Semántica y accesibilidad

> Uso de herramientas necesarias para cumplir con los estándares de accesibilidad: role, wai-aria.

Buen uso del atributo "rel"

Fuente: http://www.w3schools.com/tags/att_link_rel.asp

Nombre	Descripción
alternate	Proporciona un enlace a una versión alternativa del documento (es decir, imprimir página, traducir o reflejar). Ejemplo: <link rel =" alternate "=" application / atom + xml "title =" W3Schools News "href =" / blog / news / atom "
author	Propoerciona un enlace al autor del documento.
dns-prefetch	Especifica que el navegador debe realizar de forma preventiva la resolución de DNS para el origen del recurso de destino.
help	Proporciona un enlace al documento de ayuda. Ejemplo: <link rel="help" href="/help/">
icon	Importa un icono que representa al documento. Ejemplo: <link rel="icon" href="/favicon.ico" type="image/x-icon">
license	Proporciona un enlace a la información copyright del documento.
next	Proporciona un enlace al siguiente documento en la serie.
pingback	Proporciona la dirección del servidor del pingback que maneja los pingbacks al documento actual (Pingback es un método para que los autores de la tela solicitan una notificación cuando alguien enlaza uno de sus documentos).
preconnect	Especifica que el navegador debe conectarse de forma preventiva al origen del recurso de destino.
prefetch	Especifica que el navegador debe recuperar y almacenar en caché el recurso de destino de forma preventiva, ya que es probable que sea necesario para una navegación de seguimiento.
preload	Especifica que el agente del navegador debe recuperar y almacenar en caché el recurso de destino para la navegación actual de acuerdo con el destino dado por el atributo "as" (y la prioridad asociada con ese destino).
prev	Indica que el documento forma parte de una serie y que el documento anterior de la serie es el documento de referencia.
search	Proporciona un enlace a un recurso que se puede utilizar para buscar en el documento actual y sus páginas relacionadas.
stylesheet	Importa una hoja de estilo.

Etiquetas semánticas HTML5

Etiqueta	Descripción
<article>	Define un artículo.
<aside>	Define el contenido aparte del contenido de la página.
<details>	Define detalles adicionales que el usuario puede ver u ocultar.
<figcaption>	Define una leyenda para un elemento <figure>.

<figure>	Especifica contenido autónomo, como ilustraciones, diagramas, fotos, listas de códigos, etc.
<footer>	Define un pie de página para un documento o sección.
<header>	Especifica un encabezado para un documento o sección.
<main>	Especifica el contenido principal de un documento.
<mark>	Define el texto marcado / resaltado.
<nav>	Define vínculos de navegación.
<section>	Define una sección en un documento.
<summary>	Define un encabezado visible para un elemento <details>.
<time>	Define una fecha / hora.

alt text

Elementos de formulario

Etiqueta	Descripción
<datalist>	Define las opciones predefinidas para los controles de entrada.
<keygen>	Define un campo generador de pares de claves (para formularios).
<output>	Define el resultado de un cálculo.

Nuevos tipos de entrada (input)

- color
- date
- datetime
- datetime-local
- email
- month
- number
- range
- search
- tel
- time
- url
- week

Y los nuevos atributos para los inputs

- autocomplete
- autofocus

- form
- formaction
- formenctype
- formmethod
- formnovalidate
- formtarget
- height and width
- list
- min and max
- multiple
- pattern (regexp)
- placeholder
- required
- step

Elementos gráficos

Etiquetas	Descripción
<canvas>	Dibuja gráficos,dinámicamente, a través de secuencias de comandos (normalmente JavaScript).
<svg>	Dibuja gráficos vectoriales escalables.

Elementos Multimedia

Etiqueta	Descripción
<audio>	Define contenido de sonido.
<embed>	Define contenedores para aplicaciones externas (como complementos).
<source>	Define fuentes para y .
<track>	Define pistas para y .
<video>	Define el contenido de vídeo o película.

Otros elementos

HTML5 lista completa de etiquetas:

https://github.com/rmorenomf/formacion-isban/blob/master/Jornada%201/resources/listaetiquetasHTML5.pdf

- Webworkers
- Drag & Drop
- WebSockets WebRTC
- Caché de la aplicación
- IndexDB
- File API / Blob

Accesibilidad

El propósito de ARIA es dotar de significado explícito a un contenido que no lo tiene. En detalle:

- Establecer el rol de un elemento
- Establecer la relación entre elementos
- Informar del estado de un elemento
- Hacer accesibles los controles a través del teclado

Para ello se vale de una serie de elementos estándar:

Roles y estados

1. Roles: su misión es definir el papel que juegan los elemento dentro del documento web.

Algunos roles, muchos de ellos tienen su equivalente en etiquetas HTML 5:

- role="banner"
- role="complementary"
- role="contentinfo"
- role="form"
- role="main"
- role="navigation"
- role="search"
- role="application"

2. Estados y propiedades: determinan las características y los valores de cada elementos.

Landmark Roles (WAI-ARIA). Navegación más accesible y semántica.

El objetivo es que sea más fácil de "ojear", comprender y navegar para las personas que usan un lector de pantalla.

Cuando una página web está correctamente marcada permitimos que los usuarios que usan un lector de pantalla no tengan que hacer una lectura lineal de toda la página. Utilizando determinadas teclas podrán "ojear" el documento y acceder directamente a las partes del mismo que les interesan.

Por ejemplo, si tenemos una correcta estructura de encabezados, marcados como tales, un usuario de lector de pantalla (como JAWS o NVDA) podrá pulsar la tecla “h” para “ojear” los encabezados. Cada vez que pulse dicha tecla el lector le leerá el siguiente encabezado y podrá seguir leyendo a partir del que le interese.

Diseñando elementos interactivos.

```
    <section>
        <h1>Encuesta de satisfaccion</h1>

        Aqui vamos a meter un widget que sea un slider y un boton de enviar respuesta.

        <!-- Podemos consultar las caracteristicas wai-aria de este rol en https://www.w3.org/TR/wai-
aria/roles#slider -->
        <label for="rating_slider">Valoracion</label>
        <input type="range"
            id="rating_slider"
            min="0"
```

```
            max="5"
            value="0"
            step="1"
            aria-valuemin="0"
            aria-valuemax="5"
            aria-valuenow="1"
            aria-hidden="false"
            tabindex="0"
            oninput="outputUpdate(value)">
        <output for="rating_slider" id="rating">0</output>

            <!-- En este caso por tratarse de una etiqueta boton no es necesario role="button" está
implicito, no es necesario y además no es recomendable
                https://www.w3.org/TR/2014/REC-html5-20141028/dom.html#aria-usage-note
            -->
        <button aria-pressed="false" onclick="handleBtnClick(event)"
onKeyUp="handleBtnKeyUp(event)">Enviar respuesta</button>
        <a href="" role="button">Envia respuesta</a>

    </section>
```

Pasos y buenas prácticas para aplicar WAI-ARIA

En el documento del W3C, WAI-ARIA 1.0 Primer, se nos indican los pasos y buenas prácticas para aplicar WAI-ARIA.

- Usa marcado nativo cuando sea posible. Si puedes usar o
  úsalos en vez de
  o
  . Ten en cuenta que el rol ARIA anula el rol nativo.
- Usa los roles adecuados. Según la especificación y recuerda que el rol no se debe cambiar dinámicamente.
- Conserva la estructura semántica. Forma grupos lógicos (por ejemplo con role="group", role="toolbar"); incluye landmark roles para facilitar la navegación por teclado; define las live regions (zonas que cambian dinámicamente sin intervención del usuario)
- Construye relaciones. Busca relaciones entre los elementos y márcalas con el atributo más apropiado (por ejemplo aria-controls para las pestañas. Puedes consultar más en la especificación: Relationship Attributes) Ten en cuenta que algunas relaciones son automáticas como la de un label con su input.
- Cambia los estados y propiedades en respuesta a los eventos. Usa solo atributos soportados por el rol o el elemento elegido. Cambia dinámicamente por javascript los estados y propiedades que se introduzcan durante el ciclo de vida del elemento, por lo general en respuesta a los eventos de entrada de usuario; de esta manera los agentes de usuario notificarán a los productos de apoyo los cambios de estado.

Accesible mediante el teclado

El usuario debe poder interactuar con los controles mediante el teclado. Utilizará el tabulador para acceder a ellos (no solo debe ser accesible mediante el tabulador sino también seguir un orden de tabulación lógico) y a menudo las teclas de flechas para moverse en el interior de controles complejos.

Se debe implementar el comportamiento de las teclas de flechas, de espacio, etc. seguiendo los patrones definidos en WAI-ARIA Authoring Practices Guide (Keyboard and Structural Navigation) y WAI-ARIA Authoring Practices Guide (Design Patterns), donde se incluyen ejemplos como el comportamiento esperado de la tecla de espacio, tabulación y las flechas arriba y abajo en los grupos de radiobuttons.

Sincroniza la interfaz visual con la interfaz accesible Y para ello te es muy útil los CSS attribute selectors, teniendo en cuenta el soporte para los navegadores más antiguos. Podéis consultar un ejemplo de cómo aplicarlo en General Steps for Building an Accessible Widget with WAI-ARIA (punto 7: "Synchronize the visual UI with accessibility states and properties for supporting user agents")

WCAG:

Las pautas de accesibilidad al contenido web (Web Content Accessibility Guidelines - WCAG) son unos documentos que explican cómo hacer

el contenido Web accesible para personas con discapacidad. Cuando se habla de contenido se refiere a la información en una página web o aplicación, incluyendo texto imágenes, formas, sonidos, etc. En la actualidad existen dos conjuntos de pautas de accesibilidad de contenidos Web: WCAG1.0 y WCAG2.0. Estos dos conjuntos están organizados y estructurados de diferente forma: WCAG1.0. Son 14 pautas que engloban los principios generales del diseño accesible. En total poseen 65 puntos de verificación y cada uno de ellos está asociado a una prioridad: Simple A, Doble A y Triple A. WCAG2.0. Se organiza en cuatro principios fundamentales para la accesibilidad del contenido:

- Perceptible
- Operable
- Comprensible
- Robusto

Cada uno de estos principios tiene asociadas una o más pautas y en total forman 61 criterios de conformidad, organizados según su nivel de cumplimiento asociado: Simple A, Doble A y Triple A. Existe cierta equivalencia entre los puntos de verificación de WCAG1.0 y WCAG2.0 y con la Norma UNE 139803, que actualmente regula 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. A pesar de que esta norma es compatible las pautas WCAG1.0, existe una petición por parte de AENOR de que se actualice la regulación existente con las pautas WCAG2.0, por lo que será en las que se base el presente texto. En las WCAG2.0 existen tres niveles de conformidad: Existen tres niveles de conformidad:

1. WCAG 2.0 Nivel A: para lograr conformidad con el Nivel A (el mínimo), la página web debe satisfacer todos los Criterios de Conformidad del Nivel A, o proporcionar una versión alternativa conforme.
2. WCAG 2.0 Nivel AA: para lograr conformidad con el Nivel AA, la página web debe satisfacer todos los Criterios de Conformidad de los Niveles A y AA, o proporcionar una versión alternativa conforme al Nivel AA.
3. WCAG 2.0 Nivel AAA: Para lograr conformidad con el Nivel AAA, la página web debe satisfacer todos los Criterios de Conformidad de los Niveles A, AA y AAA, o proporcionar una versión alternativa conforme al Nivel AAA.

Además, existen las siguientes especificaciones:

- El nivel de conformidad es para páginas completas, no para fragmentos de ella.
- Cuando varias páginas forman parte de un proceso, todas las páginas implicadas deben ser conformes con el nivel especificado o un nivel superior.
- Si una aplicación está implementada en una tecnología que no es compatible con las pautas de accesibilidad, debe proporcionarse una versión que sí lo sea.
- Si una página posee una tecnología que no es compatible con la accesibilidad o no es conforme con cierto nivel, no debe impedir el acceso al contenido del resto de la página.

La declaración de conformidad es opcional, pero si se añade debe incluir la siguiente información:

- Fecha de la declaración de conformidad
- Título, versión y URI de las Pautas: "Web Content Accessibility Guidelines 2.0 en http://www.w3.org/TR/WCAG20/"
- Nivel de conformidad: A, AA o AAA
- Listado de las páginas que están incluidas en la declaración de conformidad
- Listado de tecnologías de contenido web de las que se depende (se recomienda un enlace al software concreto)
- Si se emplea un logo de conformidad, éste constituye una declaración y debe ir acompañado de todos los componentes requeridos para la conformidad.

PUNTOS DE VERIFICACIÓN PRIORIDAD 1

En general (Prioridad 1)

- 1.1 Proporcione un texto equivalente para todo elemento no textual (Por ejemplo, a través de "alt", "longdesc" o en el

contenido del elemento). Esto incluye: imágenes, representaciones gráficas del texto, mapas de imagen, animaciones (Por ejemplo, GIFs animados), "applets" y objetos programados, "ascii art", marcos, scripts, imágenes usadas como viñetas en las listas, espaciadores, botones gráficos, sonidos (ejecutados con o sin interacción del usuario), archivos exclusivamente auditivos, banda sonora del vídeo y vídeos.

- 2.1 Asegúrese de que toda la información transmitida a través de los colores también esté disponible sin color, por ejemplo mediante el contexto o por marcadores.
- 4.1 Identifique claramente los cambios en el idioma del texto del documento y en cualquier texto equivalente (por ejemplo, leyendas).
- 6.1 Organice el documento de forma que pueda ser leído sin hoja de estilo. Por ejemplo, cuando un documento HTML es interpretado sin asociarlo a una hoja de estilo, tiene que ser posible leerlo.
- 6.2 Asegúrese de que los equivalentes de un contenido dinámico son actualizados cuando cambia el contenido dinámico.
- 7.1 Hasta que las aplicaciones de usuario permitan controlarlo, evite provocar destellos en la pantalla.
- 14.1 Utilice el lenguaje apropiado más claro y simple para el contenido de un sitio.

Y si utiliza imágenes y mapas de imagen (Prioridad 1)

- 1.2 Proporcione vínculos redundantes en formato texto para cada zona activa de un mapa de imagen del servidor.
- 9.1 Proporcione mapas de imagen controlados por el cliente en lugar de por el servidor, excepto donde las zonas sensibles no puedan ser definidas con una forma geométrica.

Y si utiliza tablas (Prioridad 1)

- 5.1 En las tablas de datos, identifique los encabezamientos de fila y columna.
- 5.2 Para las tablas de datos que tienen dos o más niveles lógicos de encabezamientos de fila o columna, utilice marcadores para asociar las celdas de encabezamiento y las celdas de datos.

Y si utiliza marcos ("frames") (Prioridad 1)

- 12.1 Titule cada marco para facilitar su identificación y navegación.

Y si utiliza "applets" y "scripts" (Prioridad 1)

- 6.3 Asegure que las páginas sigan siendo utilizables cuando se desconecten o no se soporten los scripts, applets u otros objetos programados. Si esto no es posible, proporcione información equivalente en una página alternativa accesible.

Y si utiliza multimedia (Prioridad 1)

- 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.
- 1.4 Para toda presentación multimedia tempodependiente (por ejemplo, una película o animación) sincronice alternativas equivalentes (por ejemplo, subtítulos o descripciones de la banda visual) con la presentación.

Y si todo lo demás falla (Prioridad 1)

- 11.4 Si, después de los mayores esfuerzos, no puede crear una página accesible, proporcione un vínculo a una página alternativa que use tecnologías W3C, sea accesible, tenga información (o funcionalidad) equivalente y sea actualizada tan a menudo como la página (original) inaccesible.

PUNTOS DE VERIFICACIÓN PRIORIDAD 2

En general (Prioridad 2)

- 2.2 Asegúrese de que las combinaciones de los colores de fondo y primer plano tengan el suficiente contraste para que sean percibidas por personas con deficiencias de percepción de color o en pantallas en blanco y negro [Prioridad 2 para las imágenes. Prioridad 3 para los textos].
- 3.1 Cuando exista un marcador apropiado, use marcadores en vez de imágenes para transmitir la información.

- 3.2 Cree documentos que estén validados por las gramáticas formales publicadas.
- 3.3 Utilice hojas de estilo para controlar la maquetación y la presentación.
- 3.4 Utilice unidades relativas en lugar de absolutas al especificar los valores en los atributos de los marcadores de lenguaje y en los valores de las propiedades de las hojas de estilo.
- 3.5 Utilice elementos de encabezado para transmitir la estructura lógica y utilícelos de acuerdo con la especificación.
- 3.6 Marque correctamente las listas y los ítems de las listas.
- 3.7 Marque las citas. No utilice el marcador de citas para efectos de formato tales como sangrías.
- 6.5 Asegúrese de que los contenidos dinámicos son accesibles o proporcione una página o presentación alternativa.
- 7.2 Hasta que las aplicaciones de usuario permitan controlarlo, evite el parpadeo del contenido (por ejemplo, cambio de presentación en periodos regulares, así como el encendido y apagado).
- 7.4 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener las actualizaciones, no cree páginas que se actualicen automáticamente de forma periódica.
- 7.5 Hasta que las aplicaciones de usuario proporcionen la posibilidad de detener el redireccionamiento automático, no utilice marcadores para redirigir las páginas automáticamente. En su lugar, configure el servidor para que ejecute esta posibilidad.
- 10.1 Hasta que las aplicaciones de usuario permitan desconectar la apertura de nuevas ventanas, no provoque apariciones repentinas de nuevas ventanas y no cambie la ventana actual sin informar al usuario.
- 11.1 Utilice tecnologías W3C cuando estén disponibles y sean apropiadas para la tarea y use las últimas versiones que sean soportadas.
- 11.2 Evite características desaconsejadas por las tecnologías W3C.
- 12.3 Divida los bloques largos de información en grupos más manejables cuando sea natural y apropiado.
- 13.1 Identifique claramente el objetivo de cada vínculo.
- 13.2 Proporcione metadatos para añadir información semántica a las páginas y sitios.
- 13.3 Proporcione información sobre la maquetación general de un sitio (por ejemplo, mapa del sitio o tabla de contenidos).
- 13.4 Utilice los mecanismos de navegación de forma coherente.

Y si utiliza tablas (Prioridad 2)

- 5.3 No utilice tablas para maquetar, a menos que la tabla tenga sentido cuando se alinee. Por otro lado, si la tabla no tiene sentido, proporcione una alternativa equivalente (la cual debe ser una versión alineada).
- 5.4 Si se utiliza una tabla para maquetar, no utilice marcadores estructurales para realizar un efecto visual de formato.

Y si utiliza marcos ("frames") (Prioridad 2)

- 12.2 Describa el propósito de los marcos y cómo éstos se relacionan entre sí, si no resulta obvio solamente con el título del marco.

Y si utiliza formularios (Prioridad 2)

- 10.2 Hasta que las aplicaciones de usuario soporten explícitamente la asociación entre control de formulario y etiqueta, para todos los controles de formularios con etiquetas asociadas implícitamente, asegúrese de que la etiqueta está colocada adecuadamente.
- 12.4 Asocie explícitamente las etiquetas con sus controles.

Y si utiliza "applets" y "scripts" (Prioridad 2)

- 6.4 Para los scripts y applets, asegúrese de que los manejadores de eventos sean independientes del dispositivo de entrada.
- 7.3 Hasta que las aplicaciones de usuario permitan congelar el movimiento de los contenidos, evite los movimientos en las páginas.
- 8.1 Haga los elementos de programación, tales como scripts y applets, directamente accesibles o compatibles con las ayudas técnicas [Prioridad 1 si la funcionalidad es importante y no se presenta en otro lugar; de otra manera, Prioridad 2].

- 9.2 Asegúrese de que cualquier elemento que tiene su propia interfaz pueda manejarse de forma independiente del dispositivo.
- 9.3 Para los "scripts", especifique manejadores de evento lógicos mejor que manejadores de evento dependientes de dispositivos.

PUNTOS DE VERIFICACIÓN PRIORIDAD 3

En general (Prioridad 3)

- 4.2 Especifique la expansión de cada abreviatura o acrónimo cuando aparezcan por primera vez en el documento.
- 4.3 Identifique el idioma principal de un documento.
- 9.4 Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formulario y objetos.
- 9.5 Proporcione atajos de teclado para los vínculos más importantes (incluidos los de los mapas de imagen de cliente), los controles de formulario y los grupos de controles de formulario.
- 10.5 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten claramente los vínculos contiguos, incluya caracteres imprimibles (rodeados de espacios), que no sirvan como vínculo, entre los vínculos contiguos.
- 11.3 Proporcione la información de modo que los usuarios puedan recibir los documentos según sus preferencias (por ejemplo, idioma, tipo de contenido, etc.).
- 13.5 Proporcione barras de navegación para destacar y dar acceso al mecanismo de navegación.
- 13.6 Agrupe los vínculos relacionados, identifique el grupo (para las aplicaciones de usuario) y, hasta que las aplicaciones de usuario lo hagan, proporcione una manera de evitar el grupo.
- 13.7 Si proporciona funciones de búsqueda, permita diferentes tipos de búsquedas para diversos niveles de habilidad y preferencias.
- 13.8 Localice la información destacada al principio de los encabezamientos, párrafos, listas, etc.
- 13.9 Proporcione información sobre las colecciones de documentos (por ejemplo, los documentos que comprendan múltiples páginas).
- 13.10 Proporcione un medio para saltar sobre un ASCII art de varias líneas.
- 14.2 Complemente el texto con presentaciones gráficas o auditivas cuando ello facilite la comprensión de la página.
- 14.3 Cree un estilo de presentación que sea coherente para todas las páginas.

Y si utiliza imágenes o mapas de imagen (Prioridad 3)

- 1.5 Hasta que las aplicaciones de usuario interpreten el texto equivalente para los vínculos de los mapas de imagen de cliente, proporcione vínculos de texto redundantes para cada zona activa del mapa de imagen de cliente.

Y si utiliza tablas (Prioridad 3)

- 5.5 Proporcione resúmenes de las tablas.
- 5.6 Proporcione abreviaturas para las etiquetas de encabezamiento.
- 10.3 Hasta que las aplicaciones de usuario (incluidas las ayudas técnicas) interpreten correctamente los textos contiguos, proporcione un texto lineal alternativo (en la página actual o en alguna otra) para todas las tablas que maquetan texto en paralelo, en columnas de palabras.

Y si utiliza formularios (Prioridad 3)

- 10.4 Hasta que las aplicaciones de usuario manejen correctamente los controles vacíos, incluya caracteres por defecto en los cuadros de edición y áreas de texto.

Herramientas de verificación WAI-ARIA

La más importante, cerrar los ojos e intentar usar un lector de pantalla.

1. *HERA*: Herramienta online, disponible en español, que valida automáticamente la accesibilidad de la página, señalando qué puntos revisar manualmente. http://www.sidar.org/hera/
2. *Cynthia Says*: Detecta automáticamente problemas de accesibilidad tanto de WCAG 1.0 como de Section 508. http://www.cynthiasays.com
3. *Google Accessibility Developer Tool* es una extensión de Google para Chrome que permite realizar una validación automática de la página que se está visualizando en el navegador https://chrome.google.com/webstore/detail/accessibility-developer-t/fpkknkljclfencbdbgkenhalefipecmb
4. *OCAWA Web Accessibility Expert*: Herramienta de validación automática de páginas web compatible con WCAG 1.0 de Prioridad 1. http://www.ocawa.com/en/
5. *PISTA*: Permite analizar automáticamente varios sitios y todas sus páginas de vez (la gramática, las CSS y los diferentes Niveles de accesibilidad), y además programar revisiones periódicas cuyos informes te son enviados por correo. Permite revisar con varias normativas (por defecto WCAG 1.0, también permite WCAG 2.0). http://www.mityc.es/DGDSI/PISTA/ACCESIBILIDADFREEWARE/Paginas/pistaacces...
6. *TAW*: Valida automáticamente la accesibilidad de la página, señalando qué puntos revisar manualmente. Se puede seleccionar Nivel A, AA y AAA, así como revisión WCAG 1.0 y 2.0. Está disponible en español, en versión online, local o extensión para Firefox. Incorpora asimismo validación de HTML y CSS y permite analizar también el código JavaScript. http://www.tawdis.net
7. *W3C Validator for MAC OSX with Experimental WAI-ARIA Support*: Validador del W3C para instalación en sistemas operativos Mac OS X. Valida tanto webs locales como a través de internet y permite también configurarse como servicio web. Presenta un soporte limitado a WAI-ARIA. http://habilis.net/validator-sac/
8. *WAVE*: Permite analizar sitios web para ayudar a la evaluación de la accesibilidad mostrando la página original con indicadores insertados dentro de sí misma donde se muestran los problemas de accesibilidad. Dispone también de una barra de herramientas para Firefox http://wave.webaim.org/?lang=es

Otros recursos importantes:

https://www.w3.org/WAI/intro/aria

Soporte de los elementos de HTML5 en los diferentes navegadores desde el punto de vista de la accesibilidad.
http://www.html5accessibility.com/#wbdetails

Resumen sobre Accesibilidad

1. Es un derecho.
2. Puede ser una obligación
3. Pautas:
    - Estilos y diseño.
    - Marcado descriptivo
    - Facilitar la navegación por teclado
    - Roles / Semántica (Marcado)
    - Estados (Programación)
    - Probar en lectores de páginas
    - Revisar listas de verificación de forma automática y manual

En resumen: No es tarea sencilla. Requiere tiempo y esfuerzo. Un modelo de exito es incorporar la figura del experto en accesibilidad que trabaje como mentor y formador del equipo de desarrollo.

Ejemplos de marcado accesible

Ejemplo de imagen acessible

<img src="bobby.jpg" alt="My dog Bobby playing fetch in the park">

- Cuidado con usar imágenes dentro de backgrounds.
- Prestar atención a la descripción o la funcionalidad del elemento visual.

Ejemplo para la indicar la zona de navegación

- HTML 4:

```
<div id="navigation" role="navigation">
        <ul>
                <li id="active"><a id="current" href="home">Inicio</a></li>
                <li><a href="blog">Blog</a></li>
                <li><a href="contacto">Contacto</a></li>
        </ul>
</div>
```

- HTML 5:

```
<nav>
    <ul>
        <li id="active"><a id="current" href="home">Inicio</a></li>
                <li><a href="blog">Blog</a></li>
                <li><a href="contacto">Contacto</a></li>
    </ul>
</nav>
```

Podemos usar el atributo *role* para ablicar estilos:

div[role="navigation"] { color: blue; background-color: inherit; }

Complementarlos con "aria-label" o "aria-labelledby"

Se recomienda etiquetar el elemento al que añades el rol, especialmente cuando hay dos del mismo tipo, para diferenciarlos.

Puedes hacerlo con los atributos "aria-label" y "aria-labelledby". La diferencia es que en el primero pones directamente la etiqueta dentro del atributo: *aria-label*="Menú principal", y en el segundo solo indicas el ID del elemento que hace de título o etiqueta a esa zona.

```
<nav>
    <div id="navigation_label">Menú principal de navegación</div>
    <ul aria-labelledby="navigation_label">
        <li id="active"><a id="current" href="home" aria-label="Acceso al inicio de la
apliación">Inicio</a></li>
                <li><a href="blog" aria-label="Acceso al blog">Blog</a></li>
                <li><a href="contacto" aria-label="Acceso a la información de contacto">Contacto</a>
</li>
    </ul>
</nav>
```

```
<button aria-label="Close" onclick="myDialog.close()">X</button>
```

Ejemplo de formulario accesible

```
<form action="post">
     <label for="firstName">First name:</label>
     <input id="firstName" type="text" aria-required="true" />
     <br/>
     <label for="lastName">Last name:</label>
     <input id="lastName" type="text" aria-required="true" />
     <br/>
     <label for="streetAddress">Street address:</label>
     <input id="streetAddress" type="text" />
</form>
```

Haciendo que el atributo tabindex sea soportado por todos los elementos visibles de una web.

- Si no está el atributo tabindex, el comportamiento es el normal, y sólo los controles de un formulario pueden recibir el foco con el ratón o desde JavaScript con element.focus().
- Si tabindex toma un valor negativo (tabindex="-1"), el elemento puede recibir el foco con el ratón o desde JavaScript con element.focus(), pero no se puede navegar a él mediante la tecla tabulación, es decir, el elemento no está en el flujo normal de tabulación.
- Si tabindex toma el valor cero (tabindex="0"), el elemento puede recibir el foco con el ratón o desde JavaScript con element.focus(), y se puede navegar a él mediante la tecla tabulación (la posición que ocupa en el orden de tabulación es la correspondiente a su posición en el documento).
- Finalmente, si tabindex toma un valor positivo (tabindex="1"), el elemento puede recibir el foco con el ratón o desde JavaScript con element.focus(), se puede navegar a él mediante la tecla tabulación, y el valor del atributo indica su posición en el orden de tabulación; además, se colocan por delante de cualquier elemento que tenga tabindex="0" o cualquier elemento que reciba el foco de forma natural (los controles de los formularios y los enlaces).

Ejemplo de componte accesible

Un humilde checkbox:

```
<span role="checkbox"
      aria-checked="true"
      tabindex="0"
      id="simulatedcheckbox"
      onclick="onCheckClick(this);">
</span>
```

```
function onCheckClick(ev){

}
```

Enlaces a recursos WAI-ARIA

- NVDA Screen reader http://www.nvaccess.org/ lector gratutito para Windows.

- WAI-ARIA modelo de roles https://www.w3.org/TR/wai-aria/rdf_model.svg
- Web Content Accessibility Guidelines (WCAG) 2.0 en Español http://www.sidar.org/traducciones/wcag20/es/