

Creación de páginas Web accesibles con HTML5

Teresa Diez¹, M^a José Domínguez¹, José Javier Martínez¹,
Jon Joseba Sáenz de Navarrete¹

¹ Departamento de Ciencias de la Computación,
Universidad de Alcalá, España
{teresa.diez, mariajose.dominguez, josej.martinez}@uah.es
jjoseba@gmail.com

Resumen. La importancia y el alcance mundial que Internet tiene hoy en día en nuestra sociedad hace imprescindible que se garantice su uso de forma universal, de ahí la importancia que adquiere la accesibilidad a la hora de diseñar una página o sitio web. El objetivo de este trabajo es exponer las posibilidades que nos ofrece HTML5 a la hora de crear páginas accesibles para todos los usuarios, independientemente de sus problemas visuales, auditivos, sensoriales o de cualquier otro tipo. Comenzaremos por las nuevas etiquetas, semánticas y multimedia, que incorpora este estándar y las posibilidades que aportan para el correcto seguimiento de las Pautas de Accesibilidad al Contenido de la Web (WCAG). Se describen también, brevemente, las nuevas funciones y eventos que facilitan la interactividad en los nuevos diseños web, sin depender del dispositivo utilizado. En definitiva veremos que, a pesar de que se encuentra todavía en desarrollo, HTML5 constituye un prometedor avance hacia el diseño de sitios web que favorezcan la igualdad de condiciones para todos los usuarios de la red.

Palabras clave: Accesibilidad, etiquetas semánticas, estándares, pautas de accesibilidad, WCAG, HTML5

1 Introducción

En un principio el World Wide Web (WWW) fue un medio basado principalmente en texto que utilizaba la comunidad científica, pero en poco tiempo comenzó a utilizarse como plataforma comercial y el diseño y las representaciones gráficas fueron adquiriendo cada vez mayor importancia. Este proceso de crecimiento no se produjo de forma ordenada, los diseñadores y programadores no seguían pautas comunes, lo cual dio lugar a diferentes problemas de uso y acceso a la red para muchas personas. Con el objetivo de asegurar el correcto crecimiento de la red a largo plazo, en 1994 se creó el consorcio internacional World Wide Web (W3C) que trabaja conjuntamente en el desarrollo de estándares web que sirvan de guía a desarrolladores y diseñadores. Dentro del mismo surgió la Web Accessibility Initiative (WAI), que ofrece normas y recomendaciones para garantizar la accesibilidad en la Web.

Según se define en el propio W3C [6], la accesibilidad es *el acceso de todos a la Web, independientemente del tipo de hardware, software, infraestructura de red, idioma, cultura, localización geográfica, y capacidades de los usuarios.*

En este sentido, en la Convención de los Derechos de las Personas con Discapacidad de las Naciones Unidas [14] se habla de la necesidad de *garantizar a las personas con discapacidad el acceso a la información y las comunicaciones, en igualdad de condiciones, incluida la información y tecnologías de comunicación y sistemas.*

La accesibilidad se convierte por tanto en algo indispensable, además de que una página web tenga un diseño de interfaz atractivo, su contenido debe ser accesible para cualquier tipo de persona.

En este momento el acceso universal es todavía una utopía y muchos usuarios encuentran diferentes barreras a la hora de acceder a la web y a los servicios que ofrece. De igual forma que un arquitecto debe considerar las barreras arquitectónicas a la hora de diseñar un edificio, los desarrolladores deben tener en cuenta la accesibilidad desde las primeras fases del desarrollo de un sitio web.

1.1 Accesibilidad en el diseño Web

Además de difundir los principios de la accesibilidad web a las empresas, gobiernos, instituciones, desarrolladores y usuarios, la iniciativa WAI es la encargada de la redacción y publicación de las *Pautas de Accesibilidad al Contenido de la Web* (WCAG) que se han convertido en un estándar internacional.

La primera versión (WCAG 1.0) [3], publicada en 1999, contiene catorce pautas básicas relacionadas con la accesibilidad y una serie de puntos de revisión que pueden servir como base para comprobar la accesibilidad de las páginas web. No obstante el W3C recomienda que los nuevos contenidos o sitios web se desarrollen en base a la versión WCAG 2.0 [4], publicada en 2008 con una nueva organización y estructura que facilita su uso por parte de personal tanto técnico como ejecutivo.

Se basan en cuatro principios fundamentales, perceptible, operable, comprensible y robusto, bajo los que se establecen las doce pautas, que proporcionan objetivos básicos sobre los que se debe trabajar para ofrecer un contenido más accesible a usuarios con distintos tipos de discapacidad.

Para cada una de las pautas se articulan Criterios de Conformidad comprobables mediante una gran variedad de técnicas que facilitarán su cumplimiento. Junto a estas técnicas, cuando se encuentran documentados, también se proporcionan fallos comunes.

Se describen los siguientes niveles de conformidad:

- Nivel A: la página Web satisface todos los Criterios de Conformidad de nivel A o se proporciona una versión alternativa que cumple este nivel.
 - Nivel AA: la página Web satisface todos los Criterios de Conformidad de nivel A y nivel AA o se proporciona una versión alternativa que cumple este nivel.
 - Nivel AAA: la página Web satisface todos los Criterios de Conformidad de nivel A, nivel AA y nivel AAA o se proporciona una versión alternativa que cumple este nivel.
- Estas pautas que se han convertido en un estándar internacional en el que debe basarse un sitio web desde las primeras fases de su diseño.

2 El nuevo estándar HTML 5.

HTML5 es la nueva versión del lenguaje de marcado que se usa para estructurar páginas web, actualmente en desarrollo, que surge como una evolución lógica de las especificaciones anteriores con los siguientes objetivos: separar totalmente la información, y la forma de presentarla, resumir, simplificar y hacer más sencillo el código utilizado, incorporar nuevas etiquetas semánticas, páginas compatibles con todos los navegadores web incluyendo los de los teléfonos móviles y otros dispositivos utilizados en la actualidad para navegar en Internet.

En definitiva representa un importante paso adelante hacia el "diseño universal", es decir, *diseño que puedan utilizar todas las personas, en la mayor medida posible, sin necesidad de adaptación ni diseño especializado* [14].

En las secciones siguientes veremos las nuevas características que incorpora HTML5 y como facilitan la mejora de la accesibilidad en el diseño web.

2.1 Diseño web más semántico

Las nuevas etiquetas se crean para remplazar el abuso de elementos `<div>` que habitualmente se utilizan con el fin de delimitar las partes de un documento. Con su utilización, HTML5 pretende crear una estructura de vistas más homogeneizada y que el manejo de *metadata* sea más adecuado. Se trata de llegar a lo que se ha dado en denominar la "web semántica", siendo la presentación de documentos mucho más coherente y fácil de entender por otras personas y lo que es más importante, siendo trivial de entender para una máquina. Concretamente, la tarea de un buscador será mucho más fácil, pero también lo será para cualquier aplicación que "lea" páginas web.

2.1.1 Nuevas etiquetas estructurales

HTML5 incorpora nuevas etiquetas para estructurar páginas web. Estos elementos, denominados generalmente elementos semánticos sirven para dividir un documento en partes lógicas, indicando el tipo de contenido que se incluye en cada parte, representan por tanto secciones lógicas o componentes de una aplicación web o un documento: `<header>`, `<nav>`, `<article>`, `<section>`, `<aside>`, `<footer>` [9, 12, 16].



Fig. 1. Ejemplo de cambios en un documento escrito en HTML 4.0 a HTML5 con los nuevos elementos [19].

- **<header>**, representa el encabezado de un documento o de una sección de un documento. El uso más común de esta etiqueta es agrupar los elementos del encabezado principal de una página (logo, título, navegación principal...).
- **<nav>**, éste elemento proporciona una forma de especificar de forma explícita los elementos de navegación en las páginas. No todos los grupos de enlaces en una página tienen que estar incluidos en éste elemento, sólo las secciones que consisten en bloques principales de la navegación son apropiadas para su utilización.
- **<article>**, se aplica al contenido de un sitio web que puede ser publicado o distribuido de forma independiente, como entradas en blogs o artículos de noticias. Así, podríamos tener varios artículos demarcados semánticamente, por lo que una herramienta puede extraerlos fácilmente.
- **<section>**, engloba a un bloque de contenido relacionado, por lo que representa a una sección “general” dentro de un documento o aplicación, como un capítulo de un libro. Puede contener subsecciones y si lo acompañamos de **h1-h6** podemos estructurar mejor toda la página. Para agrupar contenidos que no estén relacionados, con el objetivo de darles estilo, hay que seguir utilizando el elemento **<div>**.
- **<aside>**, representa contenido secundario relacionado con el contenido principal de un documento y resulta esencial para delimitar el contenido “importante” del contenido “de apoyo”. Se utiliza para bloques publicitarios, enlaces externos, citas, calendario de eventos, etc, u otros contenidos que se consideren separados del contenido principal de la página.
- **<footer>**, representa el pie de página de un documento o de una parte del documento. Está pensado para incluir la información que normalmente se coloca al final de un contenido, como el nombre del autor, enlaces relacionados o derechos de autor.

Además HTML5 incorpora nuevas reglas para el uso de las etiquetas de cabecera **<h1>-<h6>** y los elementos **<dirección>**, con el fin de establecer de forma adecuada la estructuración de la página y jerarquización de los contenidos, permitiendo saltar bloques de información y seleccionar la sección deseada por el usuario de un lector de pantalla.[15,18]

En general, todas estas incorporaciones tienen un importante potencial para mejorar la accesibilidad, sobre todo si los lectores de pantalla y otras tecnologías de asistencia comienzan a utilizar la información asociada.

2.1.2 Nuevos tipos para la etiqueta **<input>** y sus extensiones

HTML5 define trece nuevos valores para el elemento **<input>**. Estos nuevos controles ofrecen claras mejoras, permiten lograr una mejor consistencia entre diferentes sitios web, lo que disminuye su complejidad y facilita su uso, queda por ver

cómo se representarán o serán utilizados por los navegadores web cada uno de los nuevos tipos de entrada [9, 12, 15].

- **Tipo de entrada Search**, el tipo de entrada de búsqueda representa un campo de texto que funciona como una caja de búsqueda. Permitirá a las tecnologías de asistencia proporcionar, mediante un solo comando o pulsaciones de teclas, el acceso a la casilla de búsqueda de la página actual, independientemente de dónde se encuentre en el contenido.
- **Tipos de entrada tel, url y email**, estos tipos proporcionan 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. También permite la integración con otras fuentes de datos.
- **Tipos de entrada date y time**, seis de los nuevos tipos de entrada están relacionados con fechas y horas en una u otra forma: **datetime**, **date**, **month**, **week**, **time** y **datetime-local**, se espera que los navegadores web y otras aplicaciones de usuario presentarán estos campos de entrada de forma que se facilite notablemente la lectura y la selección sencilla de los valores apropiados, esto con toda seguridad generará un aumento significativo de la accesibilidad, frente a los códigos actuales basados habitualmente en JavaScript.
- **Tipos de entrada numéricos(number , range) y Color** se utilizarán para la entrada de valores numéricos y los relacionados con los elementos visuales respectivamente.

Además de los nuevos tipos **<input>**, **HTML5** también añade cuatro nuevos atributos a todos los elementos de entrada que tienen un impacto potencial en la accesibilidad. Todos ellos implementan características y funciones que se han estado utilizando en aplicaciones web desde hace años a través de secuencias de comandos o scripts, éste estándar nos permitirá su codificación directamente en HTML [9, 12, 18].

- **Enfoque automático (autofocus)**. HTML5 introduce este atributo en los formularios, mueve el cursor a un campo **<input>** particular y por tanto la atención del usuario.
- **Marcador de posición de texto (placeholder)**. Este atributo permitirá a los desarrolladores web especificar el marcador de posición del texto directamente en el código del formulario, permitiendo la inclusión de tecnologías para ayuda de una manera más consistente. Además, el uso generalizado de éste atributo asegurará que el texto del marcador de posición tenga un aspecto y comportamiento uniforme, a lo largo de la Web, incrementando la accesibilidad y facilitando el aprendizaje de uso de los sitios web.
- **Campos requeridos (required)**. HTML5 incluye éste atributo booleano que nos sirve para definir si un campo es obligatorio o no, solucionando el problema de validación de formularios del lado del cliente.
- **Patrones (pattern)**. Indica 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.

2.2 Elementos multimedia.

La capacidad multimedia de HTML5 es uno de los grandes avances de esta última versión, el video y el audio se incorporan como dos elementos más [11], aunque la inexistencia de formatos estándar para su inserción en los sitios web es todavía una limitación importante.

2.2.1 La etiqueta <video>.

El uso de la nueva etiqueta <video> tiene el problema de que el soporte de los formatos o codecs más utilizados (Ogg Theora, H.264/MPEG-4, VP8/WebM) en los navegadores actuales no es uniforme. HTML5 permite la presentación de varios formatos para que el navegador seleccione aquél que sea compatible con sus posibilidades de reproducción:

```
<video>
  <source src="film.mp4" type="video/mp4" />
  <source src="film.webm" type="video/webm" />
  <source src="film.ogv" type="video/ogg" />
</video>
```

La etiqueta <video> permite atributos que hacen mucho más flexible su utilización mediante los navegadores que ya soportan esta característica: **Src** (URL del archivo que contiene la película a reproducir), **Autoplay** (la reproducción del video comienza en cuanto la página termina de cargarse), **Controls** (incluye los controles *play*, *pause* y *stop* para la reproducción del video), **Height** y **Width** (se indica el alto y ancho del cuadrilátero donde se ubicará la película), **Loop** (la reproducción de nuestra película se inicia de nuevo automáticamente), **Muted** (la reproducción se realizará sin sonido), **Poster** (archivo de imagen que representa la carátula del video mientras la reproducción está detenida, antes de que el usuario presione el botón *play*).

Respecto a la accesibilidad de la etiqueta <video> [4,5], y a la espera de las especificaciones de la misma y las implementaciones de los distintos navegadores, es previsible que resuelva dos de los mayores problemas que se presentan en la actualidad:

- **Accesibilidad del teclado.** Los controles del reproductor, proporcionados directamente por el navegador, se pueden manejar mediante el teclado sin problemas.
- **Soporte de subtítulos.** HTML incorpora una segunda etiqueta, *track*, que permite incluir el texto asociado a la imagen y el sonido, como en el caso de los subtítulos con un idioma diferente al original.

```
<track src="subtitulosesp.vtt" kind="subtitles"
srclang="es"
label="Castellano">
```

La etiqueta <track> posee el atributo src que permite indicar un fichero externo que contiene los subtítulos (con sus correspondientes marcas temporales). El formato propuesto para los subtítulos se llama WebVTT (Web Video Text Tracks, .vtt) se trata de archivos de texto plano que contienen diferentes tipos de información sobre el video:

- **Subtítulos:** La transcripción o traducción del dialogo.
- **Captions:** Son similares a los subtítulos pero pueden incluir información de audio
- **Descripciones:** Destinado a ser un archivo de texto independiente que describe el video a través de un lector de pantalla.
- **Capítulos:** Destinado a ayudar al usuario a navegar a través del video
- **Metadata:** Información y contenido sobre el video, que no está destinada por default a ser visualizada por el usuario, aunque es posible realizarlo mediante Javascript.

2.2.2 La etiqueta <audio>

Esta nueva etiqueta tiene prácticamente las mismas propiedades que la etiqueta <video>, también con sus correspondientes problemas de compatibilidad de formatos o *códecs* en los diferentes navegadores. 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, etc directamente desde la página web.

Los atributos que permite son los ya descritos para la etiqueta <video>: **Autoplay**, **controls**, **loop**, **preload** y **src**.

2.3 Nuevos métodos y funciones Javascript en HTML5

Hoy en día las páginas web han evolucionado hacia las llamadas aplicaciones web, que tienen un funcionamiento más dinámico con respecto a los contenidos. Para ello, se han utilizado tecnologías web como *JavaScript* o *AJAX*, que en los últimos años han tenido un crecimiento sustancial. HTML5 incorpora nuevos métodos y funciones *JavaScript* que facilitan la interactividad de las páginas con el usuario.

De cara a la accesibilidad del documento es preferible manipular la estructura de etiquetas del documento modificando dinámicamente el árbol de objetos (DOM), que muchos productos de apoyo utilizan para acceder al contenido [1, 8]. Se han incorporado nuevos métodos para este propósito, que permiten obtener un código más limpio, separando el marcado HTML del documento y el código Javascript.

- **AddEventListener(evento, función, booleano)**, permite registrar los controladores de eventos para el mismo evento en un elemento. Además del evento a escuchar y la función que deberá ser ejecutada, el método recibe un valor lógico que indica el orden de ejecución del evento en caso de varios elementos.

- **getElementsByClassName(classN)**, permite seleccionar elementos por el atributo class.
- **activeElement** nos permitirá conocer el elemento que tiene el foco, elemento activo en tiempo real.
- **Hasfocus(elemento)** Devuelve un valor booleano que indica si el documento o cualquier elemento dentro del documento tiene el foco.
- **getSelection()**, devuelve un objeto con la selección actual.

HTML5 incorpora también nuevos manejadores de eventos [17, 18]. Muchos de ellos son eventos lógicos o independientes del dispositivo por lo que permiten dotar de mayor accesibilidad [4, 6] a las páginas, destacaremos aquí los siguientes:

- Eventos relacionados con acciones de la ventana del navegador: permiten detectar por ejemplo si un documento va a imprimirse (**Onbeforeprint**) o un documento ha terminado de imprimirse (**Onafterprint**) o un documento a terminado de cargarse (**Onstorage**) o un documento ha cambiado (**Onhaschange**) o un documento tiene un error (**Onerror**), o un documento está desconectado (**Onoffline**) o un documento está activo (**Ononline**), si la ventana está activa (**Onpageshow**) u oculta (**Onpagehide**) o si la ventana cambia de tamaño (**Onresize**),etc.
- Eventos relacionados con el funcionamiento de los elementos multimedia incluidos en la página: permite detectar si los datos de los elementos multimedia van a empezar a reproducirse (**Onplay**), o si los datos de los elementos multimedia están en pausa (**OnPause**) o si el navegador empieza a cargar los datos de los elementos multimedia (**onLoadStart**) o si se produce un error durante la carga de un elemento (**Onerror**) o si un recurso de un elemento multimedia de repente se desconecta (**Onemptied**) o si los elementos multimedia se puedan iniciar (**Oncanplay**), etc.
- 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: cuando un menú contextual se activa (**Oncontextmenu**), o un menú cambia de forma (**Onformchange**) o un formulario de entrada del usuario se ejecuta (**Onforminput**) o cuando un elemento recibe la entrada del usuario (**Oninput**) o un elemento no es válido (**Oninvalid**).

3 Conclusiones

Después de la revisión realizada de la nueva versión del lenguaje de marcado HTML5, en fase de desarrollo, podemos afirmar que ofrece un futuro muy prometedor en relación con la accesibilidad de la web. Como hemos visto se han incorporado nuevas etiquetas que aportan fundamentalmente mayor semántica al diseño de la página y otras que permiten incorporar elementos multimedia con menor dependencia de las tecnologías externas utilizadas hasta el momento. También se han incorporado nuevos eventos independientes del dispositivo y funciones *Javascript* que

permiten acceder directamente a los elementos de la página, permitiendo separar el diseño de los *scripts* del código y favoreciendo la programación de éstos de forma compatible con las Pautas de Accesibilidad.

No obstante, para lograr que el objetivo se haga realidad en los futuros diseños, es necesario ver como va evolucionando el soporte y compatibilidad de los diferentes navegadores y dispositivos de asistencia en relación con este nuevo estándar.

Agradecimientos

Este trabajo ha sido financiado en parte por la Comisión Europea, a través del proyecto ESVI-AL del programa ALFA.

Referencias

1. Connor, J., Pro HTML5 Accessibility, The express voice in 'Web Accesibility. Professional Apress (2012).
2. Holguera, J., Internet Explorer 9 y HTML5, La nueva Web. Msdn (2011).
3. Web Content Accessibility Guidelines 1.0. W3C Recommendation W3C-WAI (1999). <http://www.w3.org/TR/WCAG10>. Consultada en octubre de 2012.
4. Web Content Accessibility Guidelines (WCAG) 2.0. W3C-WAI (2008) <http://www.w3.org/TR/2008/REC-WCAG20-20081211>. Consultada en octubre de 2012.
5. Shneiderman, B., Accesibilidad a los contenidos audiovisuales en la web: Una panorámica sobre legislación, tecnologías y estándares. Diseño Universal (2000)
6. Introducción a la Accesibilidad Web, Web accessibility Initiative, W3C, <http://www.w3c.es/Traducciones/es/WAI/intro/accessibility>.
7. MSDN learning resources. Microsoft.com, <http://msdn.microsoft.com/es-es/magazine/hh547102.aspx>. Consultada en octubre de 2012.
8. Gauchat, J. D., El gran libro de HTML5, CSS3 y Javascript, Marcombo ed. (2012)
9. Luján Mora, S. Departamento de Lenguajes y Sistemas Informáticos de la Universidad de Alicante (España). <http://accesibilidadenlaweb.blogspot.com.es/2010/11/el-futuro-de-la-accesibilidad-web-con.html>. Consultada en octubre de 2012.
10. Diseño Web 11Bits <http://11bits.es/etiquetas-estructurales-en-html5/>. Consultada en octubre de 2012.
11. Pimentel, V., Anexo m, Las novedades de HTML5. <http://www.anexom.es/tecnologia/disenio-web/las-novedades-de-html5-i/>. Consultada en octubre de 2012.
12. Consideraciones sobre accesibilidad, Blog AWS Desarrollo Web (2010) <http://desarrollowebaws.com/disenio-web/925/html5-consideraciones-de-accesibilidad-web/> Consultada en octubre de 2012.
13. HTML5 en la Educación, Instituto de Tecnologías Educativas, Ministerio de Educación, Cultura y Deportes. <http://www.ite.educacion.es/formacion/materiales/182/cd/catorce/html5.html>. Consultada en octubre de 2012.
14. Convención de los derechos de las personas con discapacidad, Naciones Unidas. (2008). <http://www.un.org/disabilities/default.asp?navid=12&pid=150>

15. New Elements in HTML5, Structure and Semantic, IBM (2007),
<http://www.ibm.com/developerworks/library/x-html5/?ca=dgr-lnxw01NewHTML>,
Consultada en octubre de 2012.
16. Guía de Accesibilidad para desarrolladores, UOC (2011).
17. QuirksMode.org, http://www.quirksmode.org/js/events_advanced.html.
18. HTML5 Event Attributes, http://www.w3schools.com/html5/html5_ref_eventattributes.asp, www.w3schools.org.
19. Manual de HTML5 en Español. Alejandro Castillo Cantón [www.theproc.es](http://11bits.es/etiquetas-estructurales-en-html5/).
<http://11bits.es/etiquetas-estructurales-en-html5/>