estándar que poco a poco sustituya al anterior, introduciendo mejoras y eliminando elementos de la anterior versión que han quedado obsoletos, y por lo tanto, ya no se usan o han quedado anticuados para dar soporte a los nuevos contenidos web que se están desarrollando y a los tiempos actuales.

Lo que HTML5 intenta aportar al desarrollo web, es un lenguaje más actual y más intuitivo a la hora de estructurar una página web, y aportar algunas mejoras como la forma de incrustar videos e imágenes. Las etiquetas han sido cambiadas por otras más representativas y no se abusa tanto de las etiquetas <DIV> para estructurar de forma lógica el código, sino que bastaría con un nombre de etiqueta específico. Es decir, en lugar de estar obligados a crear una división con el nombre como identificador, ahora bastaría con etiquetas específicas que hace que la estructura de la web sea más coherente y fácil de entender por otras personas y los navegadores podrán darle más importancia a según qué secciones de la web facilitándole además la tarea a los buscadores.

Etiquetas como <Font> y <center> que ya no sirven en HTML5 y pasan a definirse exclusivamente con las Hojas de Estilo o CSS [W3C, 2008 a]. Otras como <b> que se siguen soportando pero que ya no se usan como antes.

Se ha orientado también para un mejor acceso multimedia. Para esto se ha creado la etiqueta <video>, que permite la reproducción de vídeo sin la necesidad de instalar ningún plugings, ya sea QuickTime, RealPlayer o el más extendido Flash [Adobe, 2009], aunque de momento se está buscando un códec para que esto llegue a hacerse realidad. De momento sí puede usar Flash y cualquier otro plug-in o códec para reproducir videos, pero el objetivo de crear una etiqueta <video> es unificar todos esos códec en uno sólo.

Por supuesto no va a ser fácil, porque detrás hay muchos intereses económicos interesados en que el códec más usado sea el de una determinada empresa que quiera hacerse con todo el mercado de reproducción de video. Lo que plantea HTML5 es que la reproducción de vídeo sea accesible para todo el mundo, y sobretodo cómodo al no necesitar de ninguna aplicación externa para poder funcionar. Por otro lado, esta cuestión llevaría a que el estándar HTML5 dé soporte para que se pueda cumplir con el estándar UAAG [W3C UAAG, 2005] para requisitos de accesibilidad en el agente de usuario web.

Lo más novedoso que introduce este nuevo estándar de HTML, es la posibilidad de la creación de dibujos o animaciones de cualquier tipo al servicio de la imaginación de los desarrolladores, gracias a la etiqueta <canvas>, que literalmente significa “lienzo”, que abre un nuevo mundo de posibilidades a lo que a diseño gráfico de sitios web se refiere. La ventaja que esto conlleva es que no se necesita ningún tipo de plug-in para el uso de esta característica de HTML5, pero sí que el navegador sea totalmente compatible con este nuevo estándar. Sin embargo, puede producir problemas de accesibilidad al ser requerido un texto alternativo a este elemento gráfico.

En cuando a la aceptación que tienen los navegadores con HTML5 hay para todos los gustos. Hay navegadores que soportan todas sus capacidades y hay otros que no aceptan ninguna, pero es cuestión de tiempo que se acabe soportando esta nueva versión. A modo de prueba, hay una página<sup>2</sup> que comprueba la compatibilidad del navegador con HTML5 y saca una puntuación orientativa en medida de lo que soporte.

De los navegadores más usados y conocidos, Google con su navegador Chrome y Safari, soporta la gran mayoría de lo ofrecido por HTML5, seguido por la última versión de Firefox, que a pesar de ser uno de los navegadores más utilizados, no alcanza el nivel de Chrome y Safari por poco. Le sigue Opera que en su última versión 10.5 subsana las carencias del 10 acercándose un poco más a Firefox en el S.O. Windows, ya en Mac todavía se queda en la anterior versión.

Y en el otro lado tenemos a Internet Explorer de Microsoft que todavía en ninguna de sus versiones ni en ningún S.O., soporta HTML5. El navegador Internet Explorer representa una gran cuota de mercado en lo que al uso de navegadores en Internet se refiere, ya que viene instalado con el S.O. Windows de serie, lo que significa que puede que sea uno de los navegadores más utilizados, frente a sus competidores Firefox o Chrome, aunque han ido mermando su dominio en este sector, pero aún representa una gran cuota de mercado. En su última versión Internet Explorer 8, aún no soporta esta nueva versión de HTML pero ha sido

anunciado que la siguiente versión 9 sí lo soportará, lo que será la aceptación por parte de una de las empresas más importantes del mundo como es Microsoft, del nuevo estándar HTML5 y por tanto su consolidación.

---

<sup>2</sup> [www.html5test.com](http://www.html5test.com)

- Para más información, el estado de aceptación de los navegadores se describe con más profundidad en el apartado 3.3.

### 2.2.1. Fragmentación entre plataformas: ¿Flash o HTML5?

Últimamente se está hablando mucho sobre si HTML5 va a ser una alternativa real a Flash en el campo del vídeo. Actualmente Flash está en prácticamente en todo elemento relacionado con Internet, ya sea multimedia(videos, animaciones, etc..) como juegos online, y presente en gran parte de los ordenadores conectados a internet, ya que es necesario para reproducir la gran mayoría de elementos multimedia y además lleva mucho más tiempo disponible que HTML5. Por otro lado, una de las empresas más en auge de los últimos años, Apple [Apple, 2011 a], ha declarado continuamente que no dará soporte a Flash en su sistema operativo iOS, dando su apoyo a HTML5 considerándolo como el futuro del vídeo en Internet. El gran problema en este sector es el códec, ya que al estar actualmente en borrador, el nuevo estándar HTML5 no estará listo en unos años, por lo que habrá que esperar para resolver este problema, mientras algunas empresas declaran su apoyo a uno o a otro decodificador-codificador de vídeo. De momento la resolución a esta batalla por el control del vídeo en Internet es incierta, ya que el proceso de sustituir a una tecnología con unas capacidades tan amplias, y además tan aceptada y extendida en el los ordenadores del mundo, conlleva un tiempo de adaptación a un nuevo estándar.

La portabilidad es una de las características que muchos desarrolladores buscan. La de programar una aplicación en un lenguaje en una plataforma y sirva para cualquiera. Esto es lo que HTML5 propone, y aunque está lejos de cumplirse, este nuevo estándar se acerca bastante a lo que los desarrolladores desean.

En los últimos meses se anunció el lanzamiento, todavía en fase de prueba experimental, de una herramienta innovadora por parte de Adobe basada en tecnología AIR con el fin de conseguir compatibilidad tanto en sistemas Windows como Mac OS X. Esta herramienta se llama Wallaby [Adobe, 2011], y permite al desarrollador la conversión de Flash a HTML5 de una forma muy sencilla gracias a una interfaz intuitiva. Con esta herramienta se permite el soporte de algunos dispositivos que antes no permitía la tecnología de Adobe como son los sistemas operativos iOS de Apple y con el fin de cumplir con todos los estándares.

## 2.3. Algunas características que aporta HTML5

El nuevo estándar HTML5 aporta nuevas funcionalidades, que quizá no estén disponibles actualmente pero sí se han dado los primeros pasos para incluir las siguientes nuevas características a la Web.

### 2.3.1. Web Semántica

La Web Semántica es una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de metadatos semánticos y ontológicos en una

infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante [W3C, 2010 a].

En esta línea, HTML5 incluye elementos que permiten añadir semántica de la página web más precisa para los buscadores, para que estos la comparen con la que el usuario quiere encontrar y así se puedan obtener resultados más relevantes.

### 2.3.2. Geolocalización

Una de las novedades que ofrece HTML5 son las posibilidades de geolocalización a través de un punto de acceso a internet, más comúnmente conocido como Wi-Fi, en los que su uso no está bien definido pero que con el tiempo se podrá implementar en diversos servicios en redes sociales tan de moda últimamente, o para buscar negocios rápidamente sabiendo nuestra posición física. Esta funcionalidad se implementa mediante APIs que interactúan con el navegador obteniendo su IP y a través de tecnologías como JavaScript conseguir la posición geológica del usuario. Esto acarrea problemas de seguridad y privacidad, ya que enviar la posición física de un usuario no suele ser muy seguro, pero esta funcionalidad solo se podrá aprovechar si dicho usuario da su permiso expreso para que la API pueda calcular su localización geológica.

### 2.3.3. Aplicaciones offline

Uno de los conceptos que renueva HTML5 es la posibilidad de trabajar con aplicaciones web pero de manera offline. Lo que esta nueva funcionalidad permite es la de acceder a aplicaciones offline, sin conexión a internet, siendo necesario previamente conectarse por primera vez a dicha aplicación o página y el navegador descargará los ficheros necesarios de la página a la que se ha accedido, abriendo la posibilidad de poder volver a acceder a esta aplicación o página web sin la necesidad de conexión a Internet.

Si bien ya en HTML4.01 hay páginas que pueden visualizarse sin estar conectados a Internet (si previamente fue descargada), el uso de aplicaciones web offline se hace mucho más complejo. HTML5 plantea una revolución en este tema. Google, el gigante de los buscadores y de las aplicaciones en línea es una de las empresas que apuesta por HTML5 y ha mostrado especial atención en esta característica, para poder brindar a los usuarios que eligen sus servicios, nuevas opciones para trabajo offline, que irán perfeccionándose a medida de que HTML5 logre mayor aceptación y compatibilidad con los navegadores.

## 2.4. Nivel de implementación en los navegadores de HTML5

Las Tablas 1 y 2 informan al desarrollador de los elementos y atributos presentes en dichas tablas respectivamente, que están disponibles en la nueva versión HTML5 y soportados por los distintos agentes de usuario web o navegadores. Estas tablas se han elaborado a partir de la información proporcionada en la página<sup>3</sup>, la cual se actualiza mensualmente o cada vez que sale una nueva versión de los navegadores de los aquí mencionados. .

Después de testear la compatibilidad con los navegadores principales y sus versiones más recientes, quedaba por comprobar esta compatibilidad en versiones Beta, no completas pero sí avanzadas de estos navegadores, y los elegidos fueron Mozilla Firefox 4.0 Beta 7 e Internet Explorer 9.0 Beta. El resultado de comprobar la compatibilidad con HTML5 en estas nuevas versiones Beta de los navegadores más populares fue excelente.

Las anteriores versiones aceptaban muy poco o nada de los nuevos elementos de HTML5, por lo que dan un cambio de rumbo importante, en cuanto a la aceptación y expansión de éste nuevo estándar. De momento, como versiones Beta que son, no tienen una compatibilidad completa pero sí conforman un aceptable número de elementos, lo suficiente como para aceptar las etiquetas o funcionalidades más esperadas de HTML5, que son <video> y <audio>.

Lo que es evidente es que HTML5 se va a imponer y es casi obligatorio que los navegadores acaben implementándolo completamente, ya que las ventajas que ofrecen con el anterior son muy significativas.

Lo que los responsables de dicha página quieren dejar claro, es que no intentan persuadir de no usar el nuevo estándar HTML5 para desarrollar páginas web, sino informarle de los posibles problemas que podrá tener a la hora de implementar unos u otros elementos.

Actualmente la versión definitiva de Mozilla Firefox 4.0, ya no es Beta, y proporciona una excelente aceptación de HTML5 siendo prácticamente igual a mejor que su anterior versión. Mientras tanto, Internet Explorer aún no ha lanzado la versión definitiva 9.0, por lo que no se puede comprobar si la aceptación de HTML5 puede rivalizar con Firefox.

## Capítulo 2: Introducción a HTML5

Ésta tabla está actualizada a fecha de 11/03/2011:

Elementos HTML5	Chrome 10	Firefox RC	4.0	IE9 RC	Opera 11	Safari/ Webkit r74232
Elemento <b>article</b>	x	✓		x	x	x
Elemento <b>aside</b>	x	✓		x	x	x
Elemento <b>audio</b>	x	✓ x		✓ x	✓ x	x
Elemento <b>canvas</b>	x	x		✓ x	x	x
Elemento <b>datalist</b>	✓ x	✓ x		–	✓ x	–
Elemento <b>details</b>	–	–		–	–	–
Elemento <b>figcaption</b>	x	✓ x		x	x	x

---

<sup>3</sup> [www.html5accessibility.com](http://www.html5accessibility.com)

## 2.4 Nivel de implementación en los navegadores de HTML5

Elemento <b>figure</b>	x	✓x	x	x	x
Elemento <b>footer</b>	x	✓x	x	x	x
Elemento <b>header</b>	x	✓x	x	x	x
Elemento <b>hgroup</b>	–	–	–	–	–
Input <b>color</b>	–	–	–	✓x	–
Input <b>date</b>	✓x	–	–	x	–
Input <b>date and time</b>	✓x	–	–	x	–
Input <b>local date and time</b>	✓x	–	–	x	–
Input <b>e-mail</b>	–	–	–	x	–
Input <b>month</b>	✓x	–	–	x	–
Input <b>number</b>	✓x	–	–	✓x	–
Input <b>range</b>	✓x	–	–	✓x	x
Input <b>search</b>	✓	✓	✓	x	✓
Input <b>telephone</b>	x	–	–	x	–
Input <b>time</b>	✓x	–	–	✓x	–
Input <b>url</b>	–	–	–	x	–
Input <b>week</b>	✓x	–	–	✓x	–
Menú > context menú	–	–	–	–	–
Menú > list	–	–	–	–	–
Menú > toolbar	–	–	–	–	–
Elemento <b>meter</b>	x	–	–	–	x
Elemento <b>nav</b>	x	✓x	x	x	x
Elemento <b>output</b>	–	–	–	x	–
Elemento <b>progress</b>	x	–	–	x	x
Elemento <b>section</b>	x	✓x	x	x	x
Elemento <b>summary</b>	–	–	–	–	–
Elemento <b>video</b>	x	✓x	✓x	✓x	x

Tabla 1: Índice de implementación de elementos HTML5 por navegadores

Atributos HTML5	Chrome 10	Firefox 4.0 b10	IE9 Beta	Opera 11	Safari/ Webkit r74232
Atributo <b>hidden</b>	–	✓	–	–	✓
Atributo <b>required</b>	–	✓	–	x	–
Atributo <b>placeholder</b>	x	✓x	–	x	✓x

Tabla 2: Índice de implementación de atributos HTML5 por navegadores

Leyenda de Tabla 1 y 2:

- ✘ “no soportado” significa que la característica está implementada por el navegador, pero el soporte de accesibilidad requerido no está implementado.
- ✓ ✘ “parcialmente soportado”, significa que la característica está implementada pero el soporte de accesibilidad está parcialmente implementado.
- — “no implementado”, significa que la característica todavía no está implementada por el navegador.
- ✓ “soportado”, significa que el navegador expone información de la característica a través de un acceso a la API y si la función es interactiva, puede funcionar con una amplia gama de tipos de dispositivos de entrada.

### 2.5. Especificación borrador HTML5.0

Esta nueva versión de HTML no es un lenguaje totalmente nuevo, sino que utiliza muchos elementos y conserva prácticamente idéntica la estructura de su anterior versión, pero con cambios sustanciales que le hacen ser más intuitivo y fácil a la hora de estructurar una página web.

Para elaborar un documento en HTML5 lo primero que se debe situar al principio es un DOCTYPE nuevo. Es muy similar a la anterior versión 4/4.01, pero esta vez es infinitamente más simple, bastaría con el código `<!DOCTYPE html>` siendo indiferente si se escribe en mayúsculas o minúsculas, lo que ahorra la memorización del DOCTYPE antiguo o tener que abrir un documento previo y copiarlo, y la gran ventaja es que es compatible con las anteriores versiones de HTML y XHTML, con ello se simplifica muchísimo la posible inclusión de estos estándares.

Antes de HTML5, un DOCTYPE que activa el modo estándar en todos los navegadores modernos lo podemos ver en la figura 1:

```
<!DOCTYPE html
PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
```

*Figura 1: Doctype HTML4.01*

Con HTML5 podemos conservar el anterior, pero seguramente que casi siempre se optará por el siguiente, ya que la simplicidad y la comodidad con el anterior es evidente, como muestra la figura 2:

```
<!DOCTYPE html>
```

*Figura 2: Doctype HTML5*

El elemento raíz de todo documento es la etiqueta `<html>` y ha venido acumulando elementos redundantes, quedando así en XHTML y mostrado en la figura 3:

```
<html xmlns="http://www.w3.org/1999/xhtml"
lang="en"
xml:lang="en">
```

Figura 3: Elemento raíz HTML4.01

- Pero ahora todos los elementos de HTML5 están en el mismo *namespace* y no hace falta repetir el idioma, mostrado en la figura 4:

```
<html lang="en">
```

Figura 4: Elemento raíz HTML5

### 2.5.1. Nuevas etiquetas

Como se ha venido introduciendo anteriormente una de las interesantes aportaciones es la facilidad al diseñar la estructura de un sitio web. Lo que hace esto más sencillo, es evitar el abuso de las etiquetas <DIV> para estructurar de forma lógica el contenido, y en su lugar se han creado etiquetas específicas para cada situación. [W3C, 2011], [W3Schools, 2010 d].

Las nuevas etiquetas son las siguientes:

- **<article>**: representa un componente de la página o sitio web, con la intención de que pueda ser reutilizado y repetido. Sería un elemento perfecto para foros o blogs que cada día se añada información nueva o noticias.
- **<section>**: representa una sección genérica de un documento o una aplicación. Podría agrupar contenido de un tema específico dentro de un artículo en que se pueda introducir, y que tenga sentido.
- **<aside>**: representa una sección de la página que abarca un contenido tangencialmente relacionado con el contenido que lo rodea, por lo que se le puede considerar un contenido independiente.
- **<header>**: representa un grupo de artículos introductorios o de navegación. Normalmente se relaciona con el encabezado a una sección de la página, mediante la inclusión de etiquetas de título (h1-h6).
- **<nav>**: representa una sección de una página que es un link a otras páginas o a partes dentro de la página: una sección con links de navegación.
- **<footer>**: representa el pie de una sección, con información acerca de la página/sección que poco tiene que ver con el contenido de la página, como el autor, el copyright o el año.

En la figura 5, podemos ver gráficamente cómo quedarían estas nuevas etiquetas en una página web esquemáticamente, comparándolo con la versión de HTML anterior y HTML5.

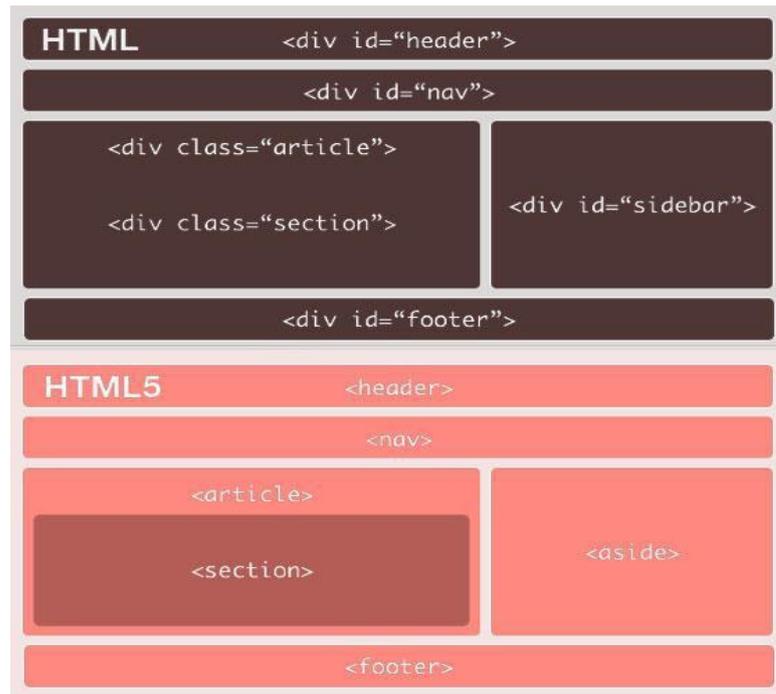


Figura 5: Esquema de etiquetas HTML5

Otras etiquetas nuevas:

- **<hgroup>**: representa el encabezamiento de una sección. Se usa para agrupar los elementos h1-h6 cuando el encabezamiento tiene múltiples niveles, como subencabezados o títulos alternativos.
- **<time>**: permite introducir la hora y fecha cuando se modifica o introduce información en la web, lo que permite saber al usuario cuando ha sido creado o modificado algo, y es muy útil a la hora de buscar ya que es más fácil encontrar lo más nuevo o lo más antiguo.
- **<progress>**: lo usaremos cuando queramos representar el estado de cierto proceso, muy útil para cargas y descargas de archivos.
- **<meter>**: se utilizará para indicar ciertas medidas dependiendo de los atributos, por ejemplo el tamaño del disco usado.
- **<mark>**: se usa para indicar importancia al texto, o cuando queramos atraer la atención del usuario como si subrayamos un texto con un rotulador fluorescente.
- **<video>**: es una de las etiquetas más nuevas más esperadas y más novedosas de este nuevo estándar. Permite introducir vídeo en el sitio web sin la necesidad de instalar ningún otro códec o aplicación externa. Se explicará con más detalle más adelante.
- **<audio>**: permite introducir pistas de audio con más facilidad que en la versión anterior de HTML.
- **<canvas>**: una de las grandes novedades junto con la etiqueta `<video>` que permite “dibujar” directamente en la pantalla del navegador, dando libertad tanto a los desarrolladores como a los usuarios en crear todo tipo de dibujos o decoraciones para su página web.
- **<details>**: permite la posibilidad de dar información adicional sobre algún elemento si el usuario lo desea
- **<dialog>**: lo usaremos para representar una conversación entre varias personas.

- **<figure>**: se utilizará para representar una imagen, o contenido relacionado entre sí o grupo de elementos (texto o imágenes).
- **<legend>**: es un elemento que se puede usar dentro de <figure>, que puede estar o no, y sirve como leyenda del contenido de su elemento padre.
- **<fieldset>**: sirve para dibujar un rectángulo y agrupar el contenido dentro de ella. En este caso, si introducimos el elemento <legend> dentro de este otro elemento, podremos escribir el título de este contenido en el mismo rectángulo.
- **<ruby>**: define anotaciones en letras y caracteres chinos.
- **<rt>**: se usa dentro de la etiqueta <ruby>. Define una explicación o pronunciación de caracteres chinos.
- **<rp>**: es similar a <rt> pero define lo que los navegadores tienen que enseñar si no soportan la etiqueta <ruby>.
- **<keygen>**: este elemento básicamente lo que hace es generar una clave privada y una pública para que el navegador y el servidor puedan comunicarse.
- **<summary>**: define el encabezamiento de un elemento <details>.
- **<wbr>**: sirve para evitar que el navegador no rompa palabras que puedan ser conflictivas o si son demasiado largas.

En el apartado de formularios, se han añadido nuevos elementos que son permitidos en el campo “type” por lo que, ahora los datos que se introduzcan en cada formulario serán más específicos y más reconocibles que antes. [W3Schools, 2010 c]

Los nuevos elementos son los siguientes:

- **<input type=“search”>**: para cajas de búsqueda.
- **<input type=“number”>**: para sumar o restar números.
- **<input type=“range”>**: para seleccionar un valor entre dos valores predeterminados.
- **<input type=“color”>**: seleccionar un color.
- **<input type=“tel”>**: útil para números telefónicos.
- **<input type=“url”>**: útil para direcciones web.
- **<input type=“email”>**: para direcciones de email.
- **<input type=“date”>**: para seleccionar un día del calendario.
- **<input type=“month”>**: meses.
- **<input type=“week”>**: semanas.
- **<input type=“time”>**: fechas.
- **<input type=“datetime”>**: para una fecha exacta, absoluta y tiempo.
- **<input type=“datetime-local”>**: para fechas locales y frecuencia.

Estos nuevos elementos vienen a sustituir al clásico “text” que poníamos en type para todo o casi todo que quisiésemos poner en un formulario. Lo que hace esto es mejorar la búsqueda a la hora de rellenar un campo, por ejemplo, si estamos rellenando un campo de tipo url, al empezar a rellenar nos saldrá una lista de las url que ya hemos visitado y así facilitar al buscador desde qué campo selecciona la lista.

### 2.5.2. Nuevos atributos:

Se han introducido nuevos atributos y características con los que dotar a los elementos del HTML5, algunos de ellos orientados a la accesibilidad:

- **Ping:** Este atributo contiene una lista de URLs, las cuáles serán *llamadas* cuando un usuario haga click en ese enlace, dentro del elemento `a`. Por ejemplo, un uso práctico sería para estadísticas.
- **Target:** Especifica el destino donde se va a abrir un enlace distinto a la ventana actual, y aplicable en `a`, `area`, `base`, `form` y `link`.
- **Autofocus:** Destinado para indicar el elemento `input />` (no `hidden`), `select />`, `textarea />` o `button />` que ha de coger el foco al cargar la página. Este atributo permite una mayor accesibilidad, ya que da la posibilidad de que cuando se carga la página, el foco se sitúe en un determinado elemento, por ejemplo, en la barra de búsqueda principal en una página de búsquedas en la Web.
- **Form:** nuevo atributo relacionado con los elementos de formularios. Indica a qué formulario pertenece un elemento y permite poner dicho elemento en cualquier parte de una página.
- **Replace:** atributo para `input />`, `button />` y `form />` que le afectará cuando el contenido del elemento sufra algún cambio.
- **Data:** Para `form />`, `select />` y `datalist />`.
- **Required:** Para elementos `input />` (Excepto `hidden` e `image`) y `textarea />`, indica que el campo es obligatorio. Muy útil para indicar sin Javascript, que un campo es obligatorio a la hora de rellenar un formulario, y aplicando una mejor accesibilidad en dichos formularios.
- **Disabled:** Para `fieldset />`, permite desactivar el `fieldset` por completo, es decir, todos los hijos de dicho `fieldset` son desactivados.
- **Autocomplete, min, max, pattern, step:** Para elementos `input />` permite delimitar las posibilidades de nuestros elementos de entrada. Estos atributos permiten una mayor accesibilidad, indicando al usuario el límite de información o datos que puede introducir en dichos elementos.
- **Scoped:** Para elemento `style />`, permitirá usar hojas de estilo “scoped”, es decir, si añadimos este atributo en un elemento “style”, podremos aplicar solo un determinado estilo a un subárbol del documento, o lo que es lo mismo, a ese elemento y a sus hijos.
- **Async:** Para el elemento `script />` el `ajax` hecho atributo. Con este atributo especificamos que el código interno se puede ejecutar en cualquier momento de la página, mejorando la velocidad de carga.
- **Hidden:** define la propiedad de que un elemento sea invisible pero ocupa el espacio que le correspondería.
- **Spellcheck** (corrector ortográfico): indica si el contenido de un elemento debe ser pasado por el corrector ortográfico.
- **Subject:** especifica el ítem correspondiente de un elemento.
- **Contenteditable:** indica que se trata de un área editable.
- **Contextmenu:** puede ser usado como punto de menú contextual proporcionado por el usuario.
- **Draggable:** indica que se trata de un elemento “draggable”, es decir, que el usuario pueda arrastrar un elemento. Útil a la hora de indicar elementos interactivos por parte del usuario.
- **TabIndex:** indica la posición numérica a la que llegaremos pulsando la tecla TAB. Aporta una gran accesibilidad a las páginas web para navegar por las mismas a través del tabulador. Permite al desarrollador indicar el orden que quiere que siga la

navegación al pulsar la tecla TAB, muy útil para personas con dificultades para la navegación o simplemente para hacer más accesible el contenido de una página web.

### 2.5.3. Elementos cambiados:

En este apartado se incluyen aquellos elementos de HTML5 que son incompatibles con HTML4:

- El elemento `<a />` sin href ahora creará un enlace al sitio.
- El elemento `<address />` es ahora un nuevo concepto de sección.
- El elemento `<b />` ahora representa un trozo de texto a ser estilizado sin ninguna importancia.
- Para elementos `<label />` el navegador no debe mover el foco desde la etiqueta al control a menos que el comportamiento sea estándar para el interfaz utilizado en la plataforma.
- `<menu />` ha sido redefinido para ser usado con los actuales menús.
- El elemento `<small />` ahora representa una impresión pequeña.
- El elemento `<strong />` definitivamente representa el énfasis puesto en trozo de nuestro texto.

### 2.5.4. Elementos eliminados:

En la nueva versión, algunos de los elementos anteriormente desaprovechados pasan a ser eliminados definitivamente.

- acronym
- applet
- basefont
- big
- center
- dir
- font
- frame
- frameset
- isindex
- noframes
- noscript (solo en XHTML5)
- s
- strike
- tt
- u

### 2.5.5. Atributos eliminados:

A continuación se muestran los atributos que ya no se pueden usar en el nuevo estándar:

- rev y charset en <link /> y <a />
- target en <link />
- nohref en <area />
- profile en <head />
- version en <html />
- name en <map />
- scheme en <meta />
- archive, classid, codetype, declare y standby en <object />
- valuetype en <param />
- charset en <script />
- summary en <table />
- header, axis y abbr en <td /> y <th />

### 2.5.6. Nuevos eventos

En HTML4 se añadió la posibilidad que se activen una serie de acciones en un navegador, como cuando un usuario hace clic en un elemento mediante JavaScript.

A continuación se presentan los nuevos eventos de atributos que se pueden insertar en HTML5 para definir acciones de sucesos [W3Schools, 2010 b].

### 2.5.7. Eventos de ventana

Eventos relacionados con acciones que se realicen en la ventana de los navegadores:

- **Onafterprint:** script de que se ejecute después de que el documento se imprima.
- **Onbeforeprint:** script de que se ejecute antes de que el documento se imprima.
- **Onbeforeunload:** Secuencia de comandos para ejecutar antes que se cargue el documento
- **OnError:** Script para ejecutarse cuando se produzca un error.
- **Onhaschange:** Script para ejecutarse cuando el documento haya cambiado.
- **OnMessage:** Script para ejecutarse cuando el mensaje se active.
- **Offline:** Script para ejecutarse cuando el documento esté desconectado.
- **Online:** Script para ejecutarse cuando el documento se ponga en línea.
- **Onpagehide:** Secuencia de comandos que se ejecutan cuando la ventana esté oculta.
- **Onpageshow:** Secuencia de comandos que se ejecutan cuando la ventana se haga visible.
- **Onpopstate:** Secuencia de comandos que se ejecutan cada vez que la entrada al historial cambia.
- **Onredo:** Script para ejecutarse cuando el documento realiza una “rehacer”.
- **Onresize:** Secuencia de comandos que se ejecutan cuando la ventana cambia de tamaño.

## 2.6 HTML5 y elementos relacionados con el contenido multimedia

- **Onstorage:** Script para ejecutarse cuando se carga un documento.
- **Onundo:** Script para ejecutarse cuando un documento realiza una “deshacer”.
- **Onunload:** Script para ejecutarse cuando el usuario abandona el documento.

### 2.5.8. Eventos de formulario

Eventos relacionados con elementos de formularios en los que el usuario tenga que introducir algún tipo de información en un cuadro de texto o similar:

- **Oncontextmenu:** Script que se ejecute cuando un menú contextual se activa.
- **Onformchange:** Script para ejecutarse cuando una cambio de forma.
- **Onforminput:** Script para ejecutarse cuando un formulario de entrada del usuario se ejecuta.
- **Oninput:** Script para ejecutarse cuando un elemento recibe la entrada del usuario.
- **Oninvalid:** Script para ejecutarse cuando un elemento no es válido.

### 2.5.9. Eventos del ratón

Eventos activados por un ratón o acciones similares de usuario. Aplicable a todos los elementos de HTML5 [W3Schools, 2010 a].

- **Ondrag:** Script para ejecutarse cuando un elemento se arrastra.
- **Ondragend:** Secuencia de comandos que se ejecutan al final de una operación de arrastre.
- **OnDragEnter:** Script para ejecutarse cuando un elemento ha sido arrastrado a un destino válido.
- **OnDragLeave:** Script para ejecutarse cuando un elemento deja un destino válido.
- **OnDragOver:** Script para ejecutarse cuando un elemento se arrastra sobre un destino válido.
- **OnDragStart:** Script se ejecute en el inicio de una operación de arrastre.
- **OnDrop:** Script para ejecutar cuando el elemento arrastrado se cayó.
- **OnMouseWheel:** Secuencia de comandos que se ejecutan cuando la rueda del ratón se está moviendo.
- **OnScroll:** Secuencia de comandos que se ejecutan cuando la barra de desplazamiento de un elemento se desplaza.

### 2.5.10. Eventos multimedia

Eventos activados en elementos multimedia como <video> y <audio>, para un mayor control por parte del desarrollador web a la hora de implementar estas nuevas funcionalidades:

Aplicable a todos los elementos de HTML5, pero es más común en los elementos multimedia [W3Schools, 2010 a].

- **oncanplay:** Secuencia de comandos que se ejecutan cuando los elementos multimedia se puedan iniciar, pero puede parar para un almacenamiento temporal.
- **Oncanplaythrough:** Secuencia de comandos que se ejecutan cuando los elementos multimedia se puedan reproducir hasta el final, sin parar para un almacenamiento

## Capítulo 2: Introducción a HTML5 temporal

- **Ondurationchange:** Secuencia de comandos que se ejecutan cuando la longitud de los elementos multimedia cambian.
- **Onemptied:** Script para ejecutarse cuando un recurso de un elemento multimedia de repente se desconecta (errores de red, errores de carga, etc...)
- **Onended:** Secuencia de comandos que se ejecutan cuando los elementos multimedia llegan al final.
- **Onerror:** Secuencia de comandos que se ejecutan cuando se produce un error durante la carga de un elemento.
- **Onloadeddata:** Script para ejecutarse cuando los datos se han cargado
- **Onloadedmetadata:** Secuencia de comandos que se ejecutan cuando la duración y los datos de otros elementos multimedia se cargan.
- **onLoadStart:** Script para ejecutarse cuando el navegador empieza a cargar los datos de los elementos multimedia.
- **OnPause:** Script para ejecutarse cuando los datos de los elementos multimedia se pausan.
- **onplay:** Script para ejecutarse cuando los datos de los elementos multimedia van a empezar a reproducirse.
- **Onplaying:** Script para ejecutarse cuando los datos de los elementos multimedia han empezado a reproducirse.
- **OnProgress:** Secuencia de comandos que se ejecutan cuando el navegador está obteniendo los datos de los elementos multimedia.
- **Onratechange:** Secuencia de comandos que se ejecutan cuando la tasa de datos de los elementos multimedia de reproducción han cambiado.
- **Onreadystatechange:** Script que se ejecute cuando cambia al estado listo.
- **Onseeked:** Script para ejecutar cuando el elemento de la búsqueda de un atributo de los elementos multimedia ya no son ciertos, y la búsqueda ha terminado.
- **Onseeking:** Script para ejecutar cuando el elemento de la búsqueda de un atributo de los elementos multimedia son ciertos, y la búsqueda ha comenzado.
- **Onstalled:** Secuencia de comandos que se ejecutan cuando se produce un error al ir a buscar datos de los elementos multimedia.
- **Onsuspend:** Script para ejecutarse cuando el navegador ha estado buscando datos de los medios multimedia, pero se detuvo antes de que el archivo de los elementos multimedia fuesen encontrados.
- **Onupdate:** Secuencia de comandos que se ejecutan cuando los elementos multimedia cambia su posición de reproducción.
- **Onvolumechange:** Secuencia de comandos que se ejecutan cuando se cambia el volumen de los elementos multimedia, también cuando el volumen esté ajustado en "silencio".
- **Onwaiting:** Secuencia de comandos que se ejecutan cuando los elementos multimedia han dejado de reproducirse, pero se espera que reanude.

### 2.5.11. Extensión de HTMLDocument

HTML5 también ha modificado el elemento padre de *Document Object Model* (DOM) Level 2. En él encontramos una serie de mejoras y otras que finalmente se hacen estándares, es decir, durante el tiempo que éste nuevo estándar se mantenga como borrador, estará en continuo cambio y se incluirán nuevas modificaciones, por lo que algunas serán imprescindibles para el correcto uso de HTML5 debido a su importancia y además también serán aplicables a través de JavaScript.

- **getElementsByClassName()**, para seleccionar elementos por el atributo class. Ya lo comentamos hace tiempo y vimos que las diferencias a nivel de tiempo de respuesta eran más que satisfactorias.
- **innerHTML**, aunque prácticamente se usa en todas, o casi todas, las aplicaciones web existentes, por fin será reconocido como estándar en la especificación. Además aprovechando su inserción se posibilita su uso en el elemento padre.
- **activeElement**, **hasFocus()**, nos permitirá conocer el elemento activo en tiempo real y el que tenga el foco.
- **getSelection()**, devuelve un objeto con la selección actual.
- **designMode** y **execCommand()**, muy usados para editar documentos.

### 2.5.12. Extensiones de Elementos HTMLElement.

A nivel de elementos de HTML5, el DOM también ha sufrido una serie de cambios:

- **getElementsByClassName()**, nos permite seleccionar los hijos de cualquier objeto que contengan una clase determinada.
- **innerHTML**, nos permite leer/modificar el contenido de un nodo (al añadir crea nodos texto con etiquetas).
- **classList**, una implementación muy interesante para vivir con className que nos permite interactuar con las clases de los elementos, proporcionando métodos como **has()**, **add()**, **remove()** y **toggle()** con los que podremos trabajar con las clases de nuestros elementos.
- **relList**, funciona de igual forma que **classList**, pero sobre los atributos rel de `<a />`, `<area />` y `<link />`.

## 2.6. HTML5 y elementos relacionados con el contenido multimedia

HTML5 introduce importantes elementos relacionados con la multimedia, foco de este proyecto, por ello que se dedique este apartado a profundizar en estos elementos.

Hasta ahora, había que escribir varias líneas de código para introducir un vídeo o un audio, usar otro lenguaje o incluso incorporar otras tecnologías para incluir elementos multimedia en las páginas web. Pero con la llegada de este nuevo estándar, se quiere simplificar muchísimo la forma de implementar estos contenidos, ya que lo que se busca es la simplicidad

## Capítulo 2: Introducción a HTML5

obteniendo prácticamente el mismo resultado. Sin embargo, se siguen encontrando algunos inconvenientes, como por ejemplo, el formato que se utilizará en los videos, y que a continuación se explicará.

### 2.6.1. Etiqueta <video>

Ésta es la etiqueta que más está dando y dará que hablar, ya que va a suponer un antes y un después en la forma de mostrar páginas web en Internet. El elemento que se marca con la etiqueta o *tag* <video> es sin duda el atractivo de este nuevo estándar y es que su llegada puede revolucionar el mundo multimedia en Internet.

Hasta ahora, nunca ha habido un estándar para mostrar videos en páginas web, por una parte porque en HTML4 no había ninguna etiqueta que englobara a todos y al haber muchos tipos de códec de vídeo diferente, cada vez se hacía con el que mejor convenía, ya sea con los codecs de QuickTime, Windows Media, Real Player, Flash u otros.. y por otra parte, porque actualmente los videos son contenido de tipo esencial en Internet, algo que antes no sucedía.

Actualmente, la mayoría de los videos son mostrados a través de la instalación de un plug-in externo, como uno de los más extendidos, el de tecnología Flash, sin embargo no todos los navegadores ni todos los reproductores usan el mismo códec.

HTML5 propone un estándar para todos los videos que se incrusten en la Web, lo cual no va a ser sencillo, ya que imponer un códec para todos y desde los actores que desarrollan los reproductores se antoja algo imposible. De cualquier manera, aún queda tiempo para que el nuevo estándar se consolide y se pueda disfrutar de todas las mejoras de esta nuevo elemento.

Actualmente, hay tres formatos de vídeo disponibles para el elemento <video>:

- Ogg<sup>4</sup>: archivos Ogg con códec de vídeo Theodora y códec de audio Vorbis.
- MPEG4<sup>5</sup>: archivos MPEG4 con códec de vídeo H.264 y códec de audio ACC.
- WebM<sup>6</sup>: archivos WebM con códec de vídeo VP8 y códec de audio Vorbis.

Sin dudas las mejoras son evidentes respecto a Flash comparado con H.264:

- La calidad nativa del códec H.264 es mucho mejor que Flash, que al final y al cabo, no fue inventado para reproducción de videos.
- El uso de procesador requerido para ver videos usando H.264 es notablemente inferior a ver el mismo vídeo usando el plug-in de Flash<sup>7</sup>.

En la siguiente tabla (ver tabla 3) se muestra la relación de varios navegadores con dos de los formatos de vídeo que se emplean en el momento [W3Schools, 2010 e]. Como se puede observar el formato Ogg es el que tiene más aceptación por parte de los navegadores. Esto se debe a que el formato Ogg es libre tanto como códec como contenedor, y tiene la ventaja de que se si usara este para cualquier navegador ya no sería necesario ningún software adicional, pero esto ya es decisión de las principales empresas de Internet como Adobe.

Por otro lado, el formato WebM ya se está usando de forma masiva en YouTube por parte de Google, que ya ha comenzado la conversión de la mayoría de los videos alojados en los servidores de la famosa página de visualización de videos online.

Formato	Firefox	IE8	Firefox 3.5	Opera 10.5	Opera 3.0	Chrome 3.0	Safari
---------	---------	-----	-------------	------------	-----------	------------	--------

## 2.6 HTML5 y elementos relacionados con el contenido multimedia

Ogg	No	3.5+	10.5+	5.0+	No
MPEG4	No	No	No	5.0+	3.0+
WebM	No	No	10.6+	6.0+	No

Tabla 3: Formatos soportados en <video>

---

<sup>4</sup> <http://www.vorbis.com/>

<sup>5</sup> <http://www.m4f.org/mpeg4/>

<sup>6</sup> <http://www.webmproject.org/>

<sup>7</sup> <http://www.adobe.com/es/>

### **<Video> con HTML4**

La forma de visualizar un vídeo en una página web, actualmente es similar a la que se muestra en la figura 6. Su implementación incluye el elemento <object> del estándar HTML. Sin embargo, el elemento <object> tiene el problema de que hay navegadores que no lo interpretan bien, y por ello, es frecuente verlo implementado a través del elemento <embed>. La etiqueta <embed> no pertenece a ningún estándar, sino que fue un invento de Netscape que se generalizó en el momento de su creación. Su utilización significa ir contra los estándares de compatibilidad, aunque resulta de facto compatible con todos los navegadores ya que la práctica totalidad de los navegadores es capaz de interpretarla correctamente. Si tratamos de validar una página que contenga una etiqueta <embed> el resultado será negativo ya que lo propuesto en el estándar es el uso de la etiqueta <object>, pero eliminar la etiqueta <embed> puede suponer que el código deje de funcionar en los navegadores [Moreno, L. et al., 2008 b].

```
<object classid="clsid:d27cdb6e-ae6d-11cf-96b8-444553540000"
width="425" height="344"
codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,40,0">
<param name="allowFullScreen" value="true" />
<param name="allowscriptaccess" value="always" />
<param name="src"
value="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&" />
<param name="allowfullscreen" value="true" />
<embed type="application/x-shockwave-flash" width="425"
height="344"
src="http://www.youtube.com/v/oHg5SJYRHA0&hl=en&fs=1&"
allowscriptaccess="always" allowfullscreen="true">
```

*Figura 6: Video en HTML4*

### **<Video> con HTML5**

```
<video width="640" height="360"
src="http://www.youtube.com/demo/google_main.mp4" controls
autobuffer><p> Try this page in Safari 4! Or you can <a
href="http://www.youtube.com/demo/google_main.mp4">download
the video</a> instead.</p>
</video>
```

*Figura 7: Video avanzado en HTML5*

Lo mostrado en la figura 7 sería un ejemplo más completo de cómo mostrar un vídeo en HTML5, pero si lo queremos simplificar, todo lo que necesitamos para mostrarlo está en la figura 8:

```
<video src="movie.ogg" controls="controls">
</video>
```

*Figura 8: Video simplificado en HTML5*

El atributo “controls” proporciona los controles nativos que ofrece HTML5 para la etiqueta video, los cuales se tendrán en cuenta pero no serán relevantes para este proyecto. Estos

controles ofrecen los botones de play, pause y control de volumen junto con una barra de progreso. Para no ocasionar problemas con el visionado de un video, siempre es recomendable añadir atributos de altura y ancho en píxeles.

Para navegadores que todavía no soporten la etiqueta <video>, bastaría con introducir texto entre estas etiquetas, por lo que el navegador ignoraría el contenido multimedia y mostraría el texto que queramos en su lugar.

El elemento vídeo permite añadir múltiples elementos, y con éstos elementos direccionar a distintos archivos de video. El navegador usará el primer formato reconocido de entre los formatos que le introduzcamos en la etiqueta <source> por lo que usaría el que aceptase y el resto los ignoraría, como se muestra en la figura 9:

```
<video width="320" height="240" controls="controls">
  <source src="movie.ogg" type="video/ogg" />
  <source src="movie.mp4" type="video/mp4" />
  Your browser does not support the video tag.
</video>
```

Figura 9: Video múltiples formatos HTML5

### ***Atributos para la etiqueta <video>***

Estos nuevos atributos de la etiqueta <video>, se pueden incluir después de dicha etiqueta directamente, como aparece en la figura 9, para realizar lo siguiente [W3Schools, 2010 e]:

- **Autoplay:** Especifica que el vídeo comenzará a reproducirse tan pronto como esté listo
- **Controles:** Especifica que los controles se mostrará como un botón de reproducción.
- **Height(píxeles):** Especifica la altura del reproductor de vídeo
- **Loop:** Especifica que el archivo multimedia se iniciará de nuevo, una vez que haya terminado.
- **Preload:** Especifica que el vídeo se cargará en la carga de la página, y listo para reproducir. Se ignora si "autoplay" está activado.
- **Src:** Especifica la dirección URL del vídeo a reproducir.
- **Width(píxeles):** Especifica el ancho del reproductor de vídeo.

### 2.6.2. Etiqueta <audio>

Ésta etiqueta también es una de las nuevas de este nuevo estándar y tiene prácticamente las mismas propiedades que su "hermano" <video>, con sus correspondientes problemas de compatibilidad de códecs. La etiqueta <audio> viene a hacer lo mismo que <video> con los videos en internet. Ésta propone un estándar para incluir archivos de audio en las páginas web sin la necesidad de instalar ninguna otra aplicación externa, ni usar reproductor en tecnología Flash, para que los usuarios puedan escuchar sonidos, canciones, podcasts,... directamente desde la página web. Como se comentaba anteriormente con <video>, ésta vez HTML5 especifica un estándar para incluir audio, con el elemento <audio> más fácil y cómodo que con HTML4, pero con los inconvenientes de los códecs que puede soportar cada navegador.

Formato	IE8	Firefox 3.5	Opera 10.5	Chrome 3.0	Safari 3.0
Ogg Vorbis	No	Si	Si	Si	No
Mp3	No	No	No	Si	Si
Wav	No	Si	Si	No	Si

Tabla 4: Formatos soportados en <audio>

En la tabla 4 se muestra la relación de varios navegadores con tres de los formatos de audio que se emplean en el momento [W3Schools, 2010 a]. Como se puede observar el formato Ogg y Wav son de los más aceptados por los navegadores, ya que desde un principio se recomendó que se aceptará Ogg como formato estándar en HTML5, sin embargo esto cambió por desacuerdos entre compañías responsables de varios navegadores por lo que se optó, de momento, de que cada uno soporte los que crea conveniente.

#### <Audio> con HTML4

Con el anterior estándar HTML4, como se ve en la figura 10 no era complicado introducir una pista de audio en la web, pero se necesitaba una aplicación externa, en este caso Flash, para poder reproducirlo. En cuanto a la etiqueta, se trataba como un objeto, como la mayoría de los elementos presentes en el código HTML, con el inconveniente que no se sabía que era un archivo de audio hasta que no se abriera el código.

```
<object type="application/x-shockwave-flash"
data="player_mp3.swf" width="200" height="20">
<param name="movie" value="player_mp3.swf" />
<param name="FlashVars"
value="mp3=music.mp3&showstop=1&showinfo=1" />
</object>
```

Figura 10: Audio en HTML4.01

#### <Audio> con HTML5

Al igual que con <video>, para incluir un archivo de audio con HTML5 es sumamente sencillo, tal y como se muestra en la figura 11:

```
<audio src="song.ogg" controls="controls">
</audio>
```

Figura 11: Audio en HTML5

De momento si queremos que nuestro archivo de audio se pueda escuchar en Firefox, Chrome, y Opera deberemos optar por usar archivos en formato Ogg. Si por el contrario queremos que se pueda escuchar en Safari, deberemos incluirlo de tipo Mp3 o Wav

Para resolver el problema de qué tipo incluir para que se escuche en cualquier navegador, optaremos por la misma solución que se menciona con el elemento <video>. Para ello, incluiremos varios enlaces de distinto tipo, y el navegador reproducirá el primer formato reconocido al igual que ocurre con la etiqueta <video>, como muestra la figura 12.

```
<audio width="320" height="240" controls="controls">
<source src="song.ogg" type="audio/ogg" />
<source src="song.mp3" type="audio/mp3" />
Tu navegador no soporta la etiqueta audio.
</audio>
```

Figura 12: Múltiples formatos de audio en HTML5

### **Atributos para la etiqueta <audio>**

Estos nuevos atributos de la etiqueta <audio>, se pueden incluir después de la etiqueta directamente, como aparece en la figura 12, para realizar lo siguiente [W3Schools, 2010 a]:

- **Autoplay:** Especifica que el audio comenzará a reproducirse tan pronto como esté listo
- **Controles:** Especifica que los controles se mostrarán como un botón de reproducción.
- **Loop:** Especifica que el archivo multimedia se iniciará de nuevo, una vez que haya terminado.
- **Preload:** Especifica que el audio se cargará en la carga de la página, y listo para reproducir. Se ignora si "autoplay" está activado.
- **Src:** Especifica la dirección URL del audio a reproducir.

### 2.6.3. Etiqueta <canvas>

Ésta etiqueta, junto con las de <video> y <audio> son las más novedosas en este nuevo estándar, pero con la gran diferencia que las anteriores ya eran conceptualmente elementos conocidos y que se usaban anteriormente.

Lo que propone esta etiqueta es algo que no se había hecho nunca antes, y es la de “dibujar” literalmente en la pantalla del navegador sin la necesidad de ninguna aplicación externa, y mediante JavaScript. Lo que se va a poder hacer con <canvas> será desde figuras geométricas, sombras, interfaces de usuarios, gráficos y tablas hasta complejos adornos para una página web y hasta juegos multimedia.

El problema de este elemento es que no es accesible directamente en el navegador, es decir, no hay ningún soporte ni alternativas directas para personas con discapacidad visual que les permita acceder a este elemento. Por lo que para otorgar a este elemento de un mínimo de accesibilidad se le debería de proporcionar al menos un texto alternativo, que le permita identificar al usuario en el momento de usar programas lectores de páginas web o similares que faciliten la accesibilidad.

Básicamente el elemento <canvas> es un área rectangular, en el que el programador controla todos y cada uno de los píxeles que lo forman, creando, modificando y añadiendo efectos para conseguir el dibujo requerido. Este elemento no genera por sí solo la habilidad de “dibujar”, sino que hay que usar JavaScript para desarrollar los objetos y efectos con los que se va a componer lo que queramos dar forma.

Con el código de la figura 13, crearíamos el área rectangular donde vamos a usar el código JavaScript.

```
<canvas id="myCanvas" width="200" height="100"></canvas>
```

*Figura 13: Elemento Canvas*

Con la función `getElementById` de JavaScript señalada en la figura 14, obtenemos el objeto con el id “myCanvas” para encontrar el elemento canvas.

```
var c=document.getElementById("myCanvas");
```

*Figura 14: Código JavaScript para Canvas*

A continuación se crea un objeto context, tal y como se muestra en la figura 15.

```
var cxt=c.getContext("2d");
```

*Figura 15: Segundo código JavaScript para Canvas*

El objeto `getContext("2d")` mostrado en la figura 15, está definido por sí mismo en HTML5 con muchos métodos para dibujar trazados, cajas, círculos, personajes, imágenes y mucho más.

El método `fillStyle` lo colorea en rojo, y el método `fillRect` especifica la forma, posición y tamaño. Significa que crea un rectángulo rojo en `<canvas>` de 150x75 empezando en la esquina superior izquierda (0,0), tal y como se muestra en la figura 16 y mostrando el resultado de la misma en la figura 17.

```
cxt.fillStyle="#FF0000";  
cxt.fillRect(0,0,150,75);
```

*Figura 16: Ejemplo Canvas*



*Figura 17: Resultado del ejemplo Canvas*

## 2.6.4. Etiqueta track

Esta etiqueta aún en desarrollo, permite al desarrollador incluir elementos de texto de forma externa para elementos multimedia tales como videos. De momento, no representa nada por sí misma, pero se espera que sea el comienzo de la introducción de los tan esperados subtítulos en HTML5.

## 2.6 HTML5 y elementos relacionados con el contenido multimedia

Según el último borrador del nuevo estándar HTML5, esta nueva etiqueta permite diferentes atributos:

- **Kind:** permite distintos valores relacionados con elementos multimedia como “subtitles”, “captions”, “descriptions”, “chapters”, “metadata”, y que otorgan a la etiqueta de distintas clases de información seguramente para distinguir lo que son los subtítulos al tipo de elementos multimedia con el que esté relacionado, o capítulos que contenga un vídeo o simplemente información.
- **Src:** como en las demás etiquetas, da la dirección donde se encuentra el elemento track el cual debe existir y contener una URL válida, tales como fichero de información como archivos XML.
- **Srclang:** este atributo indica el idioma de la información en el que está el elemento track, permitiendo subtítulos o la información necesaria en varios idiomas.
- **Label:** este atributo también indica el idioma del elemento track, pero en este caso de cara al usuario por ejemplo para que sea mostrado en su interfaz de usuario.
- **Default:** este atributo indica, en el caso de que haya varios elementos track relacionados con un mismo elemento multimedia, cuál de ellos va a ser el que se va a mostrar al principio, y también en el caso en el que el usuario no haya seleccionado ninguno anteriormente.

Un ejemplo extraído directamente del borrador de HTML5 es el mostrado en la figura :

```
<video src="brave.webm">
  <track kind=subtitles src=brave.en.vtt srclang=en
  label="English">
  <track kind=captions src=brave.en.vtt srclang=en
  label="English for the Hard of Hearing">
  <track kind=subtitles src=brave.fr.vtt srclang=fr
  label="Français">
  <track kind=subtitles src=brave.de.vtt srclang=de
  label="Deutsch">
</video>
```

*Figura 18: Ejemplo del elemento track de HTML5*

Actualmente esta nueva etiqueta no tiene un uso señalado para ninguna funcionalidad en HTML5, pero se espera que sea uno de los elementos más importante para los esperados subtítulos.

En cambio, hay una aplicación de código abierto llamada LeanBack Player [LeanBack Player, 2011], que no se apoya en ningún framework de JavaScript para funcionar, y que utiliza esta etiqueta para otorgar a un vídeo de subtítulos en varios idiomas.

Es un reproductor HTML5 que han desarrollado con la novedad de que permite la incrustación de subtítulos en los videos que queramos mostrar. La forma de hacer esto es muy sencilla, ya que se hará por medio de la propiedad “track” de HTML5.

## Capítulo 2: Introducción a HTML5

El reproductor nos permitirá acceder a los distintos controles por medio de combinaciones de teclas de acceso rápido, así por ejemplo, para el play o para el pause se utiliza la barra espaciadora, mientras que para pararlo, se puede utilizar la “s”.

Para añadir el reproductor a una página, lo primero que debemos hacer es incluir el fichero de estilo, así como los archivos de javascript que forman parte del reproductor, tal y como se observa en la figura 19.

```
<link rel="stylesheet" href="./leanbackPlayer.css"
media="screen" type="text/css" title="LeanBack Player
Style"/>
<script type="text/javascript"
src="./leanbackPlayer.js"></script>
<script type="text/javascript"
src="./leanbackPlayer_en_EN.js"></script>
```

*Figura 19: Código para incluir en la cabecera el reproductor LeanBack Player HTML5*

Lo siguiente es añadir el código correspondiente al reproductor, indicando la ruta hacia los videos, así como la ruta de los subtítulos tal y como se puede observar en la figura 20. También incluirá una parte correspondiente a la ejecución de un reproductor en flash, cuando el navegador no reconoce los tags de HTML5.

```

<div>
  <!-- Video element -->
  <video width="420" height="240" controls
preload="metadata" poster="./poster.jpg">
    <!-- Video sources -->
    <source src="./video.mp4" type='video/mp4;
codecs="avc1.42E01E, mp4a.40.2"'>
    <source src="./video.webm" type='video/webm;
codecs="vp8, vorbis"'>
    <source src="./video.ogv" type='video/ogg;
codecs="theora, vorbis"'>
    <!-- Subtitle sources -->
    <track enabled="true" kind="subtitles"
label="English" srclang="en" type="text/plain"
        src="./subtitle_en.sub"></track>
    <track enabled="true" kind="subtitles" label="France"
srclang="fr" type="application/ttaf+xml"
        src="./subtitle_tt_fr.xml"></track>
    <!-- Flash Fallback see:
https://developer.mozilla.org/En/Using\_audio\_and\_video\_in\_Fir
efox#Using\_Flash
        AND
http://camendesign.com/code/video\_for\_everybody -->
    <object name="flash_fallback" id="flash_fallback"
class="flash_fallback" width="420" height="240"
        type="application/x-shockwave-flash"
data="http://domain/player.swf">
    <param name="movie"
value="http://domain/player.swf" />
    <param name="allowfullscreen" value="true" />
    <param name="allowscriptaccess" value="always" />
    <param name="flashvars"

```

*Figura 20: Código del reproductor LeanBack Player HTML5*

# Capítulo 3

## 3. Accesibilidad Web

En este capítulo se presentan los estándares de accesibilidad del *World Wide Web* (W3C) para asegurar el acceso a la Web a cualquier usuario. Sin tener en cuenta la accesibilidad, no sirve de mucho una página web con mucho contenido, ya que habrá usuarios que no podrán acceder a dichos contenidos.

El W3C contribuye con una serie de estándares a seguir por los desarrolladores de páginas web, y que se incluyen en este capítulo. Estos trabajos son unas pautas para intentar evitar que la Web sea un caos y cada uno haga las cosas como crea conveniente sin tener en mente la accesibilidad. Para ello han creado diferentes guías que permiten al desarrollador crear una página web accesible, como son: las Pautas de accesibilidad al contenido en la Web (WCAG), que explican cómo hacer que el contenido de la Web sea accesible, las Pautas de Accesibilidad para Herramientas de Autor (ATAG) que muestran cómo crear herramientas de autor tal que sean accesibles para personas con discapacidad, y las más importantes para este proyecto, las Pautas de Accesibilidad para Agentes de Usuario (UAAG) que muestran cómo hacer para que los agentes de usuario (navegadores, reproductores multimedia, etc...) sean accesibles y por último ARIA (*Accesible Rich Internet Applications*) que aporta guías para hacer accesible el contenido dinámico y los controles avanzados de interfaz desarrollados con tecnologías como Ajax y tecnologías relacionadas. Todos estos trabajos están incluidos en el W3C dentro de la Iniciativa para la Accesibilidad Web o WAI (*Web Accessibility Initiative*).

### 3.1. Legislación y normativa relativa a la accesibilidad

El uso equitativo de las Tecnologías de la Información y Comunicación (TIC) como Internet es un derecho para todas las personas, pero para las personas con discapacidad es además una oportunidad para poder integrarlas, favoreciendo su autonomía. Algunos países, entre ellos España, han reconocido esta necesidad y se ha regulado a través de la ley. A continuación se va a mostrar legislación, normas y directivas que son aplicables a la accesibilidad web. Estas leyes, normas y/o directivas deben tenerse en cuenta cuando se va a desarrollar una aplicación web [Moreno L., 2010].

#### 3.1.1. Marco Internacional



En cuanto a legislación a nivel internacional, se pueden considerar como punto de partida las “Normas Uniformes sobre la igualdad de oportunidades para las personas con discapacidad” [UN, 1993], aprobadas por las Naciones Unidas el 20 de diciembre de 1993, cuya finalidad es “garantizar que niñas y niños, mujeres y hombres con discapacidad, en su calidad de miembros de sus respectivas sociedades, puedan tener los mismos derechos y obligaciones que los demás”. Los fundamentos políticos y morales de estas normas se encuentran en la “Carta Internacional de Derechos Humanos” [UN, 1948]. En el contenido de las normas puede leerse que “aunque no son de cumplimiento obligatorio, pueden convertirse en normas internacionales cuando las aplique un gran número de Estados con la intención de respetar una norma de derecho internacional. Llevan implícito el compromiso de los Estados de adoptar medidas para lograr la igualdad de oportunidades”. Posteriormente fue aprobada en la Asamblea General de la ONU el 13 de diciembre de 2006 la Convención de Derechos de las Personas con Discapacidad, que entró en vigor el 3 de mayo de 2008, después de haber sido firmada por más de 100 países y ratificada por 20 gobiernos o parlamentos. Los legisladores de muchos países se basan en estas Normas sobre la Igualdad de Oportunidades para las Personas con Discapacidad para legislar específicamente sobre la accesibilidad en la Sociedad de la Información, como en el caso de la accesibilidad web.

En Europa existen iniciativas como el Plan de Acción eEurope 2002, eEurope 2005 e i2010 por la Comisión y Consejo de Europa. De todas ellas destaca como primer antecedente, en relación a la accesibilidad, la línea de actuación de Europa 2002 [EU, 2002] cuyo objetivo principal fue “mejorar el acceso a la Web de personas con discapacidades”, en consonancia con “el principio de no-discriminación, proclamado en el Tratado de la Unión Europea”, donde se estableció el final de 2001 como fecha límite para la adopción de las Pautas de la Iniciativa de Accesibilidad a la Web (WAI), WCAG 1.0.

Como marco normativo relacionado destacan las normas: ISO 9241-171:2008- Accesibilidad al Software [ISO, 2008 b], ISO 9241-20:2008- Accesibilidad en productos y servicios TIC [ISO, 2008 c] y ISO 9241-151:2008-Ergonomía de interfaces web [ISO, 2008 d].

### 3.1.2. Marco nacional

En España la legislación en materia de accesibilidad web cuenta con el instrumento de “Ratificación de la Convención de Derechos de las Personas con Discapacidad”, dado el 21 de abril de 2008 [BOE, 2008]. Además están:

- Ley de servicios de la Sociedad de la Información y de comercio electrónico. La Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSICE) destaca que se promoverá la adopción de normas de accesibilidad por los prestadores de servicios y los fabricantes de equipos y software, para facilitar el acceso de las personas con discapacidad o de edad avanzada a los contenidos digitales [BOE, 2002].
- Ley de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad (LIONDAU). Ley 51/2003, de 2 de diciembre, de Igualdad de Oportunidades, no Discriminación y Accesibilidad Universal de para el desarrollo de la accesibilidad [BOE, 2003].

Para organizar la puesta en marcha de la LIONDAU se consideró conveniente la elaboración de instrumentos de planificación, y durante su redacción se diseñaron dos planes: el

Plan Nacional de Accesibilidad 2004-2012 y el II Plan de Acción para las personas con discapacidad 2003-2007. En respuesta se han ido habilitando decretos como el Real Decreto 1494/2007 de 12 de noviembre del 2007, por el que se aprueba el “Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la Sociedad de la Información y medios de comunicación social” [BOE, 2007 a]. En este decreto se establece que algunos sitios web deben ser accesibles según la Norma UNE 139803:2004 [AENOR, 2004] con unos plazos definidos. 38

La norma UNE 139803:2004 recoge los requisitos que han de cumplir los contenidos disponibles en Internet y otros tipos de redes informáticas, para que puedan ser utilizados por la mayor parte de las personas. Se aplica a cualquier tipo de contenido disponible en redes informáticas, con especial énfasis en los contenidos web que son accedidos mediante navegadores de Internet. Se establecen tres Niveles de Prioridad de los puntos normativos.

Según el Real Decreto 1494/2007 a partir del 2009 los sitios web de las Administraciones Públicas y otros organismos (financiados por la Administración Pública, entidades bancarias, aseguradoras, etc.) deben ser accesibles y deberían cumplir con los requisitos de Prioridad 2 de esta norma. Además, según el decreto hay que indicar en las páginas web el grado de accesibilidad, la fecha de la revisión e incluir un sistema de contacto específico a los usuarios para transmitir las dificultades de acceso.

Muy relativa a este trabajo la norma UNE 139802:2009 (Requisitos de accesibilidad del software) [AENOR, 2009], versión oficial en español de la ISO 9241-171:2008 [10]. Incluye cuatro pautas a cumplir para que un reproductor sea considerado accesible, estas pautas son: (1) el reproductor debe permitir que el usuario pueda detener, iniciar y pausar la reproducción del vídeo, (2) es necesario permitir que el usuario pueda repetir, rebobinar, pausar, adelantar o avanzar de forma rápida una reproducción, (3) se debe permitir que el usuario pueda controlar la presentación de múltiples flujos multimedia, (4) permitir actualizar alternativas equivalentes del contenido multimedia cuando se produzca un cambio en él. Asimismo la UNE 139802:2009 incluye otros requisitos relativos al contenido alternativo subtulado, indicando que al mostrar los subtítulos el contraste con el fondo sea suficiente, que su colocación no oculte el contenido principal a reproducir, que se puedan activar o desactivar y que permitan adaptarse a cambios en la configuración de preferencias [González M. et al, 2011].

En definitiva, existen multitud de directivas que especifican requisitos que deben cumplir las aplicaciones web en algunos ámbitos como el de la administración pública y otros. De este modo, productos que no cumplan las condiciones de accesibilidad establecidas, pueden verse rechazados en este ámbito.

## 3.2. Iniciativa de Accesibilidad Web (WAI)

Hablar de Accesibilidad Web es hablar de un acceso universal a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica y capacidades de los usuarios.

Con esta idea de accesibilidad nace la Iniciativa de Accesibilidad Web (WAI) [W3C, 2010 c]. Se trata de una actividad desarrollada por el W3C, cuyo objetivo es facilitar el acceso de las personas con discapacidad, desarrollando pautas de accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad Web, llevando a cabo una labor

educativa y de concienciación en relación a la importancia del diseño accesible de páginas web, y abriendo nuevos campos en accesibilidad a través de la investigación en este área.

La idea principal radica en hacer la Web más accesible para todos los usuarios independientemente de las circunstancias y los dispositivos involucrados a la hora de acceder a la información. Partiendo de esta idea, una página accesible lo será tanto para una persona con discapacidad, como para cualquier otra persona que se encuentre bajo circunstancias externas que dificulten su acceso a la información (en caso de ruidos externos, en situaciones donde nuestra atención visual y auditiva no esté disponible: pantallas con visibilidad reducida, etc.).

### 3.2.1. Pautas de Accesibilidad al Contenido en la Web (WCAG)

Para hacer el contenido web accesible, se han desarrollado las denominadas Pautas de Accesibilidad al Contenido en la Web (WCAG) [W3C, 2008 d], cuya función principal es guiar el diseño de páginas web hacia un diseño accesible, reduciendo de esta forma barreras a la información. WCAG consiste en 14 pautas que proporcionan soluciones de diseño y que utilizan como ejemplo situaciones comunes en las que el diseño de una página puede producir problemas de acceso a la información. Las Pautas contienen además una serie de puntos de verificación que ayudan a detectar posibles errores.

Cada punto de verificación está asignado a uno de los tres niveles de prioridad establecidos por las pautas.

- **Prioridad 1:** son aquellos puntos que un desarrollador web tiene que cumplir ya que, de otra manera, ciertos grupos de usuarios no podrían acceder a la información del sitio web.
- **Prioridad 2:** son aquellos puntos que un desarrollador web debería cumplir ya que, si no fuese así, sería muy difícil acceder a la información para ciertos grupos de usuarios.
- **Prioridad 3:** son aquellos puntos que un desarrollador web debería cumplir ya que, de otra forma, algunos usuarios experimentarían ciertas dificultades para acceder a la información.

En función a estos puntos de verificación se establecen los niveles de conformidad:

	<b>Nivel de Conformidad "A":</b> todos los puntos de verificación de prioridad 1 se satisfacen.
	<b>Nivel de Conformidad "Doble A":</b> todos los puntos de verificación de prioridad 1 y 2 se satisfacen.
	<b>Nivel de Conformidad "Triple A":</b> todos los puntos de verificación de prioridad 1,2 y 3 se satisfacen.

Tabla 5: Niveles de conformidad en WCAG 1.0

Por lo tanto en la versión 1.0 una página web tiene un nivel de adecuación A si se cumplen todos los criterios de éxito de nivel A; tiene la página web un nivel de adecuación AA si se cumplen todos los criterios de éxito nivel A y AA; y una página web puede tener nivel de adecuación AAA si se cumplen todos los criterios de nivel A, AA y AAA.

Las pautas describen cómo hacer páginas web accesibles sin sacrificar el diseño, ofreciendo esa flexibilidad que es necesaria para que la información sea accesible bajo diferentes situaciones y proporcionando métodos que permiten su transformación en páginas útiles e inteligibles.

Las Pautas de Accesibilidad de Contenidos Web 1.0 (WCAG 1.0) [W3C, 1999 b] se crearon en 1999 para guiar el diseño de las páginas web accesibles.

Las WCAG están pensadas principalmente para:

- Desarrolladores de contenido web (desarrolladores de páginas web, diseñadores de sitios web, etc.).
- Desarrolladores de herramientas de autor para la Web.
- Desarrolladores de herramientas de evaluación de accesibilidad web.

La documentación de las pautas tiene como objetivo satisfacer las necesidades de diferentes usuarios, incluyendo creadores de políticas de accesibilidad, directivos y otros.

Las WCAG 1.0 forman la versión de referencia hasta este pasado diciembre del 2008 que se publicaron las WCAG 2.0 [W3C, 2008 b] como Recomendación, después de muchos años de trabajo y borradores publicados desde 2006. Las WCAG 2.0 tienen un enfoque distinto a las WCAG 1.0 pues han sido desarrolladas para aplicarse a tecnologías actuales y emergentes y, no sólo a las tecnologías del W3C. A su vez, se busca que su utilización y comprensión sea sencilla y más precisa.

Las WCAG 1.0 y las WCAG 2.0 están organizadas y estructuradas de distinta manera tal como indica la Tabla 4 [Tesis Doctoral Lourdes Moreno, Febrero 2010].

WCAG	Estructura	Puntos de Verificación, Criterios de Conformidad	Nivel de Conformidad	Documentación
1.0 (1999)	14 pautas con 65 puntos de verificación	Los Puntos de Verificación tienen asignado un Nivel de Prioridad 1, 2 o 3.	Nivel A, AA, AAA	Técnicas fundamentales Técnicas HTML Técnicas CSS
2.0 (2008)	4 principios básicos con 12 pautas que contienen un total de 61 criterios de éxito.	Los Criterios de Conformidad tienen asignados un Nivel de Conformidad A, AA ó AAA.	Nivel A, AA, AAA	Como satisfacer / Comprender Técnicas suficientes y aconsejables.

Tabla 6: Estructura de las WCAG 1.0 y WCAG 2.0

### ***Pautas de Accesibilidad de Contenidos Web 1.0 (WCAG 1.0)***

Como se ha indicado las WCAG 1.0 constan de 14 pautas que se detallan a continuación, acompañadas por una explicación:

- **Pauta 1** - "Proporcione alternativas equivalentes para el contenido visual y auditivo".

Proporcione un contenido que, presentado al usuario, cumpla esencialmente la misma función o propósito que el contenido visual o auditivo.

### Capítulo 3: Accesibilidad Web

- **Pauta 2** - No se base sólo en el color.

Asegúrese de que los textos y gráficos son comprensibles cuando se vean sin color.

- **Pauta 3** - Utilice marcadores y hojas de estilo y hágalo apropiadamente.

Marque los documentos con los elementos estructurales apropiados. Controle la presentación con hojas de estilo en vez de con elementos y atributos de presentación.

- **Pauta 4** - Identifique el idioma usado.

Use marcadores que faciliten la pronunciación o interpretación de texto abreviado o extranjero.

- **Pauta 5** - Cree tablas que se transformen correctamente.

Asegure que las tablas tienen los marcadores necesarios para transformarlas mediante navegadores accesibles y otras aplicaciones de usuario.

- **Pauta 6** - Asegúrese de que las páginas que incorporan nuevas tecnologías se transformen correctamente.

Asegúrese de que las páginas son accesibles incluso cuando no se soportan las tecnologías más modernas o éstas estén desconectadas.

- **Pauta 7** - Asegure al usuario el control sobre los cambios de los contenidos tiempo-dependientes.

Asegúrese de que los objetos o páginas que se mueven, parpadean, se desplazan o se actualizan automáticamente, puedan ser detenidos o parados.

- **Pauta 8** - Asegure la accesibilidad directa de las interfaces de usuario incrustadas.

Asegure que la interfaz de usuario sigue los principios de un diseño accesible: funcionalidad de acceso independiente del dispositivo, teclado operable, voz automática, etc.

- **Pauta 9** - Diseñe para la independencia del dispositivo.

Utilice características que permitan la activación de los elementos de la página a través de diversos dispositivos de entrada.

- **Pauta 10** - Utilice soluciones provisionales.

Utilice soluciones de accesibilidad provisionales de forma que las ayudas técnicas y los antiguos navegadores operen correctamente.

- **Pauta 11** - Utilice las tecnologías y pautas W3C.

Utilice tecnologías W3C (de acuerdo con las especificaciones) y siga las pautas de accesibilidad. Donde no sea posible utilizar una tecnología W3C, o usándola se obtengan materiales que no se transforman correctamente, proporcione una versión alternativa del contenido que sea accesible.

- **Pauta 12** - Proporcione información de contexto y orientación.

Proporcione información de contexto y orientativa para ayudar a los usuarios a entender páginas o elementos complejos.

- **Pauta 13** - Proporcione mecanismos claros de navegación.

Proporcione mecanismos de navegación claros y coherentes, (información orientativa, barras de navegación, un mapa del sitio, etc.) para incrementar la probabilidad de que una persona encuentre lo que está buscando en un sitio.

- **Pauta 14** - Asegúrese de que los documentos sean claros y simples.

Asegure que los documentos son claros y simples para que puedan ser más fácilmente comprendidos.

### ***Pautas de Accesibilidad de Contenidos Web 2.0 (WCAG 2.0)***

A continuación se van a mostrar los cuatro principios generales y las doce pautas de las WCAG 2.0:

- **Principio 1: Perceptible** - La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos.
  - **Pauta 1.1** Alternativas textuales: Proporcionar alternativas textuales para todo contenido no textual de modo que se pueda convertir a otros formatos que las personas necesiten, tales como textos ampliados, braille, voz, símbolos o en un lenguaje más simple.
  - **Pauta 1.2** Medios tiempo-dependientes: Proporcionar alternativas sincronizadas para los medios tiempo-dependientes.
  - **Pauta 1.3** Adaptable: Crear contenido que pueda presentarse de diferentes formas (por ejemplo, con una disposición más simple) sin perder información o estructura.
  - **Pauta 1.4** Distinguible: Facilitar a los usuarios ver y oír el contenido, incluyendo la separación entre el primer plano y el fondo.
- **Principio 2: Operable** - Los componentes de la interfaz de usuario y la navegación deben ser operables.
  - **Pauta 2.1** Accesible por teclado: Proporcionar acceso a toda la funcionalidad mediante el teclado.
  - **Pauta 2.2** Tiempo suficiente: Proporcionar a los usuarios el tiempo suficiente para leer y usar el contenido.
  - **Pauta 2.3** Convulsiones: No diseñar contenido de un modo que se sepa podría provocar ataques, espasmos o convulsiones.
  - **Pauta 2.4** Navegable: Proporcionar medios para ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran.
- **Principio 3: Comprensible** - La información y el manejo de la interfaz de usuario deben ser comprensibles.
  - **Pauta 3.1** Legible: Hacer que los contenidos textuales resulten legibles y comprensibles.
  - **Pauta 3.2** Predecible: Hacer que las páginas web aparezcan y operen de manera predecible.
  - **Pauta 3.3** Entrada de datos asistida: Ayudar a los usuarios a evitar y corregir los errores.

- **Principio 4: Robusto** - El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo las ayudas técnicas.
  - **Pauta 4.1 Compatible:** Maximizar la compatibilidad con las aplicaciones de usuario actuales y futuras, incluyendo las ayudas técnicas.

### 3.2.2. Pautas de Accesibilidad para Herramientas de Autor (ATAG)

Las Pautas de Accesibilidad para Herramientas de Autor (ATAG) muestran cómo hacer que las herramientas de autor sean accesibles para personas con discapacidad. Estas herramientas son software que se utiliza para crear páginas y contenido web. Uno de los objetivos principales de las ATAG es definir la forma en la que las herramientas ayudan a los desarrolladores web a producir contenido web que cumpla las Pautas de Accesibilidad al Contenido en la Web [W3C ATAG, 2005].

Las ATAG están pensadas principalmente para desarrolladores de herramientas de autor. Entre estas herramientas de autor se incluyen, al ser un estándar no relativo directamente con este proyecto, no se va a profundizar en sus pautas.

### 3.2.3. Pautas de Accesibilidad para Agentes de Usuario (UAAG)

Los documentos de Pautas de Accesibilidad para Agentes de Usuario (UAAG) [W3C UAAG, 2005] muestran cómo hacer que los agentes de usuario sean accesibles para personas con discapacidad, en especial cómo incrementar la accesibilidad al contenido web. Entre los agentes de usuario se incluyen navegadores, reproductores multimedia y productos de apoyo, software que algunas personas con discapacidad utilizan para interactuar con los dispositivos. Por ser el estándar a seguir para hacer un reproductor accesible, sí se ha sido tenido en cuenta en este proyecto, incluyendo mucho de los requisitos en el diseño y desarrollo del interfaz caso de uso del que consta este proyecto (ver capítulo 5). Servir vídeo a través de HTML5 hace que se tenga que dar soporte en este mismo estándar o con la incorporación de otras tecnologías para poder incluir todos estos requisitos descritos en las UAAG, ya sea con la ayuda de JavaScript, Flash o CSS.

Tanto las UAAG 1.0 [W3C, 2002] como otros documentos complementarios tienen como objetivo satisfacer las necesidades de usuarios diversos, creadores de políticas, directivos y otros. Por ejemplo:

- Aquellos usuarios que deseen elegir agentes de usuario más accesibles pueden utilizar las UAAG para evaluar los agentes de usuario
- Aquellos que por otro lado quieran animar a que los desarrolladores de agentes de usuario existentes mejoren la accesibilidad en versiones futuras, pueden indicar a los proveedores de agentes de usuario como referencia las UAAG.

Las UAAG 1.0 contienen un conjunto de puntos de verificación que incluyen:

- Acceso a todo el contenido, incluyendo contenido en relación de eventos generados por el ratón o el teclado
- Control del usuario sobre la forma en que se muestra el contenido

- Control del usuario sobre la interfaz del usuario, con documentación sobre características de accesibilidad
- Interfaces de programación estándares, para permitir la interacción con tecnologías de apoyo

Desde el lanzamiento de UAAG 1.0 como Recomendación del W3C en diciembre de 2002, el Grupo de trabajo para las Pautas de Accesibilidad para agentes de usuario (UAWG) ha recibido comentarios sobre la usabilidad, comprensibilidad y aplicabilidad del conjunto de los documentos. Además, en los años transcurridos se han producido cambios y mejoras en:

- Tecnologías y técnicas que se utilizan en el contenido web.
- La funcionalidad de la tecnología de asistencia.
- Acceso a las interfaces de programación de aplicaciones (API).
- Plataformas para recibir el contenido.

Los comentarios, cambios, y la información obtenida de la evaluación de los agentes de usuario utilizando bancos de pruebas para desarrollar informes de ejecución, están impulsando el desarrollo de UAAG 2.0 [W3C, 2010 b] y es capturado como los requisitos para la misma.

El objetivo principal de UAAG 2.0 es el mismo que la versión 1.0, reducir las barreras a la accesibilidad web para personas con discapacidad.

Actualmente, las UAAG 2.0 es un documento en borrador por lo que está en continuo cambio y modificación, y como borrador que es, todavía no está reconocido por la propia W3C, y por lo tanto, no sustituye a las UAAG 1.0.

Los cambios sustanciales incluyen:

- Actualización de las definiciones de los criterios de éxito y glosario, en el que se detalla el comportamiento deseado de concentración y control del usuario sobre una indicación visual del foco.
- Actualización de los criterios de éxito para la conservación de las diferencias de tamaño relativo a las fuentes cuando su tamaño se escala. El cambio reconoce los casos de uso, cuando hay una necesidad de accesibilidad para eliminar las diferencias relativas al tamaño y que dan control al usuario.
- Actualización de los criterios de éxito en el envío de formularios para permitir al usuario redefinir los métodos abreviados de teclado para la presentación y cancelación de dichos formularios.
- Eliminar los criterios de éxito del punto 3.3.6 debido a que no era específico para usuarios con discapacidad.

Las directrices fundamentales de la versión 2.0 de las UAAG son:

- **Principio 1:** Cumplir con las especificaciones aplicables y los convenios.
- **Principio 2:** Facilitar el acceso mediante programación.
- **Principio 3:** Perceptible – La interfaz del usuario y el contenido debe ser presentado a los usuarios de manera que lo puedan percibir.
- **Principio 4:** Asegurarse de que la interfaz del usuario es operable.
- **Principio 5:** Asegurarse de que la interfaz del usuario es comprensible.

Los agentes de usuario pueden reclamar la conformidad con UAAG 2.0 en uno de los tres niveles de conformidad. El nivel alcanzado depende del nivel de los criterios de éxito que se han cumplido. Los niveles de conformidad que también comparte con la versión 1.0 de las UAAG son los siguientes:

- **Conformidad con el nivel “A”:** el agente de usuario cumple con todos los criterios de éxito de nivel A.
- **Conformidad con el nivel “AA”:** el agente de usuario cumple con todos los criterios de éxito de nivel A y AA.
- **Conformidad con el nivel “AAA”:** el agente de usuario cumple con todos los criterios de éxito.

Si se siguen los requisitos incluidos en las UAAG 2.0, hay unos elementos a incorporar en la interfaz del reproductor si se quiere incluir de manera accesible un vídeo al que se acceda a través de un reproductor en una página web. Según [González M. et al, 2011] hay unos controles básicos como: reproducir o parar el vídeo, permitir cambiar el tamaño de la ventana, permitir ajustar el volumen y unos controles adicionales tales como: permitir activar o desactivar los subtítulos, permitir activar o desactivar la audiodescripción, permitir realizar búsquedas dentro de los subtítulos de la reproducción, permitir adelantar o atrasar segundos en una reproducción, permitir cambiar el tamaño, la fuente y el color del texto, permitir acceso a la documentación de ayuda donde se informe sobre los atajos de teclado existentes. La mayoría de estos controles han sido requisito en este proyecto fin de carrera

### 3.2.4. Web Accessibility Initiative - Accessible Rich Internet Application (WAI-ARIA)

Pero la función que tienen todas estas aplicaciones no sería posible sin una buena accesibilidad, y de esto es lo que se encarga la WAI-ARIA (*Web Accessibility Initiative - Accesible Rich Internet Application*) [W3C, 2011 d].

WAI-ARIA es una iniciativa del W3C que define cómo hacer accesibles contenidos y aplicaciones web, específicamente el contenido dinámico y los controles avanzados de interfaz desarrollados con Ajax, HTML, Javascript y sus tecnologías relacionadas.

Pretende ser una ayuda en el contenido dinámico de los interfaces actuales de las aplicaciones web, que cada día son más parecidos a los entornos de escritorio y que su propio funcionamiento puede interferir en los productos de apoyo.

Esta tecnología tiene como principal objetivo aportar información acerca de las diferentes partes que constituyen los contenidos dinámicos generados, normalmente, por medio de *scripts*. Toda esta información será utilizada por los productos de apoyo para la interacción con el usuario final.

Los documentos técnicos de WAI ARIA están desarrollados por el grupo de trabajo del W3C PFWG (*Protocols and Formats Working Group*). El W3C pone a disposición varios documentos sobre WAI-ARIA, entre ellos, destaca la especificación técnica de la W3C.

### **¿Cómo funciona?**

WAI ARIA proporciona una serie de atributos que funcionan como identificadores de las diferentes partes de la aplicación que interactúa con el usuario. También se incluyen mapeo de controles y eventos para la accesibilidad de las APIs (*Application Programming Interfaces*).

WAI ARIA dispone de roles que describen tanto los *widgets* (componentes con funcionalidad propia de las interfaces de escritorio o web) de la aplicación como la estructura de la página web, como por ejemplo: los encabezados y las regiones. También dispone de varias propiedades como los estados de los *widgets*, las regiones activas de actualización de contenidos y sobre características *drag-and-drop*. A su vez, provee una manera de navegar mediante teclado dentro de los componentes [INTECO, 2010].

### **Ventajas y desventajas**

Al usar esta tecnología se encuentran una serie de ventajas:

- Al añadir valor semántico al contenido y a los *widgets* se puede situar al usuario exactamente en donde está. Además de su facilidad de implementación ya que son los atributos los que los define.
- Las actualizaciones de contenido dinámico, normalmente realizado con tecnologías de *script*, son notificadas al usuario por medio de su producto de apoyo.
- Accesibilidad de los *widgets* por medio del teclado.
- Se dota de información de cómo se utiliza un *widget* y qué tipo de datos proporciona.

En cambio, se puede identificar una desventaja manifiesta, ya que al aplicar esta tecnología, los documentos web que se desarrollen no validarán tanto en HTML4 como en XHTML 1.0. El primero no soporta espacio de nombres, por lo que no se podrá incluir los atributos específicos de WAI ARIA. Con XHTML 1.0, su validación no sería posible porque no estaría incluido en el esquema que define al lenguaje.

Una posible solución sería incluir los atributos de WAI ARIA mediante DOM, es decir, incluir con JavaScript, dinámicamente, los atributos en los elementos destinados a ellos. Dicha solución haría dependiente de la tecnología *script* cualquier tipo de implementación.

Existe otra solución que podría resolver el problema: la utilización de XHTML 1.1. Este lenguaje es una especificación de lenguaje modular y extensible que permitiría personalizar el DTD incorporando el esquema de WAI ARIA [INTECO, 2010].

### **¿Por qué WAI-ARIA?**

Tanto HTML4.01 como HTML5 fueron creados para la presentación de documentos textuales con formatos específicos para su función, como puedan ser los encabezados, marcadores de párrafos o cualquier otro tipo de elementos que tuvieran que ver con el formateo de texto. Esto es así desde que en las primeras versiones se incluyó el elemento IMG para la incorporación de elementos de imágenes dentro de los documentos.

El lenguaje fue desarrollado para un modelo de cliente-servidor, donde el usuario pedía el documento y desde un servidor se lo proporcionaba. Esta comunicación está orientada de forma secuencial, es decir, un usuario envía una petición a un servidor, normalmente de un documento HTML, y el servidor lo procesa para, finalmente, enviarlo al usuario final.

Con la evolución de la web, las páginas han ido evolucionando hacia las llamadas aplicaciones web, mucho más parecidas a los entornos de escritorio que tienen un

funcionamiento más dinámico con respecto a los contenidos. Para ello, se han utilizado tecnologías web como JavaScript, y más concretamente AJAX, que de unos años hasta hoy ha tenido un crecimiento sustancial.

También está presente el problema de la independencia de dispositivo, ya que algunos componentes de las aplicaciones web enriquecidas no son accesibles desde el teclado, orientando su uso al ratón. Esta característica entraña un grave problema de accesibilidad, dado que muchas personas con discapacidad física son incapaces de utilizar el ratón [INTECO, 2010].

### ***Las soluciones que aporta ARIA***

ARIA viene a responder a las anteriores preguntas y otras cuestiones proporcionando un marco de trabajo complementario.

- Estructuras más semánticas para las zonas funcionales.
- Mejora de la navegación mediante el teclado.
- Controles complejos (*widgets*) más accesibles.
- Accesibilidad para el contenido actualizado de forma dinámica.

### 3.2.5. Roles

Su misión es definir el papel que juegan los elementos dentro del documento web. Como se ha comentado antes, los roles dentro de WAI ARIA sirven para describir los *widgets*, así como para definir elementos propios de las páginas web. Estos atributos proporcionan información a los productos de apoyo sobre los componentes *widgets*.

Es evidente que, por definición, las etiquetas de HTML tienen un rol predefinido, el mismo que el propio elemento indica. Un rol de WAI ARIA ayuda a concretar la funcionalidad de la etiqueta en la que va definida y se podría afirmar que la sustituye, aunque en nuestro caso, HTML5 dispone de etiquetas mejor definidas semánticamente que su antecesor.

Los roles *Landmark* son un tipo que permite dotar de un significado a ciertas regiones que tienen una funcionalidad o propósito bien marcado y relevante para el uso de cualquier usuario. Las regiones destinadas a la navegación o a la búsqueda de términos pueden ser consideradas como *Landmarks*. Algunos tipos de roles *Landmarks* son:

- Article
- Banner
- Complementary
- Contentinfo
- Main
- Navigation
- Search
- Form

### 3.2.6. Estados y propiedades

WAI-ARIA define diferentes tipos de propiedades, los cuales determinan las características y los valores de cada elemento. Las propiedades pretenden ser una guía a los

productos de apoyo que proporciona información de cómo interactuar con ciertos *widgets* que se encuentran en una página web [INTECO, 2010].

A diferencia de los roles, en el que sólo existe un atributo *role* y varios valores (*navigation, menu, search, etc...*), los atributos de los estados y propiedades pueden ser varios, y cada uno puede aceptar varios valores.

Existen estados y propiedades de carácter global, es decir, pueden ser utilizados independientemente del elemento o del tipo de rol que tenga. Pero en la mayoría de los casos la clasificación de los estados y propiedades viene distribuida de la siguiente forma:

- Atributos *Widget*
- Atributos de regiones activas
- Atributos de Drag and Drop
- Atributos de relaciones

### 3.3. Accesibilidad al contenido multimedia en la Web

En esta sección se va a mostrar cómo ofrecer accesibilidad a los contenidos audiovisuales y multimedia en la Web, siguiendo las recomendaciones del estándar WCAG. Destacar que además de esta opción, hay soluciones tecnológicas de gran interés en el área de la multimedia al margen de las tecnologías W3C como puede ser la aplicación comercial Flash de Adobe [Adobe, 2008] para producir contenidos interactivos y distintos programas de presentación multimedia. La información empleada para la realización de esta apartado, se ha extraído del libro “Accesibilidad a los contenidos audiovisuales en la web: Una panorámica sobre legislación, tecnologías y estándares”, en el que se recoge las recomendaciones de la nueva versión del estándar WCAG 2.0 para los contenidos multimedia y audiovisuales en la web.

Para proveer de accesibilidad a los contenidos audiovisuales, tal como se vio en el apartado 3.2 relativa a la iniciativa WAI que incluyen las WCAG y UAAG, distintos componentes de desarrollo web e interacción deben trabajar conjuntamente para conseguirlo. A esta situación se la ha denominado “la cadena de la accesibilidad de los contenidos audiovisuales en la Web”. Esta cadena consta de los siguientes eslabones a tener en cuenta [Moreno L. et al., 2008 a]:

1. Contenido en sí: el contenido debe ser accesible.
2. Cómo llegar al contenido: el acceso a ese contenido debe ser accesible.
3. Visualización del contenido: hay que ofrecer alternativas atendiendo a preferencias del usuario, y la interacción del usuario al contenido debe ser usable.

Como primer eslabón de la cadena está el conseguir que el contenido en sí mismo sea accesible, lo que significa proveer de alternativas sincronizadas como el subtítulo, la audiodescripción o la transcripción, entre otros, siguiendo las WCAG.

Continuando con la cadena se considerará el siguiente eslabón: la forma de incluir el vídeo para reproducirlo en la Web, ya que aun siendo accesible el contenido audiovisual, cómo se implemente este acceso puede llegar a provocar barreras de accesibilidad. Como alternativas para ofrecer el contenido nos encontramos con la descarga directa, descarga progresiva o falso *streaming*, o con la modalidad de difusión por *streaming*. Cada una de estas opciones conlleva

formas distintas de implementación en una página web, pero todas las opciones deben seguir las WCAG, que se hablará en el siguiente

En el último eslabón de la cadena ante la diversidad de reproductores, tipos de conexiones, etc. es conveniente ofrecer alternativas atendiendo a preferencias del usuario. Pero el acceso del usuario al contenido debe ser, además de posible, intuitivo en la interacción, por lo que hay que ofrecer usabilidad, aportando accesibilidad en cómo mostrar la información de acceso y control al usuario.

### 3.3.1. Primer eslabón: el contenido deber ser accesible.

La accesibilidad a los contenidos audiovisuales se trata en las WCAG. En la versión WCAG 1.0 se tratan de forma muy general y poco precisa. Se habla de descripción auditiva y de subtítulo alternativo a la banda visual. En cambio, la WCAG 2.0 es más exigente y pretende ser más concreta.

Contiene más puntos a revisar: se traduce en nueve criterios de éxito, en comparación a sólo los dos puntos de verificación que hay actualmente, y se distingue entre multimedia pregrabada y multimedia en directo para el subtítulo, audiodescripción y audiodescripción extendida. Además, se considera como nuevo contenido alternativo la lengua de signos. En las WCAG 2.0 hay que ofrecer estos contenidos alternativos para obtener distintos niveles de conformidad tal como se indica en la tabla 7.

1.2.2	Captions (Prerecorded)	A	1.4	1
1.2.3	Audio Description or Full Text Alternative	A	1.4	1
1.2.4	Captions (Live)	AA	1.4.	1
1.2.5	Audio Description	AA	1.3, 1.4	1
1.2.6	Sign Language	AAA	Not mapping	
1.2.7	Audio Description (Extended)	AAA	1.3, 1.4	1
1.2.8	Full Text Alternative	AAA	1.4.	1
1.2.9	Live Audio-only	AAA	1.1.	1

*Tabla 7: Correspondencia de las pautas WCAG1.0 con las WCAG2.0 en relación a los contenidos audiovisuales*

Las dos pautas aplicables a los contenidos audiovisuales en la Web de las WCAG 1.0 son la 1.3 y 1.4, ambas de prioridad 1, y establecen lo siguiente:

- 1.3 Hasta que las aplicaciones de usuario puedan leer en voz alta automáticamente el texto equivalente de la banda visual, proporcione una descripción auditiva de la

información importante de la banda visual de una presentación multimedia. [Prioridad 1].

- 1.4 Para toda presentación multimedia tiempo dependiente (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación. Según esto, habría que proporcionar transcripción de todo el contenido de la banda. [Prioridad 1].

Así, hay que proporcionar contenidos alternativos al contenido audiovisual como la audiodescripción integrada en el contenido o bien, ofrecer una descripción textual o transcripción que transmita de forma completa toda la información (información de los personajes, transcripción completa, acciones, lenguaje corporal, contexto y cambios de escena, etc.). Además, hay que ofrecer subtítulo y de manera sincronizada. Este subtítulo debe transmitir toda la información de la banda auditiva.

En las WCAG 2.0, la pauta que trata los contenidos multimedia es la 1.2, pero hay que hacer referencia al criterio de éxito 1.1.1 también aplicable, que indica que para todo contenido no textual se debe proporcionar al usuario una alternativa textual equivalente.

La pauta 1.2 indica que se deben proporcionar contenidos alternativos sincronizados a los contenidos multimedia, con distintos criterios de éxito como:

- 1.2.1 Para todo contenido sólo-audio o sólo-vídeo pregrabado se debe proporcionar una alternativa equivalente que transmita de forma completa la información de la pista de audio o de vídeo respectivamente. [Nivel A].

Así, hay que:

- Proporcionar una descripción textual alternativa a la pista de audio que transmita la misma información que ésta.
- Proporcionar una descripción auditiva de los elementos claves de la pista visual que transmita la misma información que ésta. Dicha descripción debe incluir información de los personajes, transcripción completa, acciones, lenguaje corporal, contexto y cambios de escena, etc.
- 1.2.2 Se deben proporcionar subtítulos para cualquier contenido de audio pregrabado estando sincronizados además con la pista sonora. La única excepción es que dicho audio sea a su vez alternativa de otro contenido. [Nivel A]. Así, hay que:
  - Proporcionar subtítulos abiertos o cerrados. Éstos deben representar la información contenida en la pista de audio, por tanto deben recoger tanto los diálogos de los personajes como los eventos sonoros que se pudieran producir. Éstos deben estar sincronizados con la pista visual.
- 1.2.3 Se debe proporcionar audiodescripción o alternativa textual completa de la pista visual. [Nivel A].
  - Así, hay que:
    - Proporcionar una descripción auditiva de los elementos claves de la pista visual.
    - Dicha descripción auditiva debe transmitir de forma completa toda la información de la pista visual, a saber, información de los personajes, transcripción completa, acciones, lenguaje corporal, contexto y cambios de escena, etc.
    - Como alternativa a la pista sonora equivalente se puede proporcionar una descripción textual completa que pueda ser leída por un lector de pantalla. Esta

descripción textual debe transmitir la misma información que la pista sonora equivalente.

- La pista visual debe ir acompañada por una descripción auditiva sincronizada con la pista visual y la pista sonora.
- 1.2.4 Se deben proporcionar subtítulos para cualquier contenido audio en directo. [Nivel AA] Así, hay que:
  - Proporcionar subtítulos abiertos o cerrados. Éstos deben representar la información contenida en la pista de audio y deben estar sincronizados con ésta, lo cual requerirá técnicas especiales por tratarse de audio en directo.
- 1.2.5 Se debe proporcionar audiodescripción o de la pista visual. [Nivel AA].
  - Así, hay que:
    - Proporcionar una descripción auditiva de los elementos claves de la pista visual.
    - Dicha descripción auditiva debe transmitir de forma completa toda la información de la pista visual, a saber, información de los personajes, transcripción completa, acciones, lenguaje corporal, contexto y cambios de escena, etc.
    - La descripción auditiva estará sincronizada con la pista visual y la pista sonora.
- 1.2.6 Se proporcionará un contenido alternativo en lengua de signos para toda pista sonora pregrabada. [Nivel AAA].
  - Así, hay que controlar que:
    - Dicho contenido debe estar sincronizado con la pista de audio principal.
- 1.2.7 Se debe proporcionar audiodescripción extendida. [Nivel AAA]
  - Así, hay que:
    - En el caso de que las pausas que se produzcan en la pista de audio principal no sean suficientemente largas como para incluir audiodescripción, se deberá proporcionar audiodescripción extendida como contenido alternativo.
    - Proporcionar una descripción auditiva de los elementos claves de la pista visual.
    - Dicha descripción auditiva debe transmitir de forma completa toda la información de la pista visual, a saber, información de los personajes, transcripción completa, acciones, lenguaje corporal, contexto y cambios de escena, etc..
- 1.2.8 Se debe proporcionar alternativa textual completa para cualquier contenido audiovisual pregrabado. [Nivel AAA].
  - Así, hay que:
    - Transmitir de forma completa toda la información de la pista visual, a saber, información de los personajes, transcripción completa, acciones, lenguaje corporal, contexto y cambios de escena, etc.
    - En el caso de que se produzca algún tipo de interacción con el usuario la alternativa textual contendrá un vínculo que proporcione una funcionalidad interactiva equivalente.
- 1.2.9 Se deben proporcionar contenidos alternativos para cualquier contenido audio en directo.[Nivel AAA]
  - Así, hay que:
    - Proporcionar subtítulos abiertos o cerrados. Estos deben representar la información contenida en la pista de audio y deben estar sincronizados con esta, lo cual requerirá técnicas especiales por tratarse de audio en directo.
    - La transcripción textual completa se considera una posibilidad pero serán preferibles los subtítulos.

### **Técnicas**

Según lo visto, en este primer eslabón, para poder cumplir con el estándar WCAG 1.0, hay que cumplir con los puntos 1.3 y 1.4 de prioridad uno (es decir, para que el sitio web fuera nivel A, por ejemplo, se debería cumplir estos puntos necesariamente). Pasamos a continuación a analizar las técnicas, herramientas y guías útiles para el diseñador. Debido a que el estándar WCAG 1.0 está obsoleto respecto a la tecnología actual, nos referiremos a la documentación de las Técnicas de WCAG 2.0. Hay que destacar que respecto a las WCAG 1.0 se ha producido un cambio de enfoque en las Técnicas ofrecidas en la versión 2.0 de WCAG; además es más práctica y útil para el diseñador aportando recursos, ejemplos y procedimientos de revisión del criterio de éxito en cada caso. Así, se recomienda familiarizarse con esta documentación, y de forma más específica con las técnicas WCAG 2.0 relacionadas y Técnicas de SMIL (*Synchronized Multimedia Integration Language*) estándar de presentación multimedia desarrollado por el W3C indicadas en la segunda columna de la tabla 6, en relación a cada uno de los criterios de éxito aplicables en la accesibilidad de los contenidos audiovisuales en la Web.

Criterios de éxito del WCAG 2.0	Técnicas del WCAG 2.0 y SMIL
1.1.1 Non-text Content	G68
1.2.1 Audio-only and Video-only (Prerecorded)	G158, G159
1.2.2 Captions (Prerecorded)	G93, G87, SM11, SM12
1.2.3 Audio Description or Full Text Alternative:	G69, G78
1.2.4 Captions (Live)	G9, G93
1.2.5 Audio Description	G78, SM6, SM7
1.2.6 Sign Language	G54, G81, SM13, SM14
1.2.7 Audio Description (ext.)	G8, SM1, SM2
1.2.8 Alternative Full Text	G69, G159
1.2.9 Live Audio-only	G150, G157

Tabla 8: Técnicas del WCAG 2.0 y SMIL para la pauta 1.2

### **Herramientas de autor**

Para elaborar contenidos alternativos se necesitan herramientas de autor. Existe un gran número de herramientas orientadas al desarrollo y soporte de material multimedia audiovisual en Web [Moreno L. *et al*, 2006]. Es posible clasificar estas herramientas a través de muchos criterios que van desde la plataforma en que están desarrolladas o el sistema operativo bajo el que funcionan, pasando por el grado de usabilidad de la herramienta o el grado de accesibilidad del producto final. Asimismo, muchas de estas herramientas nos permiten modificar el material multimedia hasta hacerlo accesible [NCAM, 2006]. De esta forma, se pueden crear contenidos audiovisuales con herramientas software de autor que integren subtítulo y/o audiodescripción, o se pueden editar para incluir subtítulo en material audiovisual pregrabado. De la gran variedad de tecnologías existentes, haciendo una revisión y basándonos en lo relacionado a tecnología accesible para contenidos multimedia nos encontramos con:

- Lenguajes y formatos para sincronizar, fundamentales para conseguir la accesibilidad. Destacamos QuickTime [Apple, 2011 b], SMIL [W3C, 2008 c], SAMI [Microsoft, 2003] o Timed Text [W3C, 2006 a].
- Reproductores como Real Media [Realnetworks, 2008], QuickTime [Apple, 2011 b], Windows Media [Microsoft, 2008], etc.

- Editores de subtítulo y/o audiodescripción para contenidos audiovisuales como MAGpie [NCAM, 2003], Hi-Caption Studio [Hi Software, 2011], etc. o utilidades como Captionmenow de IBM [IBM, 2005], etc.
- Reproductores para contenidos multimedia realizados con Flash accesible de Adobe [Adobe, 2008], [Webaim, 2006] y utilizado por muchos diseñadores.

Todas estas posibilidades a veces no son compatibles unas con otras. Se trata de un mare magnum de formatos, plataformas, reproductores, lenguajes y tecnologías distintas que hacen del ejercicio de hacer un contenido multimedia accesible algo complicado, pero no imposible. Es importante que se cumplan y sigan los estándares y recomendaciones del W3C, como por ejemplo, que los navegadores y reproductores multimedia cumplan las pautas de accesibilidad para agentes de usuario (UAAG).

### ***Buenas prácticas para elaborar contenidos alternativos***

En este apartado se muestran buenas prácticas a la hora de elaborar contenidos alternativos como subtítulos, audiodescripción y lengua de signos.

En cuanto al subtítulo, no hay una normativa específica para su inclusión en Web, pero sí se dispone de documentos que pueden ayudar a su elaboración, como la documentación de buenas prácticas en el subtítulo elaboradas por el CESyA [CESyA, 2006] basadas en la norma AENOR “Subtítulo para personas sordas y personas con discapacidad auditiva. Subtítulo a través del teletexto” [AENOR, 2003]. No obstante, es conveniente mencionar que existen otros documentos que pueden ayudar a complementar la información aquí contenida, entre los que cabe destacar las buenas prácticas para el subtítulo de Joe Clark [Clark, J., 2004].

Sobre la audiodescripción tampoco hay una normativa específica para su inclusión en Web. En España se encuentra reglada su creación por medio de la norma AENOR “Audiodescripción para personas con discapacidad visual. Requisitos para la audiodescripción y elaboración de audioguías” [AENOR, 2005]. Se van a presentar las buenas prácticas de audiodescripción elaboradas por el CESyA [CESyA, 2006] basándose en esta norma.

### ***Buenas prácticas de subtítulo***

Dentro del subtítulo existen una serie de aspectos generales que son válidos tanto para programas grabados como en directo.

La adjudicación de colores a los subtítulos dependiendo de los personajes es una estrategia que facilita el seguimiento de la trama argumental y del contexto sonoro, y permite la accesibilidad a personas con discapacidad auditiva.

En el caso de que el color del carácter coincida con el fondo, se puede enmarcar el subtítulo en un recuadro de color para evitar ese problema. Existen ciertas combinaciones de colores que resultan más legibles, porque producen mayor contraste y menos fatiga visual facilitando la lectura, como el carácter amarillo sobre fondo negro, seguido por el carácter verde y el cian sobre el mismo fondo negro. Por tanto, estos son los colores elegidos para aquellos personajes con más presencia en la obra audiovisual.

La norma establece que los subtítulos deben aparecer en la parte inferior de la pantalla ocupando dos líneas y, excepcionalmente, tres. Además dice que para cada personaje se deben asignar líneas distintas. El texto debe estar centrado respecto a la caja.

A la hora de analizar cuándo introducir y sacar un subtítulo, se observan las pausas naturales que hace el ponente, respetando los criterios gramaticales y las unidades lógicas del discurso, o bien los planos. La norma sigue unas guías básicas en la división de subtítulos consistentes en:

- No separar palabras.
- Separar las frases largas según las conjunciones.
- La propia voz muchas veces marca las separaciones mediante pausas o inflexiones.
- Separar las líneas o subtítulos haciéndolos coincidir con comas y puntos.
- Colocar tres puntos suspensivos al final de subtítulo y otros tres al principio del siguiente.

La velocidad recomendada por profesionales del sector experimentados se establece en torno a unos 12 caracteres por segundo.

Además, la comprensión de los subtítulos se mejora con una adecuada sincronización con el sonido. Esto se consigue gracias a la entrada y salida de subtítulos coincidiendo con el movimiento labial, lo que posibilita un apoyo a aquellas personas que poseen restos auditivos. Las voces en off (las emitidas por personajes que no salen en pantalla) van igualmente subtituladas en el color del personaje que la está realizando.

Existen unas pautas particulares en el subtitulado como son:

- No reproducir las abreviaturas.
- Escribir la numeración con letras del cero al diez y con caracteres arábigos el resto de las cantidades.
- Utilizar mejor los paréntesis que los corchetes.
- Utilizar siglas y acrónimos y las formas cortas de entidades u organismos.
- Evitar las muletillas.
- Utilizar los pronombres siempre que se pueda.
- Utilizar las formas cortas de los nombres de personalidades y cargos.

Los subtítulos, en la medida de lo posible, deben ser literales. Además, no hay que olvidar describir el máximo de efectos sonoros necesarios para un buen seguimiento del argumento (a ser posible en la zona superior de la pantalla). También debe aparecer la información contextual, aunque ésta suele ir entre paréntesis y en la misma línea del subtítulo correspondiente [Moreno, L. et al., 2008 b].

#### ***Buenas prácticas de audiodescripción***

La audiodescripción es un servicio destinado a personas ciegas, tanto totales como con resto de visión, con ceguera congénita o adquirida. Por tanto, completa las necesidades del colectivo de personas con ceguera total, favoreciendo a aquellos con deficiencia visual, además de beneficiar a personas con problemas perceptivos y cognitivos

La norma AENOR UNE 153020:2005 habla de seis requisitos necesarios para audiodescribir:

- Análisis de la obra: no todas las obras audiovisuales permiten una buena audiodescripción. Para saber si una obra puede ser audiodescrita se analizan distintos criterios en un primer visionado, como la existencia de huecos de mensaje para

introducir información, así como la saturación o ausencia de dicha información y que se realice en el mismo idioma de la información sonora de la obra.

- **Confección del guión:** el guión está formado por unidades de información incluidas en los huecos de mensaje. Para que el guión sea coherente, el audiodescriptor debe consultar la documentación referente al entorno y la temática de la obra. Así, la información se adecuará al tipo de obra y a las necesidades del público a la que se dirige. Además, existen otras características a tener cuenta, como la trama de la acción dramática, los ambientes y los datos plásticos. No hay que olvidar que el estilo de escritura debe ser sencillo, fluido, con frases de construcción directa, terminología específica y adjetivos concretos. La norma UNE 153020 asegura que debe aplicarse la regla espacio-temporal [...] respetarse los datos que aporta la imagen [...] no descubrir ni adelantar la trama y evitar transmitir cualquier punto de vista subjetivo.
- **Revisión y corrección del guión:** las correcciones necesarias que se hagan servirán para adecuar el guión a las normas de audiodescripción. La norma asegura que deberían ser revisadas las correcciones por una persona distinta del descriptor e incorporarlas después al guión final.
- **Locución:** se realiza en presencia de la imagen que se describe y el locutor se selecciona según el tipo de voces y el tono adecuado para la obra, de manera que esas voces sean siempre claras para los oyentes. Se debe evitar la entonación afectiva, procurando realizar locuciones neutras.
- **Montaje.** En la mezcla se equiparan los volúmenes, efectos de ambiente y equalizaciones con la B.S.O (banda sonora original).
- **Revisión:** según la norma, una vez finalizada la grabación en el soporte elegido para el caso, debe comprobarse que el producto audiodescrito cumple los requisitos.

### 3.3.2. Segundo Eslabón: El acceso al contenido multimedia debe ser accesible

Pasamos al segundo eslabón de la cadena; como se ha mencionado con anterioridad, no servirá de nada haber realizado un contenido accesible si el acceso al mismo no goza igualmente de accesibilidad.

Básicamente se pueden distinguir dos formas distintas para ofrecer un contenido multimedia audiovisual en Web a un usuario: mediante descarga del contenido en el equipo del usuario o mediante la emisión del contenido por medio de un reproductor incluido en la página web.

#### ***Descarga directa del contenido***

Los problemas con los que se encuentran los desarrolladores web para incluir contenidos audiovisuales de manera accesible y que se verán en el apartado 2.3.2 hacen que la opción de descarga directa sea muy utilizada. En este caso, se proporciona al usuario un enlace desde el que poder descargar el contenido, corriendo la reproducción del mismo por parte del software que el usuario tenga instalado en su propio equipo.

A pesar de ser una buena opción para ofrecer multimedia accesible, la descarga directa hay que realizarla teniendo en cuenta que se va a realizar un almacenamiento de información en el equipo del usuario y por lo tanto hay que avisarle claramente de este hecho. Esta situación en las pautas de accesibilidad al contenido web 1.0 (WCAG 1.0) quedaría recogida en la pauta 13.1

que dice: “Identifique claramente el objetivo de cada vínculo” [prioridad 2]. Y en las WCAG 2.1 quedaría recogida en los criterios de éxito 2.4.4 [nivel A] y 2.4.9 [Nivel AAA].

Por otra parte hay que tener en cuenta que las alternativas generadas como son el subtítulo y la audiodescripción pueden venir integradas en el vídeo (“open”) o haber sido generadas como elementos independientes (“closed”) para poderlos activar en función de las necesidades del usuario. En este último caso, será necesario proporcionar todos los elementos en la descarga de forma que el usuario pueda seleccionar en su propio equipo lo que desea reproducir.

Por consiguiente, si se ofrecen contenidos audiovisuales por descarga directa, no sólo habrá que cumplir las pautas vistas en el apartado 2.2, sino que habrá que ser cuidadoso con el texto del enlace con el que se activa la descarga y su etiquetado, de lo contrario no se alcanzaría el nivel AA de adecuación de las WCAG 1.0, ni de las WCAG 2.0.

#### ***Emisión del contenido. Técnicas***

Pese a que la descarga directa suponga una forma sencilla de proporcionar al usuario el acceso a un contenido audiovisual accesible, la difusión del contenido a través de un reproductor integrándolo en la propia página web es la forma más utilizada.

Ello es debido a que es menos complejo para los usuarios, por ser una opción más usable al acceder a ella por medio de la navegación. Al integrarlo en la Web podemos realizar la emisión del vídeo de dos formas distintas:

- **Emisión en descarga progresiva (falso *streaming*):** con este tipo de emisión el contenido se va descargando en el equipo del usuario y cuando alcanza un porcentaje determinado de descarga se empieza a reproducir. Este tipo de emisión supone una gran ventaja respecto a la descarga ya que no es necesario que el archivo se haya almacenado por completo para iniciar su reproducción. Sin embargo, hasta que el contenido no se ha descargado en su totalidad no se permite hacer uso de los controles de reproducción (reproducir, parar, pausa, etc.) en su totalidad.
- ***Streaming*:** es una descarga del contenido bajo demanda; este tipo de emisión tiene la ventaja de que el usuario puede manejar completamente los controles de reproducción desde un principio y además no es necesario almacenar el vídeo en el equipo del usuario.

Para estas opciones, hay dos maneras de implementación y ambas están asociadas a un reproductor: la primera es activar la descarga con un enlace, forma más frecuente en la opción de *streaming*, y la segunda integrando el contenido audiovisual a través del elemento <object> del XHTML. En cualquiera de los dos casos, hay que cumplir con las WCAG.

Para la opción de activación del enlace, al igual que se ha comentado en la opción anterior de descarga directa, es necesario etiquetar correctamente dicho enlace para cumplir con la pauta 13.1 de las WCAG 1.0, o los criterios de éxito 2.4.4 Y 2.4.9 de las WCA 2.0.

**3.3.3. Tercer eslabón:** Hay que ofrecer alternativas atendiendo a preferencias del usuario y la interacción del usuario al acceder a contenido debe ser usable

### Capítulo 3: Accesibilidad Web

Por último, el acceso del usuario al vídeo debe ser además de posible, intuitivo en la interacción por lo que hay que proporcionar accesibilidad en la forma de mostrar el acceso y control de la información por el usuario. De esta manera el punto de verificación 14.1 (*Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio*. [Prioridad 1]) se debe cumplir de acuerdo a la WCAG 1.0.

Como técnicas para el punto 14.1 recomendamos seguir las técnicas de Diseño Centrado en Usuario [Henry S., 2007], donde el usuario participe en el diseño y en la elaboración de los contenidos así como la aplicación de Reglas de Lectura Fácil; estas técnicas se identificarán como Técnicas del Lenguaje en la tabla resumen mostrada en el siguiente apartado.

Además, los usuarios deberían poder tener acceso a la reproducción del vídeo, de acuerdo a sus características y preferencias. Hay factores que deben tenerse en cuenta, tales como el tamaño del vídeo, la duración del vídeo, la información de progreso en la reproducción, la velocidad y el tipo de conexión del usuario, el agente de usuario asociado a la reproducción, el formato del vídeo, si el usuario va a tener el control, etc. No todos estos factores son siempre determinantes, dependerá de la modalidad que haya sido elegida al servir el contenido.

Por ejemplo, la información del tamaño del vídeo será importante para el usuario en las opciones de descarga o descarga progresiva, pero no es necesario en streaming.

En función de estos tipos, las posibles características a considerar son las mostradas en la tabla 9.

Características	Descarga directa	Descarga progresiva	Streaming
<b>Tamaño del recurso</b>	SI	SI	NO
<b>Duración del contenido audiovisual</b>	SI	SI	SI
<b>Opciones de velocidad y tipo de conexión para preferencias del usuario</b>	SI*	SI*	SI*
<b>Barra de progreso</b>	NO	SI	NO
<b>Agente/s de usuario asociados a la reproducción</b>	SI	SI	SI
<b>Formatos (Windows media, Real, QuickTime, SMIL)</b>	SI	SI	SI
<b>Se permite control al usuario</b>	NO*	*	NO
<b>*A excepción de SMIL</b>			

Tabla 9: Información a presentar al usuario

Esta situación queda recogida en el punto de verificación 11.3 (Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (Por ejemplo, idioma, tipo de contenido [Prioridad 3]) de WCAG 1.0.

Aparte de ofrecer toda la información al usuario comentada anteriormente, es importante considerar cómo hay que presentar esa información al usuario. Como Técnicas para el punto 11.3 está la aplicación de los principios de Diseño Universal [Shneiderman, B, 2000] y criterios de usabilidad en el interfaz web donde se incluya el contenido multimedia y sus características de acceso. Denominaremos a dichas técnicas "Técnicas de Diseño Universal", para identificarlas en la tabla resumen ofrecido en el siguiente apartado. Puede ser muy conveniente contar en el diseño y desarrollo con una revisión experta humana de usabilidad, además de una validación por usuarios para garantizar que el usuario puede acceder con facilidad al contenido.

### 3.3.4. Tabla resumen (WCAG 1.0|WCAG 2.0) de la accesibilidad a los contenidos audiovisuales en la web

Como soporte para diseñadores y desarrolladores que quieran incluir contenidos audiovisuales en la Web, considerando la accesibilidad de una manera completa, teniendo en cuenta los tres eslabones de la cadena vistos, se presenta a continuación en la tabla 10 una tabla resumen, que incluye por cada eslabón, los puntos de verificación de las WCAG 1.0 y los criterios de éxito de las WCAG 2.0 relacionados junto con sus técnicas como ayuda.

La cadena de la accesibilidad de los contenidos audiovisuales en la Web		WCAG 1.0	WCAG 1.0	WCAG 2.0	WCAG 2.0	Técnicas WCAG 2.0, SMIL y otras
Listado de eslabones		Puntos de verificación	Prioridad	Criterios de Éxito	Nivel de Adecuación	
<b>(1) El contenido en sí mismo debe ser accesible</b>		1.1	1	1.1.1 Non-text Content	A	G68
				1.2.1 Audio-only y video-only (Prerecordes)	A	G158, G159
				1.2.9 Live Audio-only	AAA	G150, G157
		1.3	1	1.2.5 Audio Description	AA	G78, SM6, SM7
				1.2.7 Audio Description (ext.)	AAA	G8, SM1, SM2
		1.4	1	1.2.2 Captions (Prerecorded)	A	G93, G87, SM1, SM2
1.2.3 Audio Description or FullText Alternative	A			G69, G78		
1.2.4 Captions (Live)	AA			G9, G93		
1.2.8 Alternative Full Text	AAA			G69, G159		
		No hay correspondencia		1.2.6 Sign Language	AAA	G54, G81, SM13, SM14
<b>(2) El acceso al contenido debe ser accesible</b>	Descarga completa	13.1	2	2.4.4 Link Purpose (In Context)	A	G91, G53
	Descarga progresiva streaming			2.4.9 Link Purpose (Link Only)	AAA	G91, C7
	Descarga progresiva streaming	6.3	1	No hay correspondencia directa	A	H46, H53, Técnicas objeto
		3.2	2	4.1.1 Parsing	A	G134, G154
		No hay correspondencia		4.1.2 Name, Role, Value	A	G10, G108, 135
<b>(3) Hay que ofrecer alternativas atendiendo a las preferencias del usuario; además, la interacción del usuario al acceder al contenido debe ser usable</b>		14.1	1	No hay correspondencia directa pero considerar: 3.1.5 Reading Level	AAA	G86, G103, G79 Técnicas Lenguaje
		11.3	3	No hay correspondencia directa pero considerar: 2.2.2 Pause, Stop, Hide	A	G4, G11 Técnicas Diseño Universal

Tabla 10: Tabla resumen para diseñadores sobre cómo ofrecer accesibilidad a los contenidos audiovisuales en la Web [Moreno L. et al., 2008 a]



# Capítulo 4

## 4. Accesibilidad en elementos relativos al contenido multimedia en HTML5

En este capítulo se va a explicar qué elementos de HTML5 están relacionados con la accesibilidad y el contenido multimedia tan extendido actualmente. Desde el video, pasando por el soporte de subtítulos, requisito de accesibilidad en los contenidos multimedia tal como se ha visto en el capítulo anterior. Como se mencionará más adelante no hay soporte de subtítulo hasta el momento para incluirlo directamente con HTML5, pero sí existen soluciones a integrar de otras tecnologías tal como se ha utilizado en este proyecto, y que se verán en el capítulo cinco.

### 4.1. Elementos semánticos

HTML5 introduce varios elementos semánticos que representan secciones lógicas o componentes de una aplicación web o un documento: `<section>`, `<nav>`, `<article>`, `<aside>`, `<hgroup>`, `<header>`, `<footer>`, así como nuevas reglas para el uso de las etiquetas de cabecera `<h1>`-`<h6>` y los elementos `<dirección>`. Las nuevas etiquetas proporcionan una forma más sencilla para los desarrolladores a la hora de definir las distintas partes de un documento.

En general, todo esto tiene un importante potencial para mejorar la accesibilidad, sobre todo si los lectores de pantalla y otros productos de apoyo comienzan a utilizar la información asociada.

### 4.1.1. Uso exclusivo de <h1> en <section> y <article>

En las versiones actuales de HTML, la única forma de definir las secciones y la jerarquía del esquema de un documento es el uso de la <h1>-<h6> en una estructura plana. Esto es muy beneficioso para la accesibilidad, pero puede ser un poco incómodo para los desarrolladores de páginas web. HTML5 introduce dos nuevos elementos, <section> y <article>, que definen secciones lógicas y artículos para organizar el contenido y los niveles de jerarquía de lo mismo.

Esta solución puede originar algunos problemas de accesibilidad durante la fase de transición hacia el estándar, ya que es posible que se sigan usando las mismas técnicas anteriores de HTML4.01 en HTML5 siendo esto innecesario, pero haciendo posible que continuemos usando etiquetas de encabezados aunque estén anidadas dentro de bloques <section> y <article>.

### 4.1.2. El elemento <nav>

El elemento <nav> en HTML5 proporciona una forma de especificar de forma explícita los elementos de navegación en las páginas. Al igual que antes, esto será realmente destacable cuando los productos de apoyo (y los navegadores web estándar) ofrezcan e a los usuarios la capacidad de tomar ventaja del estándar directamente.

### 4.1.3. Nuevos tipos para la etiqueta <input> y sus extensiones

HTML5 define 13 nuevos valores para el elemento <input> (búsqueda, teléfono, url, correo electrónico, fecha y hora, fecha, mes, semana, tiempo, fecha y hora local, número, intervalo y color).

Si bien queda por ver cómo cada uno de los nuevos tipos de entrada se representará o será utilizada por los navegadores web. Estos son campos de texto utilizados por secuencias de comandos y se enviarán por GET y POST como tal. En algunos casos (como la fecha y la hora) HTML5 dicta que los datos se almacenarán internamente o serán representados de manera específica, de forma que los cambios en su manera de ser utilizados será mínimos.

Por defecto, para un elemento <input> en HTML se le asigna un type = "text", así que éste es un cambio perfectamente compatible con versiones anteriores, y no debería causar ningún problema (de accesibilidad o de otro tipo).

### 4.1.4. Tipo de entrada Search

El tipo de entrada de búsqueda representa un campo de texto que funciona como una caja de búsqueda. Respecto a este nuevo tipo su utilidad y los beneficios para la accesibilidad son bastante grandes, ya que permitiría a las tecnologías de asistencia proporcionar, mediante un solo comando o pulsaciones de teclas, acceder a la casilla de búsqueda de la página actual, independientemente de dónde se encuentre en el contenido.

#### 4.1.5. Tipos de entrada tel, url y email

Mediante estos nuevos tipos, el desarrollador tiene una ayuda para la validación de entrada de los datos, de forma que se puede exigir que el texto introducido cumpla las especificaciones del formato URL o de direcciones de correo. Además, existe la posibilidad de integración con otras fuentes de datos, mediante lo cual, por ejemplo, los agentes de usuario podrían proporcionar un mecanismo para obtener direcciones de email o números de teléfono de un navegador o de la libreta de direcciones y cumplimentar de esta forma los campos de entrada.

La accesibilidad de estas interacciones basadas en el navegador será responsabilidad de los proveedores de navegadores.

#### 4.1.6. Extensiones para la etiqueta <input>

Además de los nuevos tipos <input>, HTML5 también añade cuatro nuevos atributos a todos los elementos de entrada que tienen un impacto potencial de la accesibilidad: autofocus, placeholder, required y pattern. Todos ellos implementan características y funciones que se han estado utilizando en aplicaciones web desde hace años a través de secuencias de comandos, pero ahora el estándar nos permite ser codificadas directamente en HTML.

Los navegadores, exceptuando Internet Explorer, admiten estas extensiones en mayor o menor medida, pero se espera que las implementen en todas sus funcionalidades en breve. Mientras tanto, las soluciones basadas en scripts se puede utilizar junto con los atributos de HTML5, con la incorporación de algunas utilidades para determinar el user-agent del navegador del usuario.

El atributo de enfoque automático permite a los desarrolladores especificar que un elemento, de forma particular, debe recibir el foco de entrada tan pronto como la página que lo contiene se carga, de modo que puede empezar a escribir de inmediato sin tener que hacer click específicamente en ese control. Las capacidades de autofoco basado en scripts son muy utilizadas en la actualidad, pero conllevan una serie de problemas, incluyendo el hecho de que no hay forma de deshabilitarlas. Con el atributo autofocus los navegadores web y otras aplicaciones de usuario puede permitir la capacidad de deshabilitar esta característica, ya sea en un sitio web específico o para toda la Web.

Hay que tener en cuenta que existen algunos problemas de acceso con el enfoque automático, y esto no cambiará con el nuevo estándar, pero en los casos los que queramos utilizar esta característica, el atributo de enfoque automático en HTML5 proporciona una manera más fácil y más simple realizarlo.

### 4.1.7. Marcador de posición de texto (placeholder)

El marcador de posición de texto en los campos de entrada en HTML está considerado como un requisito de accesibilidad. Actualmente se implementa con un texto por defecto que contiene la entrada o el área de texto de los formularios, siendo, este texto, eliminado automáticamente por un script cuando se incluye texto por parte del usuario.

El atributo placeholder de HTML5 permite a los desarrolladores web especificar el marcador de posición del texto directamente en el código del formulario. Esto ahorra el desarrollador el problema de tener que utilizar una solución basada en scripts, y permite a los proveedores de navegadores incluir tecnologías para ayuda de una manera consistente. Además, el uso generalizado del atributo (en lugar de secuencias de comandos existentes) nos asegura que el texto del marcador de posición tendrá un aspecto y comportamiento, a lo largo de la Web, uniforme, incrementando la accesibilidad y facilitando el aprendizaje de uso de los sitios web.

Hay que tener en cuenta que está explícitamente prohibido por la especificación HTML5 dejar sin etiquetas explícitas los marcadores de posición de texto.

### 4.1.8. Campos requeridos (required)

La capacidad de indicar que un campo es requerido ha sido posible con ARIA desde hace algún tiempo, e incluso se apoyó en los navegadores más modernos y los lectores de pantalla. Esta solución es mucho mejor que la comúnmente utilizada de “añadir un asterisco al texto de la etiqueta” ya que permite a los agentes de usuario y tecnologías de apoyo informar al usuario de que un campo en particular se requiere. HTML5 va un paso más allá añadiendo una característica directamente a la sintaxis HTML (no requiere ARIA), con la idea de que el navegador no sólo mostrará los elementos requeridos de forma diferente, sino que proporcionará automáticamente los mensajes de error (en una forma accesible, suponemos) cuando alguien intenta enviar un formulario sin haber llenado en todos los valores requeridos.

#### 4.1.9. Patrones (pattern)

El atributo `pattern` de HTML5 incrementa las capacidades del atributo `required` al indicar que un campo de entrada particular debe tener un valor que coincide con una cierta expresión regular definida por el desarrollador. Esto permite recibir los datos estrictamente con un formato predefinido (por ejemplo, en el caso de los números de tarjetas de crédito) de forma que, la validación automática del lado del cliente, permite reportar errores.

La especificación exige que, cuando se utiliza el atributo patrón, también se añada un título que explique los detalles del modelo que se verificará.

Si este atributo no nos cubre todas las necesidades de nuestra aplicación, HTML5 proporciona una interfaz para acceder a las funciones de validación de formularios a través de secuencias de comandos.

# Capítulo 5

## 5. Diseño y desarrollo de interfaz basado en HTML5

En este capítulo se va a proceder a la explicación y descripción del diseño e implementación de una página web en HTML5.

El primer paso fue hacer el diseño de la presentación abstracta, es decir definir la estructura lógica de la página web. La implementación fue sencilla gracias a las nuevas etiquetas. HTML5 resulta muy intuitivo ya que estas etiquetas tienen nombres representativos para lo que están destinadas. Para ello se estudiaron las nuevas etiquetas principales, en el que se seleccionaron las etiquetas más importantes para dar forma al HTML, que son `<header>`, `<nav>`, `<section>`, `<article>`, `<aside>` y `<footer>`.

A continuación se hizo el diseño de la presentación concreta, y su implementación con CSS3. Así, se ha obtenido el aspecto gráfico deseado mediante las técnicas habituales de estilo, ya que el efecto de CSS es el mismo en HTML5 que en HTML4, con la mínima diferencia que ahora como las etiquetas principales tienen nombres muy representativos, se pueden dar estilos más cómodamente que antes.

Una vez implementado la interfaz en HTML5 y CSS3, se paso a implementar las funciones necesarias para la reproducción de imágenes agregar fotos, y por supuesto que se observa en los dispositivos. Como tecnologías ha sido necesario incorporar tecnología JavaScript, J query para la incorporación de diversos controles necesarios según requisitos de diseño.

## 5.1 Elementos de HTML5 utilizados

Para la realización de esta página se han usado muchos elementos nuevos introducidos en HTML5, pero también bastantes tomados de la anterior versión HTML4.01, ya que hay numerosos elementos que no se han eliminado y hasta ahora son útiles para ciertos casos.

Etiquetas nuevas usadas:

- `<header>`: artículo introductorio para artículos o navegación, utilizados tanto para la página web, como para cualquier esquema diseñado como introducción a algo.
- `<nav>`: diseñado para crear barras de navegación muy comunes en páginas web, donde normalmente figuran las distintas secciones o menús de un sitio web.
- `<section>`: representa una sección genérica de un documento o una aplicación. Podría agrupar contenido de un tema específico dentro de un artículo en que se pueda introducir.
- `<video>`: nueva etiqueta que permite introducir un vídeo dentro de una página web sin necesidad de software adicional.
- `<audio>`: nueva etiqueta que permite introducir una pista de audio dentro de una página web sin necesidad de software adicional.
- `<button>`: etiqueta ideada para introducir un botón con el fin de crear una acción o evento.
- `<source>`: se usa principalmente en las etiquetas `<video>` y `<audio>` para indicar la dirección donde se encuentra el archivo con el que se quiera interactuar.
- `<aside>`: división lógica de un apartado independiente dentro de la página relacionado con el contenido que lo rodea. Se puede usar como una sección vertical de información adicional al contenido de la página.
- `<footer>`: representa el pie de una sección, con información acerca de la página/sección que poco tiene que ver con el contenido de la página.
- `<article>`: representa un componente de la página o sitio web, con la intención de que pueda ser reutilizado y repetido.
- `<form>`: sección especialmente creada para introducir etiquetas de formularios.

Etiquetas usadas ya permitidas en HTML4.01:

- `<head>`: sección inicial donde se introduce el título de la página y los archivos necesarios de CSS y JavaScript.
- `<body>`: sección principal de un archivo HTML donde se introduce el grueso de la información que se necesita para la construcción de una página web.
- `<title>`: en la etiqueta `<head>` indica el nombre de la página web, y si se introduce en otras etiquetas como atributo, sirve para mostrar un pequeño texto encima del puntero del ratón.
- `<link>`: etiqueta para enlazar archivos CSS en la página web.
- `<script>`: etiqueta para enlazar archivos JavaScript en la página web.
- `<h1>`, `<h2>`, `<h3>`: etiqueta para mostrar, normalmente, cabeceras de texto en cualquier sección de la página web.
- `<p>`: define principalmente un párrafo de texto o elementos en html.

- <div>: etiqueta para dividir secciones lógicas dentro de una página web.
- <span>: define una sección cualquiera de un documento.
- <ul>: etiqueta el inicio de una lista sin orden.
- <li>: define uno de los elementos de una lista sin orden.
- <a>: normalmente se utiliza para señalar un hipervínculo.
- <label>: define un campo para formularios.
- <input>: define una entrada normalmente para formularios.

### 5.1.1. Estructura lógica de la página creada en HTML5

A continuación se explicará detalladamente cada sección de la que está formada la página creada para este proyecto, en la que se aplican los conocimientos que se han obtenido sobre HTML5 y sus elementos y etiquetas a lo largo del documento.

#### ***Cabecera***

En esta sección introduciremos los enlaces relacionados con fichero CSS3 y JavaScript necesarios para el correcto funcionamiento de la página, que se encargará de dar formato gráfico y estructura a los elementos del fichero en HTML5, y responsable de que el contenido multimedia incluido en este proyecto se visualice correctamente, respectivamente.

Claramente se aprecia que esto es idéntico a anteriores versiones de HTML5 y que es imprescindible para su correcta visualización. El código responsable de lo comentado anteriormente se puede ver en la figura 23.

```

<!DOCTYPE html>
<!--[if IE 8 ]><html class="ie ie8" lang="en"> <![endif]-->
<!--[if (gte IE 9) || !(IE)]><!--><html lang="en"> <!--<![endif]-->
</html>
<head>

<!-- Basic Page Needs
===== -->
<meta charset="utf-8">
<title>WifiTickets</title>
<link type="image/x-icon" href="images/favicon.ico" rel="icon">
<!-- Mobile Specific Metas
===== -->
<meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1">

<!-- CSS
===== -->
<link rel="stylesheet" href="css/style.css">
<link rel="stylesheet" href="css/colors/blue.css" id="colors">

<!--[if lt IE 9]>
    <script
src="http://html5shim.googlecode.com/svn/trunk/html5.js"></scri
pt>
<![endif]-->

<!-- Java Script
===== -->
<script src="http://code.jquery.com/jquery-
1.11.0.min.js"></script>
<script src="http://code.jquery.com/jquery-migrate-
1.2.1.min.js"></script>
<script src="scripts/jquery.flexslider.js"></script>
<script src="scripts/jquery.selectnav.js"></script>
<script src="scripts/jquery.easing.1.3.js"></script>
<script src="scripts/jquery.contact.js"></script>
<script src="scripts/jquery.isotope.min.js"></script>
<script src="scripts/jquery.jcarousel.js"></script>
<script src="scripts/jquery.fancybox.min.js"></script>
<script src="scripts/jquery.layerslider-
transitions.js"></script>
<script src="scripts/jquery.layerslider.min.js"></script>
<script src="scripts/jquery.shop.js"></script>
<script src="scripts/custom.js"></script>

</head>

```

Figura 23: Cabecera de la página HTML5

### ***Navegación y título***

La sección navegación contendrá una barra de navegación horizontal en la que se mostrará un menú con los que se podrá dirigir a las distintas secciones de la página, y el título para mostrar con una frase el nombre de la página.

Para ello, simplemente se usarán dos etiquetas nuevas de HTML5, <header> para mostrar el título de la página mediante una etiqueta de encabezado <h1>, y <nav> para mostrar el submenú inicial de la página mediante las etiquetas de listas sin orden ya presentes en anteriores versiones de HTML, y la cual se muestra en la figura 24.

```

<header id="header">

    <!-- Logo -->
    <div class="ten columns">
        <div id="logo">
            <h1><a href="#"></a></h1>
            <div id="tagline">Comparta su red Wi-Fi
de la manera m&aacute;s sencilla</div>
            <div class="clearfix"></div>
        </div>
    </div>

    <!-- Social / Contact -->
    <div class="six columns">

        <!-- Social Icons -->
        <ul class="social-icons">
            <li class="twitter"><a
href="http://www.twitter.com/wifitickets"
target="twitter">Twitter</a></li>
            <li class="facebook"><a
href="https://www.facebook.com/wifitickets" target="facebook"
>Facebook</a></li>
            <li class="youtube"><a
href="http://www.youtube.com/channel/UCvRKFoEfjDAWMhTgumPtR-w"
target="youtube">Youtube</a></li>
            <li class="linkedin"><a
href="https://www.linkedin.com/company/wifitickets"
target="linkedin">LinkedIn</a></li>
        </ul>

        <div class="clearfix"></div>

        <!-- Contact Details -->
        <div class="contact-details">Correo:
Francesc.Burrull@upct.es</div>

        <div class="clearfix"></div>

        <!-- Search -->
        <nav class="top-search">
            <form action="http://www.google.es"
target="google" method="get">
                <button class="search-
btn"></button>
                <input class="search-field"
type="text" onblur="if(this.value=='')this.value='Buscar';"
onfocus="if(this.value=='Buscar')this.value='';"
value="Buscar" />
            </form>
        </nav>

```

Figura 24: Código HTML5 para la cabecera

Con esto conseguiríamos una lista de elementos independientes que posteriormente daremos formato y estructura para reordenarlo horizontalmente, aplicando estilos a través de CSS3 como se puede observar en la figura 25.

```

/* Header
===== */
#top-line {
    width: 100%;
    background: #169fe6;
    height: 5px;
    display: block;
}

#header { min-height: 130px; }

.contact-details {
    float: right;
    margin: -1px 0 1px 0;
    display: block;
    color: #888;
    font-size: 11px;
}

.top-search {
    float: right;
}

.top-search input {
    width: 180px;
    font-size: 11px;
    color: #888;
}

.search-btn {
    float: right;
    position: relative;
    padding: 0;
    top: 0;
    right: 0;
    margin: 1px 0 0 -35px;
    background: url('../images/search.png') center center no-repeat;
    width: 35px;
    height: 32px;
    cursor: pointer;
    border: none;
    z-index: 100;
    box-shadow: none;
}

```

Figura 25: Código CSS pceraara cab

Con este código CSS3 conseguiríamos reordenar los elementos de la lista sin orden `<ul>` horizontalmente, dando el aspecto de menú rápido.

El resultado final sería el mostrado en la figura 26:



Figura 26: Resultado final de título

```
<!-- Navigation
===== -->
<nav id="navigation" class="style-1">

<div class="left-corner"></div>
<div class="right-corner"></div>

<ul class="menu" id="responsive">

    <li><a href="inicio.html" id="current"><i
class="halflings white home"></i> Principal</a></li>

    <li><a href="quees.html"><i class="halflings white
file"></i>Qu&eacute; es </a></li>

    <li><a href="Galeria.html"><i class="halflings white
picture"></i>Galeria</a></li>

    <li><a href="#"><i class="halflings white globe"></i>
Idiomas</a>
    <!-- Second Level / Start -->
    <ul>
        <li><a href="index.html">2
Ingl&eacute;s</a></li>
        <li><a href="index_fr.html">3
Franc&eacute;s</a></li>
        <li><a href="index_ar.html">4
&Aacuterabe</a></li>
    </ul>
    <!-- Second Level / End -->
</li>

    <li><a href="Tienda.html"><i class="halflings white
shopping-cart"></i> Tienda</a></li>

    <li><a href="contacto.html"><i class="halflings white
envelope"></i> Contacto</a></li>
```

Figura 27: Código HTML para navegación

Cómo se aprecia en el código anterior la imagen no está en HTML, sino que por comodidad se decidió referenciarla a través de CSS para poder añadir atributos y características propias diseñadas para la página.

```
#navigation {
    height: 49px;
    width: 1040px;
    margin-left: -10px;
    background-color: #4c4c4c;
    float: left;
    left: 1px;
}

/* Style Without Shadows */
.style-2 .left-corner,
.style-2 .right-corner { display: none; }
.style-2 { width: 1020px !important; margin-left: 0 !important; }
.style-2 ul li:first-child { margin-left: 1px !important; }

.js .selectnav {
    display: none;
}

.menu i.halflings { margin: 1px 2px 0 -2px; }

.menu ul, .menu li > div {
    visibility: hidden;
    display: none;
}

.menu li:hover > ul, .menu li:hover > div {
    visibility: visible;
    display: block;
}
```

Figura 28: Código CSS para la navegación

El resultado final sería el mostrado en la figura 29:



Figura 29: Imagen de la navegación

```

<!-- LayerSlider -->
<section id="layerslider-container">
  <div id="layerslider" style="width: 1020px; height:
450px; margin: 0 auto;">

      <!-- Slide 1 -->
      <article class="ls-layer" style="slidedelay: 10000;
slidedirection: top;">
        
          
          <h3 class="ls-s1 caption-transparent"
style="top: 383px; left: 105px;">Pulsar_Buton</h3>
          
          
          <h3 class="ls-s1 caption-transparent"
style="top: 383px; left: 435px;">Sacar_Ticket</h3>
          
          
          <h3 class="ls-s1 caption-transparent"
style="top: 383px; left: 745px;">Navigar_Internet</h3>
        </article>

      <!-- Slide 2-->

      <article class="ls-layer" style="slidedelay: 10000;
slidedirection: top;">
        
          
        </article>

```

*Figura 30: Código de HTML usando código JQuery para animar las fotos*