# UD 6\_2: BUENAS PRACTICAS DE ACCESIBILIDAD

## WEB ÚTILES

* Accesibilidad: [WCAG 3.0 - Novedades del último borrador (8 de junio de 2021) (olgacarreras.blogspot.com)](https://olgacarreras.blogspot.com/2021/01/wcag-30-novedades-del-ultimo-borrador.html)
* Comprender las pautas o criterios de accesibilidad: <http://www.sidar.org/traducciones/wcag20/es/comprender-wcag20/>
* Legislación: [Legislación (olgacarreras.blogspot.com)](https://olgacarreras.blogspot.com/2009/04/dos-anos-de-usable-y-accesible.html#t1_3)
* WCAG 2 Checklist: <http://webaim.org/standards/wcag/checklist#sc1.1.1>
* WCAG 2 Checklist (en castellano): <http://qweos.net/blog/2009/01/28/guias-practicas-para-profesionales-web-puntos-de-verificacion-de-las-pautas-de-accesibilidad-al-contenido-web-wcag-20/>

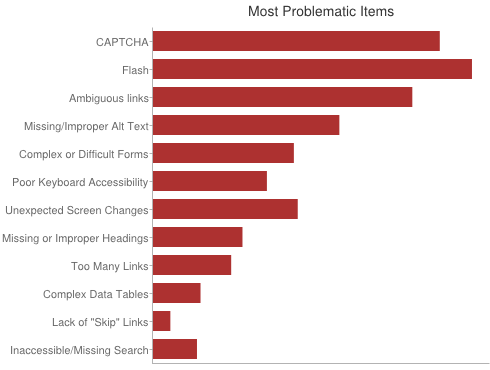
## CONTENIDO ACCESIBLE

El contenido web hace referencia a la información contenida en una página web o en una aplicación web, incluyendo texto, imágenes, formularios, sonido, etc. Es decir, todo lo que se incluye en una página web o es enlazado desde una página web se considera contenido web.

### LOS 4 PRINCIPALES PROBLEMAS RELACIONADOS CON EL CONTENIDO

En mayo de 2012, el WebAIM, un centro dependiente de la Utah State University, que se dedica a la accesibilidad web, realizó por cuarta vez una encuesta sobre el uso de los lectores de pantalla, en ella realizó la siguiente pregunta a los participantes:

¿Qué elemento es el más problemático en una página web?



Del gráfico se deduce que los cuatro contenidos más problemáticos son:

* Contenido Flash inaccesible.
* Los CAPTCHA.
* Los enlaces o botones que no tienen sentido fuera del contexto en el que se encuentran.
* Las imágenes sin un texto alternativo que las describa o un texto alternativo inapropiado.

#### FLASH inaccesible

El primer problema que presenta mayor dificultad de acceso es el uso de la tecnología FLASH de forma inaccesible.

¿Podemos usar FLASH en nuestros sitios web de forma que sigan siendo accesibles?

Sí, pero con prudencia:

* No desarrolles tu sitio web 100% en FLASH.
* No desarrolles elementos importantes como los menús de navegación de FLASH.

¿Cómo usamos FLASH de forma accesible?

* FLASH accesible de forma nativa, es decir, no necesita alternativa HTML (Aclaración: <https://olgacarreras.blogspot.com.es/2011/01/respuesta-25-dudas-habituales-sobre.html#1904094> )
* FLASH no es accesible, necesita de alternativa en HTML accesible.

Podéis encontrar 36 técnicas donde se explica cómo implementar FLASH accesible de forma nativa, en la siguiente dirección web: <http://www.w3.org/TR/WCAG20-TECHS/flash.html>

#### CAPTCHA

El segundo problema que presenta dificultad de acceso a los usuarios de lectores de pantalla es el uso de CAPTCHA.



En la imagen anterior, se muestra el test más habitual que es incluir en una imagen letras o palabras distorsionadas para que el usuario las reconozca y las introduzca por teclado. Esto lo hacemos para evitar que accedan robots de spam u otro tipo de software automático a zonas restringidas de nuestro sitio web.

El problema, es que esta forma de incluir un CAPTCHA crea una barrera de acceso a los usuarios que no puedan ver la imagen.

Los CAPTCHA pueden ser:

* Visuales, en ellos se muestra una imagen con las letras o palabras distorsionadas que hemos de introducir.



* También pueden ser auditivos, se pronuncia la palabra que hay que reconocer sobre un ruido de fondo.



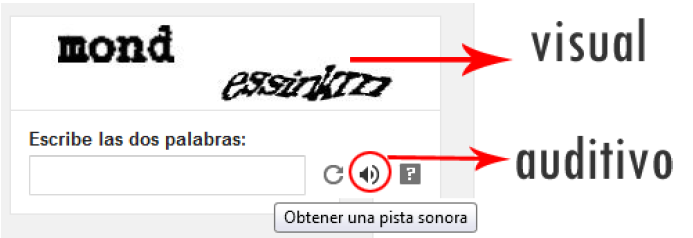
* O pueden ser lógicos, en los que se realiza una pregunta lógica, como por ejemplo:
  + ¿Cuántas son dos más dos?
  + ¿Cuál es la tercera palabra de: Hoy está lloviendo?

¿Cómo programamos un CAPTCHA de forma accesible?

* En primer lugar se ha de proporcionar una alternativa textual que describa su propósito.

<img src=“micaptcha.gif” alt=“Introduzca las letras de la imagen” />

* En segundo lugar, tenemos que proporcionar otro CAPTCHA, con el mismo propósito, pero en una modalidad sensorial diferente:



Cada modalidad sensorial, usada por separado crea problemas de acceso:

* Los CAPTCHA visuales crean una barrera de acceso a las personas que no ven.
* Los auditivos son inaccesibles para aquellas personas que no puedan oír.
* Los lógicos pueden presentar problemas para usuarios con problemas cognitivos ya que por ejemplo pueden no entender el idioma.

Por tanto se considera suficiente usar dos de estas modalidades.

#### Enlaces o botones que no tienen sentido fuera de contexto.

Una de las estrategias de interacción de las personas que usan lectores de pantalla es el uso del tabulador para recorrer los enlaces del sitio web.

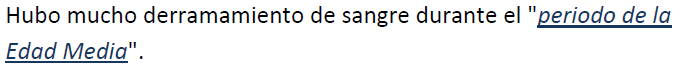
Si realizamos enlaces como el del siguiente ejemplo:



Y navegamos con el tabulador, el lector de pantalla nos leerá “pinche aquí”. Con esta información no podemos saber que en el destino del enlace hay más información sobre accesibilidad web.

Veamos unos consejos para que nuestros enlaces cumplan con su propósito y sean contenidos accesibles:

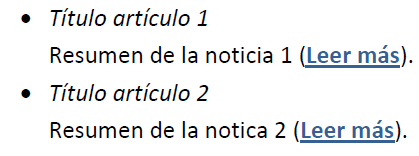
* **Usar el propio texto del enlace para describir su propósito**



El texto del enlace “periodo de la Edad Media”, describe su propósito.

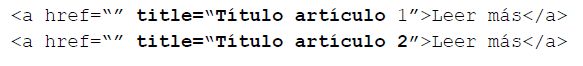
* **Proporcionar una descripción adicional al texto del enlace**

Ejemplo: Una web que contiene una colección de nuevos artículos.

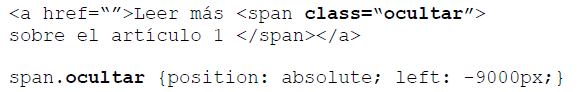


Los enlaces de acceso al contenido de la noticia son todos iguales “leer más”. Para hacerlos distinguibles a una persona que utilice un lector de pantalla hemos de proporcionar una descripción adicional al texto del enlace, para ello:

* + Podemos usar el **atributo title** para complementar la información

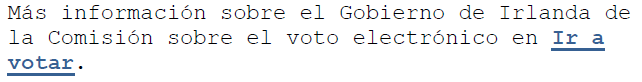


* + Otra opción es usar las hojas de estilo para **ocultar parte del texto del enlace**.



De esta forma lo que hacemos es enviar parte del contenido del enlace fuera de la pantalla, visualmente no lo veremos pero sí lo leerán los lectores de pantalla. Se suele usar esta opción porque a veces los lectores de pantalla tienen dificultad para la lectura de los titles

* **Si el texto del enlace no es suficientemente descriptivo pero sí que lo es la frase en la que se encuentra, hemos de proporcionar el enlace al final de la misma ya que mediante una combinación de teclas con el lector de pantalla, los usuarios son capaces de leer el párrafo actual**.

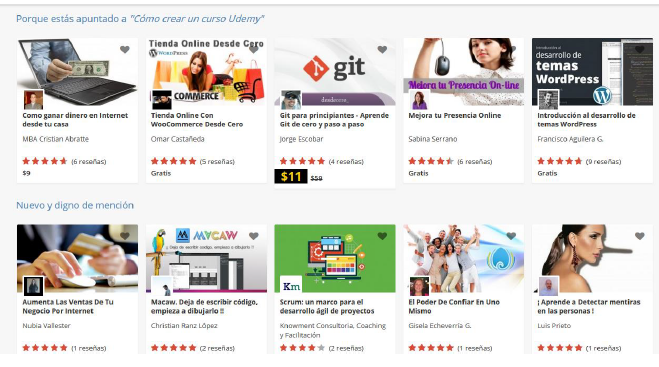


#### Imágenes sin un texto alternativo que las describa o un texto alternativo inapropiado

En primer lugar hemos de saber distinguir entre:

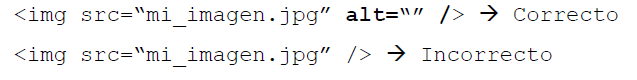
* una imagen decorativa,
* una imagen necesaria, es decir imágenes importantes que transmiten información relevante.

**Imágenes decorativas**



Esta imagen es la página principal de la web de Udemy. Todas las imágenes que acompañan a los cursos serían decorativas ya que no aportan información relevante. Actualmente todas ellas incluyen un texto alternativo que se redundante con el título del curso que está inmediatamente después de las imágenes. En este ejemplo, un lector de pantalla leería dos veces “Como ganar dinero en Internet desde su casa” creando ruido innecesario al usuario.

Una imagen decorativa es aquella que no transmite información importante, o no aporta nueva información a nuestro documento y por lo tanto hemos de incluir un texto alternativo vacío.



Es obligatoria la inclusión del atributo alt ya que, en otro caso las herramientas de evaluación mostrarán error.

**Imágenes necesarias**

Las imágenes necesarias por el contrario sí que aportan información a nuestro documento y por tanto hemos de incluir una alternativa textual. Este texto alternativo no ha de superar los 150 caracteres, en caso de que se necesite una descripción más larga hay que usar el atributo longdesc.

Veamos algunos ejemplos.

* Imagen que no posee información textual



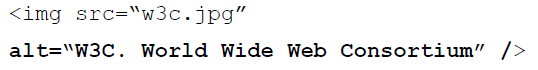
Imagen que no posee información textual pero sí que transmite información importante, transmite la idea de advertencia por lo que hemos de indicarlo a través del texto alternativo.



* Imagen que posee información textual.



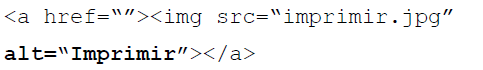
En las imágenes que posean información textual hemos de proporcionar como alternativa textual, el texto que aparece en ellas.



* Imagen que funciona como enlace.



Aquellas imágenes que funcionan como enlace, pondremos como alternativa textual la función que desempeña.



Si os fijáis, en el ejemplo no esta puesto como alternativa textual impresora sino la función de imprimir.

Por otro lado no hemos de ser redundantes, es decir, si tenemos el icono de la impresora y junto al texto imprimir. Por un lado incluiremos ambos elementos, icono y texto dentro del mismo enlace, dejando el texto alternativo vacío, para que un lector de pantalla no lea dos veces “Imprimir”.

Los fallos más comunes con las imágenes suelen ser:

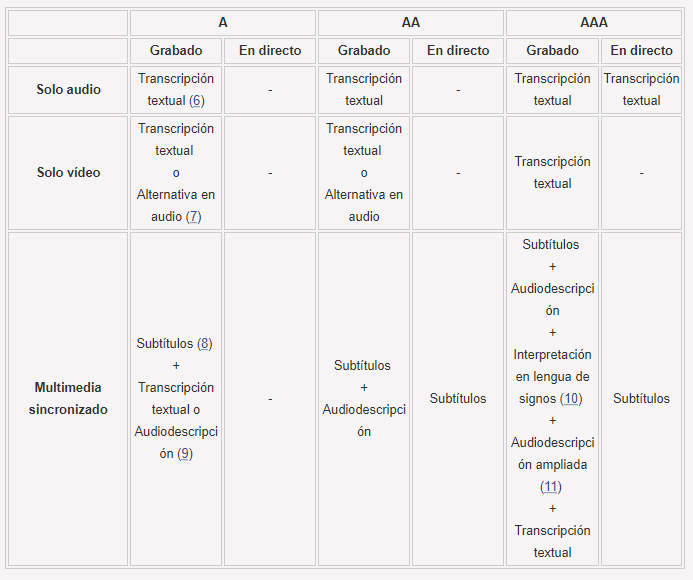
* Usar alternativas inapropiadas que no proporcionan la misma información o función, como por ejemplo usar nombres de archivos, texto de relleno o genéricos como por ejemplo poner “Foto” o “Imagen”, etc.
* No actualizar las alternativas textuales cuando cambia el contenido no textual, por ejemplo si tenemos una imagen con información meteorológica, la alternativa textual debe variar a la vez que la imagen.

Herramienta de ayuda para proporcionar textos alternativos accesibles:

<http://www.usableyaccesible.com/textosalternativosaccesibles/mapa_decision_texto_alternativo.php>

### AUDIO Y VIDEO ACCESIBLES

<https://olgacarreras.blogspot.com.es/2012/08/tabla-resumen-de-los-requisitos-de.html>



### OTROS REQUISITOS DE ACCESIBILIDAD RELACIONADOS CON EL CONTENIDO MULTIMEDIA

Aparte de lo visto en el apartado audio y video accesibles, en las WCAG 2.0 hay otros requisitos relacionados con el contenido multimedia.

**Audio**

* Criterio de conformidad 1.4.2 Control del audio. (Nivel A)

Si el audio de una página web suena automáticamente durante más de 3 segundos, se proporciona ya sea un mecanismo para pausar o detener el audio, o un mecanismo para controlar el volumen del sonido que sea independiente del nivel de volumen global del sistema.

* Criterio de conformidad 1.4.7 Sonido de fondo bajo o ausente. (Nivel AAA)

Para el contenido de solo audio grabado que contiene habla en primer plano, no es un CAPTCHA sonoro o un audiologo, y que no es una vocalización cuya intención principal es servir como expresión musical (canción), se cumple al menos uno de los siguientes casos:

* + Ningún sonido de fondo: el audio no contiene sonidos de fondo.
  + Apagar: los sonidos de fondo pueden ser apagados.
  + 20 dB: los sonidos de fondo son, al menos, 20 decibelios más bajos que el discurso en primer plano, con la excepción de sonidos ocasionales que duran solamente uno o dos segundos.

La herramienta gratuita Audacity te permitirá comprobar el contraste entre el ruido de fondo y el de primer plano.

**Destellos**

* Criterio de conformidad 2.3.1 Umbral de tres destellos o menos. (Nivel A)

Las páginas web no contienen nada que destelle más de tres veces en un segundo, o el destello está por debajo del umbral de destello general y de destello rojo. (Nivel A)

* Criterio de conformidad 2.3.2 (Nivel AAA)

No se permite que en las páginas haya ningún contenido que destelle más de tres veces por segundo)

<https://olgacarreras.blogspot.com.es/2007/01/cmo-evitar-causar-ataques.html>

**Tiempo suficiente**

* Criterio de conformidad 2.2.1 Tiempo ajustable. (Nivel **A**)

Para cada límite de tiempo impuesto por el contenido, se cumple **al menos uno** de los siguientes casos:

* + Apagar: el usuario puede detener el límite de tiempo antes de alcanzar el límite de tiempo.
  + Ajustar: el usuario puede ajustar el límite de tiempo antes de alcanzar dicho límite en un rango amplio que es, al menos, diez veces mayor al tiempo fijado originalmente.
  + Extender: se advierte al usuario antes de que el tiempo expire y se le conceden al menos 20 segundos para extender el límite temporal con una acción simple (por ejemplo, "presione la barra de espacio") y el usuario puede extender ese límite de tiempo al menos diez veces.
  + Excepción de tiempo real: el límite de tiempo es un requisito que forma parte de un evento en tiempo real (por ejemplo, una subasta) y no resulta posible ofrecer una alternativa al límite de tiempo; o
  + Excepción por ser esencial: el límite de tiempo es esencial y, si se extendiera, invalidaría la actividad.
  + Excepción de 20 horas: el límite de tiempo es mayor a 20 horas.
* Criterio de conformidad 2.2.3 (Nivel **AAA**)

No se permite que el tiempo sea parte esencial del evento o actividad presentado por el contenido, exceptuando los multimedias sincronizados y los eventos en tiempo.

* Criterio de conformidad 2.2.2 Poner en pausa, detener, ocultar. (Nivel **A**)

Para la información que tiene movimiento, parpadeo, se desplaza o se actualiza automáticamente, se cumplen todos los casos siguientes:

* + Para toda información que se mueve, parpadea o se desplaza, que comienza automáticamente, dura más de cinco segundos y se presenta en paralelo con otro contenido, existe un mecanismo para que el usuario la pueda poner en pausa, detener u ocultar, a menos que el movimiento, parpadeo o desplazamiento sea parte esencial de una actividad.
  + Para toda información que se actualiza automáticamente, que se inicia automáticamente y se presenta en paralelo con otro contenido, existe un mecanismo para que el usuario la pueda poner en pausa, detener u ocultar, o controlar la frecuencia de actualización a menos que la actualización automática sea parte esencial de una actividad.

### ENCABEZADOS

En enero de 2014, el WebAIM, realizó por quinta vez una encuesta sobre las preferencias de los usuarios de lector de pantalla.

En ella realizó la siguiente pregunta a los participantes: ¿Cuál de las siguientes acciones es más probable que realices primero al intentar encontrar información en una página web larga?

Las posibles respuestas eran:

* Navegar a través de los encabezados de una página.
* Usar la función “Buscar”.
* Navegar a través de los enlaces de una página.
* Navegar a través de los “landmark roles” de la página.
* Leer a través de la página

La mayoría de los usuarios contestaron “Navegar a través de los encabezados de la página”.

Navegar a través de los encabezados, es una forma de obtener un resumen de las ideas principales de la página y saltar entre contenidos para leer las partes que más les interesa. El principal inconveniente es que muchas páginas carecen de encabezados, y sin ellos este método de ojear el contenido es completamente inútil.

Por tanto, debemos usar los encabezados para proporcionar una estructura lógica, es decir, un esquema preciso a nuestros contenidos.

Los usuarios de lector de pantalla pueden saltar entre encabezados y sacar la lista de todos ellos, con combinaciones de teclas.

Facilitar la navegabilidad de nuestras páginas es uno de los propósitos del marcado de los encabezados recogido en el criterio de conformidad 2.4.6 Encabezados y etiquetas: Los encabezados y etiquetas describen el tema o propósito. (Nivel **AA**).

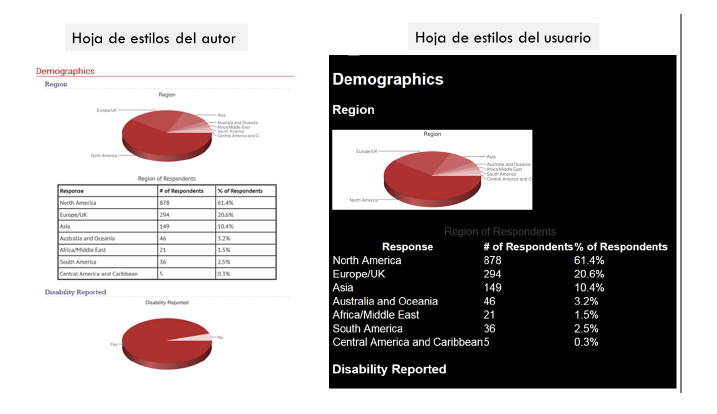
Lo más importante a destacar de este criterio de conformidad es que los encabezados han de ser claros y descriptivos para encontrar más fácilmente la información buscada y comprender la relación entre las distintas partes del contenido.

Si no tuviésemos los encabezados marcados en nuestro código html, y los simulamos a través de una hoja de estilos, si un usuario visualiza la página sin hoja de estilos o con otra adaptada a sus necesidades perdería esa información, vería todo el contenido del mismo modo, lo cual dificultaría su comprensión.

Esta es otra de las razones por la que hemos de marcar los encabezados, es decir, para que nuestros contenidos sean adaptables según está recogido en el criterio de conformidad 1.3.1 Información y relaciones: La información, estructura y relaciones comunicadas a través de la presentación pueden ser determinadas por software o están disponibles como texto (Nivel **A**).

Con este criterio de conformidad se pretende asegurar que la información y las relaciones implícitas en el formato visual o auditivo se mantengan cuando cambie el formato de la presentación.

Por ejemplo, el formato de la presentación cambia cuando un lector de pantalla lee el contenido o cuando se sustituye la hoja de estilo proporcionada por el autor, por la hoja de estilos del usuario, como en este ejemplo que se ha cambiado por una hoja de estilos de alto contraste, blanco sobre negro.



En HTML existen seis niveles de encabezados, que podemos definir con las etiquetas semánticas h1, h2, h3, h4, h5 y h6, siendo h1 el encabezado más importante de nuestra página web y el h6 el de menor importancia.

Hemos de anidarlos de forma correcta, es decir después de un encabezado de primer nivel puede ir uno de segundo nivel u otro de primer nivel pero no uno de tercer nivel.

Cada encabezado ha de contener texto, no pueden existir encabezados que no contengan contenido.

Si los estilos por defecto que poseen los encabezados no nos gustan, no es un problema porque los podemos cambiar con CSS y obtener la presentación de nuestro documento de la forma que queramos.

Para comprobar el anidamiento de los encabezados podéis usar la barra Web Developer Toolbar a través de la opción View Document Outline.

**Fallos más comunes**

Los fallos más comunes que se suelen cometer en el uso de los encabezados son:

* Anidarlos de forma incorrecta.
* Utilizar hojas de estilo CSS para simular los encabezados.
* Utilizar elementos semánticos h1-h6 para dar formato al texto: imaginaros una página html en la que tengo una frase que quiero destacar “Es muy importante que el aceite esté bien caliente”, si utilizo un encabezado de nivel cuatro <h4> para destacarla, cuando extraiga el esquema de mis contenidos aparecerá como si de una sección de mi documento se tratase y esto es incorrecto.

### OTROS ELEMENTO SEMÁNTICOS

En las Pautas de Accesibilidad al Contenido Web 2.0 y en el Criterio de Conformidad:

1.3.1 Información y relaciones: La información, estructura y relaciones comunicadas a través de la presentación pueden ser determinadas por software o están disponibles como texto. (Nivel A)

y en concreto en las técnicas:

G115: Usar elementos semánticos para marcar la estructura y H49: Usar marcado semántico para identificar el texto especial o enfatizado (HTML)

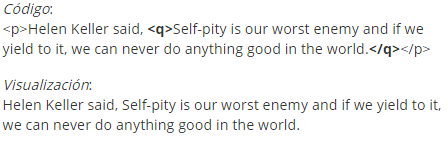
Nos indican, a través de una serie de ejemplos la necesidad incluir elementos semánticos y cómo incluirlos en nuestras páginas.

**Párrafos**

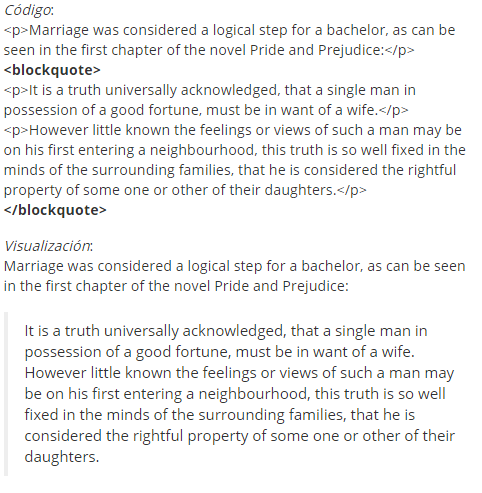
* Limitar los párrafos a única idea principal.
* No utilizar párrafos vacíos para separar contenidos ya que algunos lectores de pantalla leerán, "en blanco, en blanco, en blanco..." por cada párrafo vacío, generando ruido al usuario.

**Citas**

* Citas en línea: <q></q>



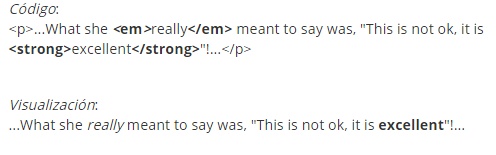
* Citas en bloque: <blockquote></blockquote>



No uses los elementos de cita para dar formato al texto. Un error muy común es utilizar la cita en bloque para dar sangría al texto.

**Énfasis**:

* Cursiva: <em>...</em>
* Negrita: <strong>...</strong>



Utiliza para enfatizar una o varias palabras estos elementos estructurales en vez de usar las propiedades de las hojas de estilos "font-style:italic" o "font-weight:bold".

### IDIOMA

El marcado del idioma empleado en un documento o sitio Web incrementa la legibilidad del mismo, beneficiando a todos los usuarios (no sólo a las personas con discapacidad).

En primer lugar facilita la lectura de los documentos en distintos idiomas por parte de los lectores de pantalla y programas de síntesis de voz, así los programas que leen el documento en braille podrán insertar los códigos adecuados donde ocurren los cambios de idioma, para asegurar que el programa de traducción a braille generará los caracteres correctos (por ejemplo, caracteres acentuados). También es útil para usuarios que no saben idiomas ya que obtendrán la traducción de los textos de idiomas desconocidos mediante programas de traducción.

En segundo lugar se consigue que los motores de búsqueda encuentren las palabras claves e identifiquen los documentos en el idioma deseado.

En nuestras páginas web hemos de marcar:

* El idioma de la página al inicio del documento
* El idioma de las partes, es decir cualquier cambio de idioma que se produzca a lo largo de nuestra página Web.

El atributo utilizado para marcar el idioma es lang

¿Hemos de marcar todos los cambios de idioma que se produzcan en nuestro documento? La respuesta es no.

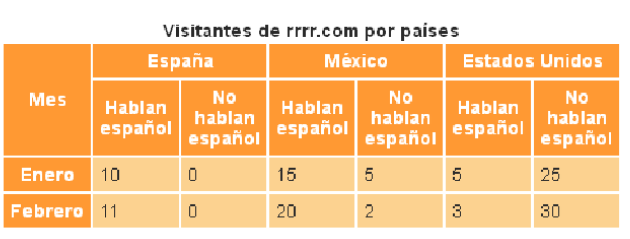
Hay una serie de excepciones:

* Nombres propios.
* Términos técnicos, títulos o términos de uso común: adoptados en el idioma con pronunciación igual o similar.

Por ejemplo: gourmet, homo sapiens, e-mail, mailing, chat, password, copyright, software, etc…

### TABLAS

Las tablas sirven únicamente para mostrar información tabular. La información tabular es aquella que permite visualizar, consultar y analizar datos.



Las tablas no sirven, para dotar de presentación a los contenidos de nuestro sitio web, y por lo tanto no debemos usarlas para maquetar nuestros sitios web.

Para entender la importancia de la programación accesible de las tablas de datos tenemos que analizar la forma de leer las tablas.

Como persona vidente, para analizar los datos de esta celda, leería: “En el mes de enero, en España 10 visitantes de rrrr.com hablaban español”. Y así sucesivamente con el resto de las celdas.

Un lector de pantalla, si nuestra tabla no es accesible, la leería de forma secuencial, es decir leería: “Visitantes de rrrr.com por países, mes, España, México, Estados Unidos, hablan español, no hablan español,…”. Resultando incomprensible para el usuario que utiliza un lector de pantalla.

Si nuestra tabla es accesible, leería:

* En primer lugar la etiqueta caption , es decir, el título de la tabla: “Visitantes de rrrr.com por países.”
* Por último leería la relación de las celdas que hemos indicado desde la programación, es decir en “España , Si, Enero = 10, España , No, Enero = 0, México , Si, Enero = 15, México , No, Enero = 5, EEUU, Si, etc… “.

En el Criterio de Conformidad 1.3.1 Información y relaciones hay una serie de técnicas que tenemos que tener en cuenta cuando tengamos que insertar en nuestros contenidos una tabla de datos:

* H51 Usar tablas para presentar información tabular.
* H39 Usar elementos caption para asociar títulos de tabla con las tabla de datos.
* H63 Usar el atributo scope para asociar celdas de encabezado y celdas de datos en las tablas de datos.
* H43 Usar atributos id y headers para asociar las celdas de datos con las celdas de encabezado en las tablas de datos.

**H51 Usar tablas para presentar información tabular.**

Las tablas de datos estructuran la información en filas y columnas describiendo una relación entre cada celda de datos con otras celdas en su misma fila y/o columna.

Esta información, que de forma visual se aprecia fácilmente, se debe transmitir también a los usuarios que deseen acceder a la tabla de forma no visual, es decir a través de un lector de pantalla.

Para construir una tabla de datos se precisa de ciertos elementos HTML básicos:

* El elemento table identifica toda la tabla y contiene el resto de elementos.
* El elemento tr actúa como contenedor para una fila de celdas.
* El elemento th define una celda que contiene información de encabezado.
* El elemento td define una celda que contiene datos.

**H39 Usar elementos caption para asociar las tablas de datos con su título.**

Esta técnica nos indica que debemos asociar las tablas de datos con su título. Cuando las tablas de datos tienen un título que identifica su contenido, hemos de marcarlo con el elemento caption. De esta forma se relaciona el título con la tabla de datos a nivel del marcado, y no sólo visualmente.

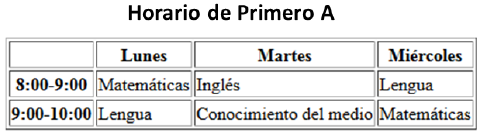
Además, este elemento permite a los lectores de pantalla saltar directamente a los títulos de las tablas y poder ojear el documento más fácilmente.

El elemento caption no es obligatorio pero es recomendable utilizarlo ya que es beneficioso para que los usuarios puedan ojear el documento.

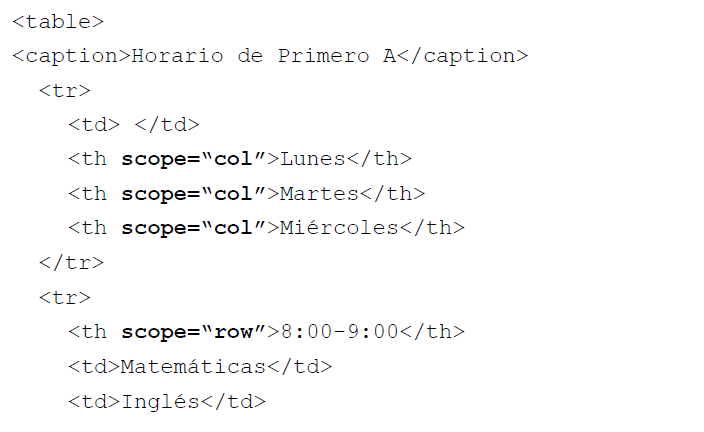
**H63 Asociar las celdas de encabezado y las celdas de datos en las tablas de datos mediante el atributo scope .**

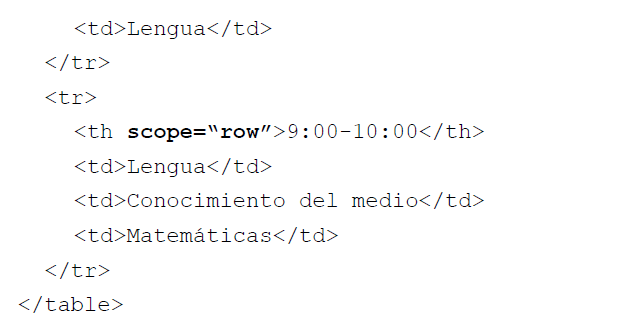
Vamos a explicar cómo se utiliza el atributo scope para asociar las celdas de encabezado y las celdas de datos en las tablas de datos.

Veamos un ejemplo



Código HTML correspondiente a la tabla:





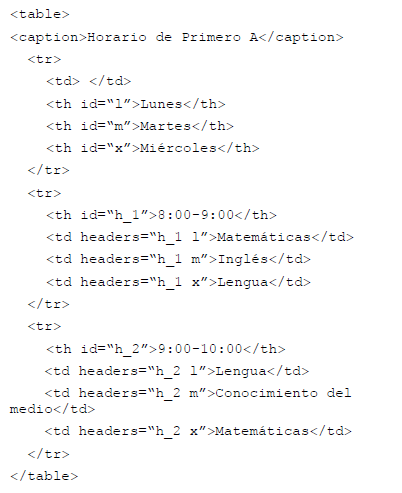
El atributo scope indica el ámbito sobre el que actúa el encabezado.

* Con el valor row para encabezados que actúan sobre las filas.
* Con el valor col para encabezados que actúan sobre las filas.

El problema es que existen agentes de usuario que no tienen compatibilidad con el atributo scope

**H43 Asociar las celdas de encabezado y las celdas de datos en las tablas de datos mediante los atributos id y headers**

Siguiendo con la tabla de ejemplo anterior, el código HTML utilizando esta técnica sería:



Es una tabla de datos compleja puesto que para una celda de datos tiene asociados dos encabezados. Así matemáticas tiene dos encabezados uno de columna lunes y uno de fila de las 8:00 a las 9:00.

Mediante los atributos id y headers los lectores de pantalla serán capaces de informar a los usuarios sobre cuáles son los encabezados correspondientes a la celda actual, independientemente de la complejidad de la tabla:

* id, se utiliza en las celdas de encabezado <th> para proporcionar un identificador que ha de ser único.
* headers se usa en las celdas de datos <td>, con el valor de los id correspondientes.

Una tabla de datos compleja, es aquella en la que una celda de datos se relaciona con más de un encabezado de fila y/o columna, y por tanto hemos de establecer la asociación entre las celdas de datos con sus celdas de encabezado, utilizando la técnica H63 o la técnica H43.

### REDACCIÓN DEL CONTENIDO

A continuación incluimos una relación de requisitos de accesibilidad relacionados con la redacción del contenido.

**Legible**

Criterio de conformidad 3.1.3 Palabras inusuales. (Nivel AAA)

Se proporciona un mecanismo para identificar las definiciones específicas de palabras o frases usadas de modo inusual o restringido, incluyendo expresiones idiomáticas y jerga.

Criterio de conformidad 3.1.4 Abreviaturas. (Nivel AAA)

Se proporciona un mecanismo para identificar la forma expandida o el significado de las abreviaturas.

Evita en la medida de lo posible el uso de abreviaturas, especialmente si no son de uso común. Intenta utilizar siempre que puedas proporcionar la expansión antes del acrónimo o la forma abreviada, al menos la primera vez que la utilices en la página. Por ejemplo: Web Content Accessibility Guidelines (WCAG)

Criterio de conformidad 3.1.5 Nivel de lectura. (Nivel AAA)

Cuando un texto requiere un nivel de lectura más avanzado que el nivel mínimo de educación secundaria una vez que se han eliminado nombres propios y títulos, se proporciona un contenido suplementario o una versión que no requiere un nivel de lectura mayor a ese nivel educativo.

Criterio de conformidad 3.1.6 Pronunciación. (Nivel AAA)

Se proporciona un mecanismo para identificar la pronunciación específica de las palabras cuando el significado de esas palabras, dentro del contexto, resulta ambiguo si no se conoce su pronunciación.

Hay que indicar que en español las palabras no tienen diferente significado según su pronunciación, siempre y cuando la ortografía sea correcta. No es lo mismo "médico", "medíco" o "medicó". Por ello, una ortografía incorrecta puede ser un grave problema de accesibilidad, ya que los lectores de pantalla verbalizan el contenido tal cual está escrito, pudiendo dar lugar a pronunciaciones ininteligibles o con significado alterado.

**Características sensoriales**

Criterio de conformidad 1.3.3 Características sensoriales. (Nivel A)

Las instrucciones proporcionadas para entender y operar el contenido no dependen exclusivamente en las características sensoriales de los componentes como su forma, tamaño, ubicación visual, orientación o sonido.

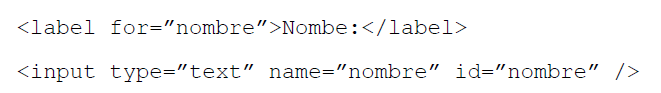
Así, en vez de usar mensajes del estilo "Pulse el botón cuadrado de la derecha" o "Pulse en la flecha roja de la izquierda", los sustituiríamos por mensajes del tipo "Pulse el botón con la etiqueta IR" o "Pulse el botón con forma de flecha y la etiqueta siguiente".

### FORMULARIOS

Vamos a repasar los principales requisitos de accesibilidad que deben cumplir los formularios web.

**Etiquetar correctamente los controles del formulario**

El elemento label permite indicar semánticamente qué texto funciona como etiqueta de un campo de formulario.



Para establecer la relación entre la etiqueta y el control del formulario debemos asociarla explícitamente con el atributo for, cuyo valor debe ser igual al del id del control del formulario, id, que a su vez, debe ser único en la página.

El elemento label debe usarse con los elementos: “textarea”, “select”, y los “input” de tipo: “text”, “checkbox”, “radio”, “file” y “password”.

Por el contrario, el elemento label no debe usarse con los “input” de tipo “image”, que se etiquetan utilizando el atributo alt.



La etiqueta label tampoco debe usarse en los “input” de tipo “submit” o “reset”, los cuales se etiquetan con su atributo value.



Tampoco debe usarse con los “input” de tipo “hidden”, que no se etiquetan, ni con el elemento “button”, en el cual se usa su contenido como etiqueta.

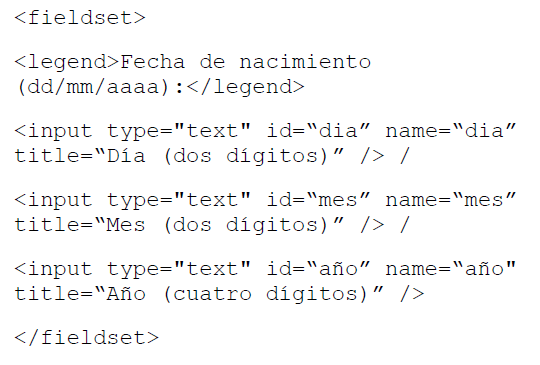


¿Dónde situamos la etiqueta respecto al campo?

* Antes del campo, bien encima, bien a la izquierda, con los elementos “textarea”, “select”, y los “input” de tipo “text”, “file” y “password”.
* Detrás del campo con los “input” de tipo “radio” y “checkbox”.

Cuando no podemos usar el elemento label para etiquetar un control de formulario, porque en el diseño no existe un texto que podamos etiquetar como tal o mostrar una etiqueta podría generar confusión, las WCAG 2.0 permiten utilizar el atributo title para etiquetar los controles del formulario e identificar su propósito.

Es habitual utilizar esta técnica cuando hay varios campos seguidos para introducir partes de un dato, como una fecha, o un número de cuenta o tarjeta.



Otro ejemplo habitual es usar title para etiquetar el campo del buscador de la página.

Es importante mantener la consistencia de los nombres a lo largo del sitio. Los campos que cumplen una misma función deben etiquetarse igual, por ejemplo, no podemos etiquetar en una página el campo “contraseña”, como “contraseña” y en otra página como “password”.

**Usar controles estándar**

La primera norma es utilizar controles de formulario estándar (<input>, <textarea>, <select>, <button>, etc.) siempre que sea posible y hacerlo conforme a la especificación de HTML5.

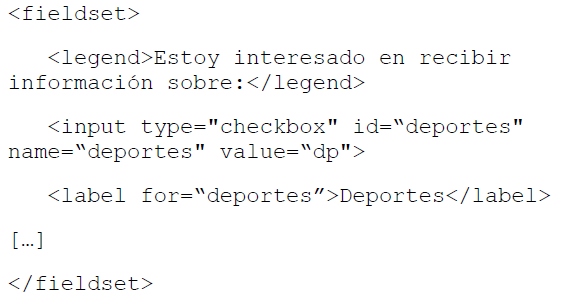
**Estructura y relaciones**

También es importante informar semánticamente de la estructura y las relaciones entre los campos que se están comunicando a través de la presentación, para que puedan ser determinadas por software, es decir, que las aplicaciones (como los productos de apoyo) pueden extraer y presentar este tipo de información de distintas maneras.

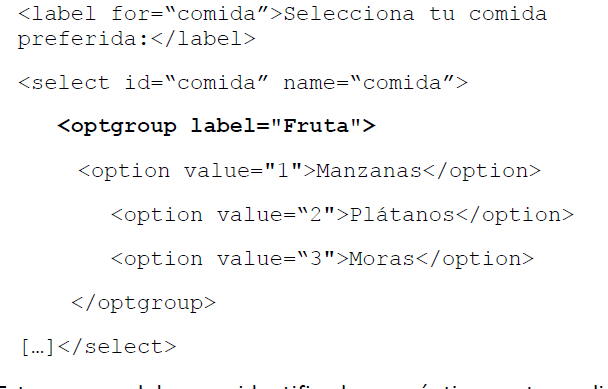
Para ello, además de aplicar los requisitos que hemos visto en relación con el correcto etiquetado de los campos, podemos usar elementos como fieldset y optgroup.

La etiqueta fieldset nos permite agrupar semánticamente una serie de controles de formulario y etiquetarlos en su conjunto con una descripción del grupo, que incluimos mediante el elemento legend, y de esta manera comprender la relación entre los controles e interactuar con el formulario de forma más rápida y efectiva.

El uso de fieldset es especialmente importante en los grupos de radios y checks. En estos casos la etiqueta individual de cada uno no transmite plenamente el contexto descriptivo del grupo, y es esencial que se agrupen para facilitar que sean tratados semánticamente como un único conjunto con una descripción adicional a nivel de grupo.



Otro elemento que nos permite informar semánticamente de la relación entre una serie de elementos es optgroup. Permite agrupar una serie de options dentro de una select, de selección múltiple o no, porque a menudo las listas de selección tienen grupos de opciones relacionadas.



Además, optgrouop ayuda a romper visualmente las listas largas y a que los usuarios puedan localizar más fácilmente la opción deseada.

**Cambios de contexto**

Por último, los controles de formulario no pueden provocar un cambio de contexto, como abrir otra página o ventana, mover el foco a otro componente o enviar un formulario, de forma automática o sin conocimiento del usuario, ni cuando el control coja el foco, ni al entrar datos o cambiar su estado.

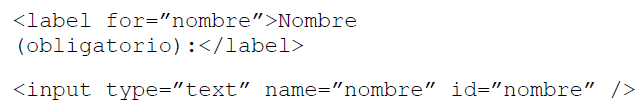
Por ejemplo, el formulario se debe enviar con un botón “submit”, que permite a los usuarios solicitar explícitamente enviar los datos:



O por ejemplo, si tenemos una “select” que permite navegar a una página diferente en función de la opción seleccionada, no podemos hacerlo automáticamente cuando el usuario seleccione la opción, sin avisarle antes, sino que preferiblemente incluiremos junto a la “select” un botón para que sea el usuario el que lo solicite explícitamente.

**Identifica los campos obligatorios**

En primer lugar debemos identificar los campos obligatorios, preferiblemente con un texto en la etiqueta del campo, y por tanto antes del mismo.



No podemos identificar los campos obligatorios solo con el color, por ejemplo con el color del propio campo o de su etiqueta, si acaso puede ser una pista visual adicional. Si te basas solo en el color para identificar los campos obligatorios, esta información pasará desapercibida para muchas personas, por ejemplo, a aquellas con problemas para percibir el color; o por ejemplo a aquellas que acceden con un producto de apoyo, como un lector de pantalla o una línea braille.

**Informa del formato requerido**

Otra manera de evitar errores a los usuarios es informándoles de las restricciones en el formato de datos que deben introducir, aunque debería intentarse que los formularios fueran flexibles para introducir diferentes formatos de datos.

Por ejemplo, se puede indicar el formato requerido en la etiqueta del campo y proporcionar un ejemplo bajo el mismo.

**Prevenir errores legales, financieros y de datos**

Por otro lado, el criterio de conformidad “3.3.4 Prevención de errores (legales, financieros, datos)” de las WCAG 2.0 trata sobre cómo prevenir errores legales, financieros y de datos para las páginas que:

* Representan para el usuario compromisos legales o transacciones financieras;
* Modifican o eliminan datos controlables por el usuario en sistemas de almacenamiento de datos;
* Envían las respuestas del usuario a una prueba.

En estos casos se debe cumplir al menos con uno de estos requisitos:

* El envío es reversible.
* Se verifica la información para detectar errores en la entrada de datos y se proporciona al usuario la oportunidad de corregirlos.
* Se proporciona un mecanismo para revisar, confirmar y corregir la información antes de finalizar el envío de los datos.

Para alcanzar el nivel AAA debe cumplirse alguno de estos requisitos en todas las páginas que requieren del usuario envío de información.

**Mejorar la accesibilidad de los mensajes de validación**

Vamos a ver ahora las técnicas relacionadas con la información que se ofrece cuando el usuario comete un error en el formulario. Veamos cómo hacer más accesibles los mensajes de validación de los formularios

En primer lugar es necesario identificar y describir los errores, bien porque no se han rellenado campos obligatorios, bien porque no se respeta determinado formato o valor y hacerlo mediante texto. Complementariamente, si se desea, se puede apoyar esta información con colores, estilos de texto o imágenes.

El objetivo es asegurarse de que todos los usuarios se percatan de que se ha producido un error y dónde.

¿Cómo lo hacemos?

Preferiblemente, aunque se admiten otras opciones (como ventanas de alertas o información contextual):

* Informando mediante texto, al comienzo del formulario, de los errores existentes.
* Describiendo el problema e identificando los campos con error (sin basarse solo en el color).
* Si es posible, indicando o sugiriendo valores posibles.

Ten en cuenta que:

* La validación debería ser realizada tanto en cliente como en servidor.
* El mensaje de error al comienzo de la página debe coger el foco para que no pase desapercibo a ningún usuario.
* Si la validación es en cliente, mediante JavaScript, y por tanto los mensajes se crean dinámicamente, deben insertarse correctamente con funciones DOM. De esta manera, los agentes de usuario pueden acceder al DOM para recuperar el nuevo contenido y en el orden correcto.
* Las zonas de la página que cambian dinámicamente, sin intervención del usuario, como es este caso, deben anunciarse al usuario de productos de apoyo para que no le pasen desapercibidos los cambios en las mismas. Esto se hace mediante WAI-ARIA.

### DOCUMENTOS ELECTRÓNICOS

Un documento electrónico es todo aquel que no está en un soporte tradicional como el papel, que creamos con una herramienta informática, o bien digitalizamos por ejemplo al escanearlo, y luego lo consumimos desde un ordenador, una tablet, un móvil, etc.

Por tanto, documentos electrónicos son una página web, un PDF, un documento de texto, un email, una presentación, una hoja de cálculo, etc.

Cuando hablamos de hacer sitios web accesibles, se entiende que debemos hacer las páginas HTML y sus contenidos accesibles, pero una duda habitual es si esto incluye también a los documentos electrónicos que enlazan desde sus páginas, como un PDF o un documento Word. La respuesta es “Sí”.

La información que se incluye dentro de estos documentos forma parte del contenido del sitio y por tanto deben ser igualmente accesibles.

Las WCAG 2.0 incorporan un concepto nuevo respecto a la versión anterior, las WCAG 1.0, que es importante conocer y comprender. Es el concepto de “compatible con la accesibilidad”. Simplificando, una tecnología compatible con la accesibilidad (como puede ser HTML, PDF) es aquella que dispone de los mecanismos necesarios para proporcionar información de accesibilidad a las aplicaciones de usuario y a los productos de apoyo (como un lector de pantalla) los cuales a su vez son capaces de comprender estos mecanismos y proporcionar dicha información a los usuarios que la requieran.

Las WCAG 2.0 permiten usar cualquier tecnología que sea compatible con la accesibilidad siempre y cuando:

* Se use de forma accesible, es decir compatible con los productos de apoyo;
* Los agentes de usuario y productos de apoyo soporten dicha tecnología.

Por ejemplo, PDF es una tecnología compatible con la accesibilidad, igual que una página HTML. Por tanto no es necesario dar ninguna alternativa al PDF en otro formato, basta con hacer los PDF accesibles nativamente de la misma manera que hacemos accesible de forma nativa una página HTML.

Las WCAG 2.0 son neutrales, independientes de una tecnología concreta, y por tanto aplicables a todas ellas. Por ejemplo, el criterio 1.4.3 “Contraste”, que veremos en una lección posterior, indica que el color del texto tiene que tener suficiente contraste con el color del fondo. Este criterio se aplica igual en cualquier tipo de documento, no hay técnicas específicas para una tecnología u otra.

Pero hay otros criterios de conformidad de las WCAG 2.0 que se cumplirán aplicando diferentes técnicas según la tecnología utilizada.

Por ejemplo, el criterio de conformidad 2.4.2 indica que todos los documentos deben tener un título que describa su temática o propósito. La manera de incluir el título será diferente según la tecnología utilizada. En HTML el título se incluirá con el elemento title, pero en un PDF, o en otros documentos, se especificará en las propiedades del documento. Aunque las características que debe tener el título son iguales en todas ellas: ser conciso, comprensible fuera de contexto, etc.

En estos casos, por cada criterio de conformidad, en el documento “Comprender las WCAG 2.0”, nos indicarán la técnica a aplicar en función de la tecnología utilizada, si es HTML o si es PDF (donde también incluyen a menudo la forma de aplicarlo en Word).

## NAVEGACIÓN ACCESIBLE

Una navegación más accesible tiene por tanto como objetivo:

* Que el usuario encuentre lo que necesita
* Que el usuario en todo momento pueda contestar a las preguntas:
  + ¿Dónde estoy? (ubicación y orientación)
  + ¿A dónde puedo ir? (navegación)

Una navegación más accesible mejorará no solo la accesibilidad de la página sino también su usabilidad. Beneficia a todos los usuarios, pero especialmente a los usuarios que acceden con un lector de pantalla o una línea braille, y a los usuarios con discapacidad cognitiva, facilitándoles la navegación y la localización de la información.

Para lograr una navegación más accesible, los criterios de conformidad de esta pauta hacen referencia a:

* Saltar contenido repetido.
* Identificar el contenido. Identificar correctamente el contenido permite que la página sea más fácil de comprender y navegar. Esto implica un título para cada la página, o incluir encabezados y etiquetas que describan correctamente el tema o propósito.
* Describir adecuadamente el propósito de los enlaces. Los enlaces son uno de los elementos de la navegación más importantes.
* Información sobre la localización del sitio, permitiendo más de un camino para localizar una página (múltiples vías) y proporcionando información acerca de la ubicación del usuario dentro de un conjunto de páginas (ubicación).
* Orden adecuado del foco y visibilidad del mismo, que asegure que los usuarios pueden interactuar mediante el teclado, el foco se puede localizar visualmente, y es posible navegar secuencialmente por el contenido y encontrar la información en el orden correcto para conseguir que las páginas sean predecibles y el usuario no se desoriente.

### EL TÍTULO DE LA PÁGINA

Todos los documentos HTML deben tener un elemento title en la sección head.

Si el título de una página web no es significativo, no es adecuado, se pueden dar situaciones confusas. Cuando una página web se carga la primera vez, un lector de pantallas anuncia el título de la página. Si el título de nuestra página “Página 1”, cuando lo anuncie el lector de pantallas no le proporcionará mucha información al usuario. Además, un usuario ciego que vaya saltando de una pestaña a otra, tendrá bastantes dificultades para identificar el contenido de cada una de las páginas web que se están visualizando en el navegador.

Un título adecuado no sólo beneficia a los usuarios ciegos, beneficia a otros grupos de usuarios con discapacidad, ya que un título adecuado ayuda a todos los usuarios a identificar si el contenido de una página web es relevante para el usuario.

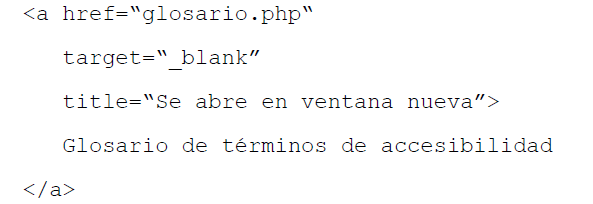
En vez de usar un título como "Introducción", que no proporciona mucha información acerca del contexto, los autores deberían poner en su lugar un título del estilo "Introducción a la apicultura medieval".

El título de una página web realiza varias funciones.

* Cuando se realiza una búsqueda en un buscador como Google el título de la página se emplea para identificar los resultados. Además, un buen título puede ayudar a mejorar el posicionamiento, el SEO de una página web.
* Cuando en un navegador tenemos abiertas varias pestañas, el título de la página ayuda a identificar qué página web se está mostrando en cada pestaña.
* El título de la página web también se emplea para definir el nombre por defecto de un marcador o favorito en el navegador web. Un título significativo ayuda a localizar fácilmente una página web en la lista de marcadores o favoritos del navegador.

### ENLACES ACCESIBLES

Los enlaces son la base de la navegación de nuestras páginas, por tanto es fundamental asegurarnos de que sean perceptibles, operables y comprensibles por todos los usuarios.



Un enlace se incluye con la etiqueta a. El destino del vínculo, es decir, la localización del recurso al que enlaza, se indica con el atributo href. En la mayoría de los navegadores, el destino del enlace se muestra en la barra de estado del navegador, incluso si la barra de estado no está visible.

Opcionalmente, el elemento <a> puede tener otros atributos, entre los que nos interesa resaltar ahora:

* target: por defecto, un enlace se abre en la misma ventana. El atributo target=“\_blank” permite abrirlo en ventana nueva.
* title: que permite incluir información adicional sobre el enlace. En la mayoría de los navegadores el contenido de este atributo se visualiza como un tooltip al colocar el cursor sobre el enlace, pero no se visualiza al coger el foco con el teclado.

Repasemos ahora cómo acceden a los enlaces de las páginas los usuarios de lector de pantalla. Para ello imaginemos que tenemos en la página un enlace “Mapa web”:



Si el usuario está haciendo una lectura lineal mediante las flechas del teclado, al tratarse de un enlace estándar incluido mediante el elemento nativo <a>, el lector de pantalla lo reconocerá y le anunciará al usuario que es un enlace, diciendo “enlace: mapa web”.

El usuario de lector de pantalla también puede ojear los enlaces. Puede recorrerlos pulsando el tabulador, puesto que los enlaces son uno de los elementos de HTML que pueden coger el foco por teclado.

Por último, el usuario de lector de pantalla puede sacar una lista con los enlaces de la página. En la ventana de listado de enlaces, cada enlace se lista con su texto de enlace. Además, en esta ventana, los usuarios disponen de diferentes opciones, según el lector de pantalla que utilicen, para ordenarlos, filtrarlos, ir a un enlace, etc.

Teniendo en cuenta todo ello, vamos ahora a repasar cuáles son las características de accesibilidad que harán que un enlace sea accesible, esto es, que sea perceptible, operable y comprensible por todos los usuarios.

#### Enlaces perceptibles

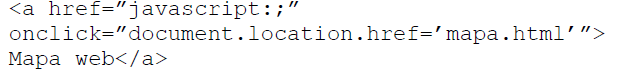
Para que los enlaces se puedan percibir tienen que cumplir dos requisitos:

* Tener un color de texto que ofrezca suficiente contraste con el color del fondo;
* La información de que es un enlace no debe depender únicamente del color. Imaginemos que tenemos un texto con varios párrafos, y en el interior de uno de estos párrafos un enlace definido solo por el color rojo, sin subrayar. Una persona daltónica tendrá dificultades para distinguirlo del resto del texto que lo rodea, y seguramente todos nosotros en determinados contextos, por ejemplo desde un dispositivo móvil con bajo contraste. Por ello, los enlaces incluidos dentro de un texto más amplio, como este, deben estar siempre subrayados para que sea sencillo distinguirlos del texto que los rodea.
* Es importante diferenciar visualmente los enlaces visitados de los no visitados, definiendo un estilo diferente en la CSS, para facilitar la navegación. Los usuarios de lector de pantalla, aunque no puedan ver esta diferenciación visual, tienen otros mecanismos.

#### Enlaces operables

Para que los enlaces sean accesibles, es necesario que sean operables por todos los usuarios en todos los contextos de uso. No deben depender de JavaScript

Un ejemplo incorrecto de enlace, porque depende innecesariamente de JavaScript, sería:



Un ejemplo correcto de enlace que no depende de JavaScript sería:



#### Enlaces comprensibles

Para que un usuario comprenda un enlace, es decir, el contenido o información que se encuentra en el documento de destino, o la acción o funcionalidad que invoca, y en base a ello pueda decidir si desea pulsarlo o no, el propósito del enlace debe quedar claro, preferiblemente por su propio texto de enlace, sin depender del contexto en el cual se inserta.

Los usuarios de productos de apoyo, como un lector de pantalla, pueden recorrer y ojear los enlaces de una página con el tabulador, pulsando una tecla concreta o sacando un listado de enlaces. Por tanto, si el texto del enlace no es descriptivo, deben consultar el texto que le sigue o precede; o en el caso del listado de enlaces no tienen dicho contexto para comprender su propósito.

Por ello, en las WCAG 2.0 encontramos dos criterios de conformidad que hacen referencia al propósito de los enlaces:

* El 2.4.4, de nivel A, que indica que el propósito del enlace debe quedar claro mediante el propio texto del enlace o bien por medio del texto del enlace sumado a su contexto determinado por software.
* El 2.4.9 de nivel AAA, en el cual ya no se admite que el propósito del enlace pueda ser determinado por su contexto, pues como hemos dicho, nuestro objetivo debería ser siempre que el texto del enlace fuera suficientemente descriptivo por sí mismo.

**Enlaces descriptivos**

Enlaces como “más información”, “pinche aquí”, “leer más”, “ver otros”, o unos puntos suspensivos no son textos de enlaces descriptivos, no permiten conocer el propósito del enlace fuera de contexto.

Por otra parte, imaginemos que cada uno de los enlaces “Más información” navega a una URL diferente. No podemos diferenciarlos fuera de contexto.

Por ello, además de tener textos descriptivos fuera de contexto, debes recordar que:

* Los enlaces que tienen el mismo texto de enlace deben tener siempre el mismo destino
* Dos enlaces que tienen el mismo destino tienen que tener el mismo texto de enlace.

Un texto de enlace descriptivo indica:

* El formato y tamaño del recurso al que enlaza, cuando este no es una página web (por ejemplo un archivo PDF);
* Si se abre en ventana nueva (porque incluye el atributo target=”blank”);
* Si navega a un sitio web diferente;
* Si abre otro programa, por ejemplo el gestor de correo del usuario cuando el destino del enlace es un email.

La URL del enlace no se considera descriptiva y por tanto no debe ser el texto del enlace







Es importante que la URL sea lo más comprensible posible, usando nombres de directorio que reflejen la estructura del sitio y nombres de páginas que identifiquen su contenido. Esto puede dar información adicional a los usuarios, no solo a los usuarios de lector de pantalla, sino a todos, puesto que la URL de destino se muestra en la barra de estado, y es especialmente útil cuando otros sitios enlazan al nuestro con enlaces poco descriptivos. Además, es un factor que influye en el posicionamiento de nuestras páginas.

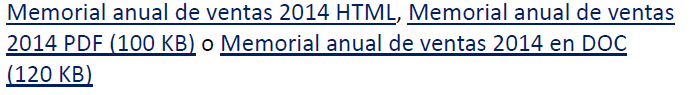
Existen medios para lograr que un texto de enlace, que por defecto no es completamente descriptivo, lo sea sin depender de su contexto.

* Ocultar parte del texto del enlace con CSS
* Incluir un control que permita cambiar el texto de los enlaces de cortos a largos.

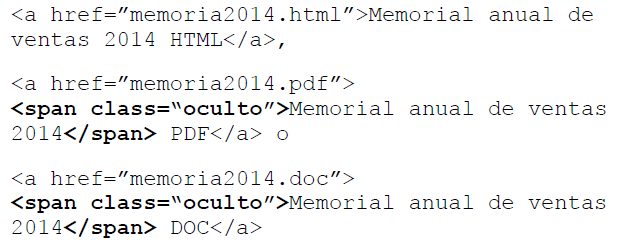
Imaginemos que tenemos un enlace al “Memorial anual de ventas 2014 en HTML”, y a continuación los enlaces a la versión en PDF y DOC:



Los enlaces “PDF” y “DOC” no son descriptivos fuera de contexto. Sin embargo, incluir el texto completo de cada enlace puede no ser la mejor solución para todos los usuarios, puesto que puede dificultar su lectura y comprensión:



En estos casos, podemos ocultar parte del texto del enlace mediante la hoja de estilo.





De esta forma, los navegadores gráficos con soporte de CSS solo mostrarán la porción del texto del enlace que queremos, mientras que el resto del texto estará disponible para los usuarios de productos de apoyo como un lector de pantalla.

De esta manera los usuarios de lector de pantalla se benefician de tener textos de enlace descriptivos fuera de contexto, pero no dificultamos la lectura y compresión de los enlaces para otros usuarios.

Esa técnica es muy recomendable para los típicos enlaces “leer más” asociados a cada noticia o evento en muchos portales.

Existen diferentes maneras de ocultar contenido mediante CSS y que no todas son adecuadas en este caso.

* Oculto visualmente pero disponible para los lectores de pantalla
  + Posicionando el texto fuera de pantalla con “text-indent”
  + Técnica “clip”.
* Ocultar visualmente y para los lectores de pantalla: “display: none” o “visibility:hidden”

**Enlaces descriptivos en contexto**

Aunque lo ideal, como decimos, es intentar que el texto del enlace sea descriptivo fuera de contexto, el criterio de conformidad 2.4.4 de las WCAG 2.0, que es necesario cumplir para alcanzar el nivel de conformidad A, también permite que el propósito del enlace pueda ser determinado a través del texto del enlace sumado a su contexto determinado por software.

¿Qué quiere decir que el propósito del enlace pueda ser determinado a través del texto del enlace sumado a su contexto determinado por software? Significa que la información adicional al texto del enlace, para que este sea descriptivo, se puede obtener a partir de su relación con el enlace, y esta información se puede presentar al usuario, combinada con el texto del enlace, de modo que las aplicaciones de usuario, incluyendo los productos de apoyo, puedan extraer dicha información y presentarla al usuario de diferentes maneras.

Contextos que se pueden determinar por software

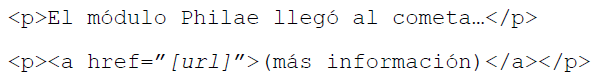
* Texto de la frase que contiene el enlace

Uno de los contextos que podemos usar para aclarar el propósito del enlace es el texto de la frase que lo contiene, puesto que los lectores de pantalla tienen combinaciones de teclas que les permiten leer la frase o el párrafo actual.

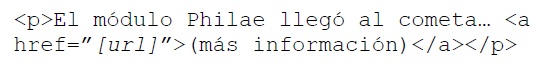


La información de contexto que aclara el propósito del enlace debe estar situada antes del mismo, de lo contrario resultaría confuso para los usuarios de lector de pantalla que leen el contenido linealmente.

* El párrafo <p> que contiene al enlace. Los lectores de pantalla tienen combinaciones de teclas que les permiten leer la frase o el párrafo actual. Un fallo habitual es incluir el enlace no descriptivo (por ejemplo “más información”) en un párrafo diferente al que nos está dando la información adicional para comprenderlo.

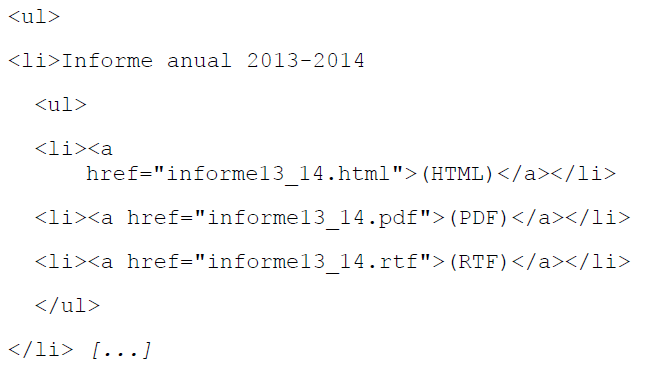


Lo correcto es que se sitúen en el mismo párrafo



* El elemento <li> que contiene el enlace o el elemento <li> padre

Otro de los contextos posibles será el elemento de lista que contiene el enlace, o en el caso de listas anidadas, el elemento de la lista padre que contiene la lista en la que se encuentra el enlace.



En esta lista anidada tenemos tres enlaces no descriptivos: HTML, PDF y RTF. La información adicional nos la ofrece el elemento <li> de la lista padre que contiene la lista de enlaces no descriptivos, es decir “Informe anual 2013-2014”. Los usuarios de lector de pantalla tienen combinaciones de teclas para recorrer los elementos de una lista. Como en los demás casos, es preferible que la información adicional para comprender el propósito del enlace esté antes del mismo.

* La celda <td> que contiene el enlace y sus encabezados <th>

Otro de los contextos posibles será la celda de la tabla que contiene el enlace junto con sus celdas de encabezado, puesto que los usuarios de lector de pantalla también tienen combinaciones de teclas para interactuar con las tablas de datos.

* Atributo title: la información del title se muestra al colocar el cursor sobre el enlace pero no cuando coge el foco por teclado, es decir, cuando tabulamos hasta él. El problema es que no todos los lectores de pantalla leen el texto del enlace más el contenido de este atributo. Por tanto, el atributo title solo se debe utilizar para proporcionar información adicional y nunca información relevante.
* Atributos de WAI-ARIA

Señalar que las WCAG 2.0 indican que la necesidad de clarificar el propósito de los enlaces tiene como excepción: “aquellos enlaces cuyo propósito es ambiguo para todos los usuarios, ya que las personas con discapacidad no tendrán ninguna desventaja frente al resto de usuarios”.

Las WCAG 2.0 ponen como ejemplo un juego en el que haya que elegir una puerta y las posibles respuestas son tres enlaces: puerta 1, puerta 2 y puerta 3.

En este caso no podemos dar más información sobre la página de destino de estos enlaces o la acción que invocan, porque la ambigüedad es intrínseca a la naturaleza de estos enlaces, al suspense del juego, y las personas con discapacidad no tienen ninguna desventaja frente al resto de los usuarios.

### NAVEGACIÓN A SALTOS

Cuando navegamos en un sitio web y vamos saltando de una página a otra es muy normal que encontremos unas partes comunes, como la cabecera y el menú principal, que se repiten de una página a otra.

Cuando se navega de forma visual por una página web, es posible fijar la mirada directamente en la parte del contenido principal que cambia de una página a otra, sin fijarnos en aquello que se repite de una página a otra. De este modo, la cabecera, el menú principal y otras partes comunes a todas las páginas de un sitio web desaparecen ante nuestros ojos.

Sin embargo, para una persona ciega que usa un lector de pantallas, la consulta de una página web es lineal, de arriba hacia abajo y supone pasar por todo el contenido de la página web hasta llegar al contenido principal que es lo que busca cuando navega de una página a otra.

Las Pautas de Accesibilidad al Contenido en la Web 2.0, proporcionan la solución en el criterio de conformidad 2.4.1 Evitar bloques, que dice: Existe un mecanismo para evitar los bloques de contenido que se repiten en múltiples páginas web.

Normalmente emplearemos los enlaces “saltar a” para saltar a secciones concretas de la página que suelen tener un encabezado.

La página del Web Accessibility Initiative (WAI) presenta un enlace con el texto “Skip to Content” , “saltar al contenido”, al principio de la página. Si revisamos el código fuente, vemos que el enlace conduce a un destino en la propia página etiquetado con “skip” en el atributo href. Este destino está situado un poco más abajo en el contenido principal de la página que comienza con el encabezado h1 “Web Accessibility Initiative”.

Muchas veces estos enlaces “saltar a”, por defecto, no están visibles, aunque sí que están disponibles para los usuarios que utilizan un lector de pantallas.

Unos consejos:

* Los enlaces “saltar a” al principio de la página, debe ser lo primero que se encuentre un usuario.
* No abuses, dos o tres enlaces “saltar a” son suficientes.
* No seas creativo con el texto de los enlaces, utiliza lo que la mayoría de la gente utiliza, como “saltar navegación” o “saltar a contenido principal”.

### NAVEGACIÓN POR TECLADO

Uno de los requisitos más relevantes para asegurar la navegación accesible de un sitio web es que pueda ser navegable por teclado, puesto que para muchas personas es el único o el principal medio de interacción con las páginas.

Las WCAG 2.0 incluyen una pauta, la 2.1 Accesible por teclado, con varios criterios de conformidad que hacen referencia a la operabilidad de las páginas mediante el teclado.

* 2.1.1 Teclado: toda la funcionalidad del contenido es operable a través de una interfaz de teclado sin que se requiera una determinada velocidad para cada pulsación individual de las teclas, excepto cuando la función interna requiere de una entrada que depende del trayecto de los movimientos del usuario y no sólo de los puntos inicial y final. (Nivel A)
* 2.1.3 Teclado (sin excepciones): Toda la funcionalidad del contenido se puede operar a través de una interfaz de teclado sin requerir una determinada velocidad en la pulsación de las teclas. (Nivel AAA)

Nos vamos a centrar en los cuatro problemas habituales que dificultan o impiden la navegación por las páginas mediante el teclado y que están relacionados con el foco:

* El foco no está visible
* El orden del foco no tiene sentido
* Las páginas tienen trampas para el foco
* Recibir el foco provoca automáticamente un cambio de contexto sin avisar previamente al usuario.

**El foco no está visible**

Cuando un control de formulario coge el foco también se aprecia visualmente. Por ejemplo un checkbox presenta su contorno punteado o un campo de texto muestra el cursor de texto parpadeante en su interior.

Estos recursos nos permiten reconocer visualmente dónde se encuentra el foco en todo momento. Sin embargo, muchos sitios anulan este efecto por razones meramente estéticas, dificultando mucho la navegación por teclado, en este caso entre los usuarios que pueden ver el foco, pues no saben que elemento lo ha cogido ni donde se encuentra en ese momento.

**El orden del foco no tiene sentido**

Como dice el criterio de conformidad 2.4.3 de nivel A, los componentes que pueden recibir el foco lo debe hacer en un orden que preserve su significado y operabilidad.

No solo es necesario asegurar que el orden de lectura, en una lectura lineal, por ejemplo con un lector de pantalla o sin CSS cargadas, tenga sentido, sino también que el orden del foco por los elementos de interacción lo tenga.

Para conseguirlo hay que organizar el contenido en el código HTML de forma que el orden por defecto sea el correcto.

Si por alguna circunstancia tenemos que cambiar el orden del foco por defecto (con “tabindex”), entonces hemos de hacerlo de forma que el orden del foco siga teniendo sentido.

**Trampas para el foco**

El tercer problema relacionado con el foco es que hay páginas en las cuales nos encontramos con trampas para el foco, zonas de la página a las que podemos llegar mediante el teclado, como una ventana modal, una película Flash o los controles de reproducción de un vídeo, pero de las cuales solo podemos salir, para regresar a otra parte de la página, mediante el ratón.

Por ejemplo, en un menú desplegable, si una vez que abrimos el menú y accedemos a su contenido con el tabulador, ya no podemos volver a salir con el teclado de esta capa, lo consideraríamos una trampa para el foco.

Pero si podemos cerrarlo (por ejemplo accediendo al botón “Cerrar”) con el teclado y el foco sigue normalmente por el resto de la página, entonces no lo consideraríamos una trampa para el foco.

**Recibir el foco provoca un cambio de contexto**

El cuarto problema es que haya componentes de la página que cuando reciben el foco (o bien cuando se cambia su estado) provoquen automáticamente un cambio de contexto sin advertir al menos previamente al usuario.

Los cambios de contexto solo deberían producirse cuando el usuario realiza una acción que habitualmente se utilice para solicitar un cambio de contexto, como hacer clic en un enlace o botón.

Un cambio de contexto es un cambio importante en el contenido de la página, que cuando se hace sin el consentimiento del usuario, puede desorientar a quienes no pueden ver la página o no pueden ver toda la página al mismo tiempo.

Esto incluye cambios en:

* El agente de usuario, como abrir un gestor de correo o un lector de PDF;
* El foco, por ejemplo mover automáticamente el foco a otro elemento.

Una mala práctica muy habitual en los formularios es mover el foco de un control a otro automáticamente, por ejemplo al rellenar los campos de una cuenta bancaria.

* El área de visualización del contenido: abrir otra ventana, otra pestaña o cambiar de marco.
* El contenido, enviar un formulario, ir a otra página, o que lo parezca porque se cambia el orden del contenido de la misma.

Todos estos casos son cambios de contexto, y como decimos, no pueden producirse automáticamente sin avisar al usuario. Los cambios de contexto solo deberían producirse cuando el usuario los solicite, por ejemplo al pulsar un enlace o botón.

### MÚLTIPLES VÍAS, UBICACIÓN Y CONSISTENCIA

#### Múltiples vías

Hay que ofrecer múltiples vías para localizar el contenido, permitiendo así a los usuarios navegar por el sitio de diferentes maneras que satisfagan sus diferentes necesidades.

2.4.5 Múltiples vías: se proporciona más de un camino para localizar una página web dentro de un conjunto de páginas web, excepto cuando la página es el resultado, o un paso intermedio, de un proceso. (Nivel AA)

Para algunas personas es más sencillo por ejemplo usar funciones de búsqueda que recorrer largos menús de navegación. Otras personas pueden preferir tablas de contenidos o un mapa del sitio para navegar o comprender la organización del mismo.

Por ello las WCAG 2.0 indican que debemos aplicar dos o más técnicas para ofrecer múltiples vías:

* Mapa del sitio.
* Función de búsqueda.
* Enlaces a páginas relacionadas.
* Tabla de contenidos.
* Otras técnicas para sitios pequeños.

**Mapa del sitio**

Un mapa de sitio facilita la comprensión del contenido y la organización del sitio web, y es una alternativa a la navegación mediante menús y barras de navegación.

Requisitos de un mapa de sitio

* Debe estar enlazado desde todas las páginas, o al menos desde cada una de las páginas que se listan en el mapa web. Suele ser habitual incluir el enlace en el pie o la cabecera de las páginas.
* Desde el mapa web se tiene que poder acceder a todas las páginas del sitio, directa o indirectamente.
* Sus elementos deben ser enlaces.
* La presentación y la estructura del mapa deben reflejar la organización del sitio, por ello se debe maquetar con los elementos HTML adecuados para reflejar dicha organización, como son los encabezados y las listas.
* Se debe revisar que el orden de lectura y de tabulación por los elementos del mapa es correcto.
* Debe actualizarse cada vez que se actualiza el sitio.

**Función de búsqueda**

Otra técnica es proporcionar una función de búsqueda en la página. De esta manera, los usuarios pueden buscar el contenido sin necesidad de comprender la estructura del sitio o navegar por el mismo. Es especialmente útil en sitios de gran tamaño.

La función de búsqueda se puede incluir a través de un único campo de búsqueda presente en todas las páginas del sitio, o bien se puede proporcionar una página propia para el buscador y sus opciones.

**Enlaces a páginas relacionadas**

Otra técnica es proporcionar enlaces a páginas relacionadas. De esta manera los usuarios pueden seguir estos enlaces para encontrar información adicional.

**Tabla de contenidos**

Otra técnica es ofrecer una tabla de contenidos, que no hay que confundir con un menú de navegación o un mapa del sitio.

Una tabla de contenidos es un índice con las secciones y subsecciones de una página (o conjunto de páginas) que proporciona un acceso directo a una sección específica.

**Otras técnicas**

Si el sitio es pequeño, se pueden aplicar otras técnicas:

* El menú de navegación incluye el acceso a todas las páginas del sitio;
* Todas las páginas están enlazadas desde la página de inicio.

#### Ubicación del usuario

Un requisito de accesibilidad que hará que la navegación por tu sitio sea más accesible es facilitar la ubicación del usuario, proporcionarle información sobre dónde se encuentra dentro de un conjunto de páginas y que, de este modo, pueda contestar en todo momento a la pregunta "¿Dónde estoy?“.

2.4.8 Ubicación: se proporciona información acerca de la ubicación del usuario dentro de un conjunto de páginas web. (Nivel AAA).

**Migas de pan**

Las migas de pan ayudan a:

* comprender cómo se estructura el contenido de un sitio
* navegar a páginas visitadas anteriormente
* identificar la ubicación actual de una página dentro de un conjunto de páginas o dentro del sitio

Los requisitos que deben cumplir las migas de pan son:

* Estar siempre situadas en el mismo lugar en todas las páginas.
* Estar maquetadas mediante listas, aunque visualmente se presenten en línea.
* Emplear separadores visuales entre las mismas.
* Deben ser enlaces, salvo la ubicación actual.
* Deben indicar el camino usado para llegar a la página actual o bien indicar su situación en la jerarquía u organización del sitio.

**Indicar la situación actual en la navegación**

Otra técnica para facilitar la ubicación del usuario es indicar la situación actual en menús, barras de navegación, pestañas o pasos en un proceso.

Hay que recordar que toda información que transmitimos mediante cambios en la presentación también se debe poder determinar mediante software para que no le pase desapercibida a ningún usuario.

Por ello, el menú o paso en el que nos encontramos no solo se debe indicar visualmente, sino también estructural y semánticamente o mediante texto si fuera necesario.

Se puede utilizar para este fin el elemento link

<head>

<title>Capítulo 2</title>

<link rel="index" href="index.html“ title=“Índice" />

<link rel="start" href=“intro.html“ title=“Introducción“ />

<link rel="next" href=“cap3.html“ title=“Capítulo 3“ />

<link rel="prev" href="cap1.html“ title=“Capítulo 1“ />

<link rel="glossary" href="glosario.html“ title="Glosario“ />

</head>

Este elemento, que se incluye en el head de la página, permite describir la relación que existe entre esa página y el resto de las páginas de una colección, indicando el tipo de vínculo existente entre las mismas mediante su atributo rel.

La ventaja de usar link para establecer las relaciones es que este elemento se puede interpretar mediante software.

#### Consistencia

Otro aspecto que hace más accesible la navegación es la consistencia, tanto en la presentación y maquetación de los mecanismos de navegación, como en la identificación de los componentes que tienen la misma funcionalidad dentro de un conjunto de páginas

**Navegación coherente**

Si los mecanismos de navegación que se repiten en diferentes páginas de un conjunto (como el buscador, el menú de navegación, las migas de pan, el enlace saltar al contenido, etc.) se presentan en el mismo orden en las diferentes páginas del conjunto, los usuarios pueden predecir dónde estarán en cada página y localizarlos con más facilidad.

**Identificación coherente**

Tan importante es la navegación coherente, como la identificación coherente, porque si en un sitio web se usan diferentes etiquetas para una misma función se dificulta considerablemente la comprensión y la interacción con el sitio, y en definitiva su uso.

Identificación coherente quiere decir, en los enlaces, que aquellos que tienen un mismo destino tienen el mismo texto de enlace.

Por ejemplo, si el destino del enlace es la página de inicio del sitio, el texto del enlace no puede ser unas veces “Ir a la home”, en otras “Ir a la página de inicio” y en otras “Ir a la página principal”. El texto del enlace además deberá ser consistente con el título de la página, el nombre con el que se identifica a la misma en las migas de pan o en el mapa del sitio.

En el caso de destinos similares, los textos de enlace deberán ser consistentes. Por ejemplo, en un portal de noticias, el texto del enlace para ampliar información sobre cada noticia, no puede ser en un caso “Más información”, en otro “Leer más” y en otro “Más info”.

La identificación coherente también se aplica al texto alternativo de los elementos no textuales. Por ejemplo, si se usa el mismo icono en diferentes páginas, se entiende que con la misma función, debe tener el mismo texto alternativo, si tenemos un icono para descargar un fichero, su texto alternativo no puede ser en unos casos “descargar”, en otros “guardar” y en otros “bajar”.

Y la identificación coherente se aplica también a las etiquetas de los campos de formulario: los mismos controles de formulario deben estar etiquetados igual. Por ejemplo, no podemos etiquetar el campo contraseña en unos casos como “password”, en otros como “contraseña” y otros como “clave”.

## DISEÑO ACCESIBLE

Las Pautas de Accesibilidad para el Contenido Web (WCAG) se centran principalmente en el contenido, la navegación y la interacción. Respecto el diseño accesible, lo más importante aparece en la Pauta 1.4 Distinguible: Facilitar a los usuarios ver y oír el contenido, incluyendo la separación entre el primer plano y el fondo en la que se proporcionan algunos consejos sobre el uso del color, del contraste y del tamaño del texto.

### CONTRASTE

Existen numerosas discapacidades asociadas con el sentido de la vista, una de ellas es la “sensibilidad reducida al contraste”.

Cuando se realiza un diseño, si no queremos discriminar a algunos usuarios, tenemos que seleccionar colores que tengan un alto contraste para lograr un diseño universal. Además una combinación de colores de alto contraste puede ayudar a todos los usuarios.

En algunas situaciones, como puede ser un día muy soleado en la playa o en la nieve, una gran cantidad de luminosidad produce que no se vean correctamente las pantallas de los ordenadores, de los teléfonos móviles o de las tablets. También hay problemas de contraste cuando se emplea un proyector.

En el criterio de conformidad 1.4.3 contraste mínimo (nivel AA), se define un contraste de 4,5 para el tamaño normal del texto y de 3 para el tamaño del texto grande.

El criterio de conformidad 1.4.6 contraste mejorado es más estricto (nivel AAA): se define un contraste de 7 para el tamaño normal del texto y de 4,5 para el tamaño del texto grande.

Afortunadamente, existen numerosas herramientas, tanto online como para descargar, que podemos emplear para calcular el contraste entre dos colores. Un ejemplo es la herramienta Wave.

### COLORES

Para las personas con ceguera al color o daltonismo y/o con sensibilidad reducida al contraste el empleo de ciertos colores puede producir barreras de accesibilidad.

El criterio “1.4.1 Uso del color”, dice que no se debe usar el color como el único medio visual para transmitir la información.

El criterio de conformidad 1.4.8 Presentación visual, dice que “en la presentación visual de bloques de texto, se proporciona algún mecanismo para lograr que los colores de fondo y primer plano puedan ser elegidos por el usuario”.

Los navegadores ofrecen algunas opciones de accesibilidad que permiten cambiar el color del texto, de fondo y de los enlaces. Pero esto no soluciona todas las situaciones problemáticas que pueden existir.

Lo más adecuado es que el desarrollador de un sitio web proporcione algún mecanismo para lograr que los colores de fondo y primer plano puedan ser elegidos por el usuario.

Eso es muy sencillo hoy en día con el empleo de CSS. Un ejemplo es el sitio web de la BBC, que ofrece la opción para cambiar la combinación de colores de la página.

### TIPOGRAFÍA Y DISLEXIA

La Federación Mundial de Neurología define la dislexia como “un trastorno que se manifiesta por la dificultad en aprender a leer (y escribir) a pesar de recibir una enseñanza correcta, mostrar una inteligencia adecuada y disfrutar de las oportunidades socioculturales normales”.

Veamos 6 consejos fáciles de entender y aplicar para favorecer a los usuarios con dislexia.

**Efecto río**

Evitar el efecto río que se produce cuando existen grandes huecos vacíos en el texto en líneas consecutivas. Esto se puede deber a:

* El texto justificado, así que es mejor alinear el texto a la izquierda.
* Y cuando se dejan dos espacios en blanco después de cada punto.

**Efecto borroso**

También hay que evitar el efecto borroso. Esto se puede deber a:

* El uso del color negro puro para el texto sobre un fondo blanco puro.
* Bloques de texto muy grandes.

**Efecto de lavado**

Evitar el efecto de lavado que produce que el texto se vea débil e indistinguible. Esto se puede deber a:

* El uso de tipos de letra serif. Por lo tanto, mejor usar sans serif.
* El uso de texto en cursiva.

Más consejos

* Utiliza una fuente clara y que no produzca confusión porque dos letras distintas tienen un glifo similar (como las letras b, d, p, q, que normalmente son el mismo glifo pero rotado). Curiosamente, en este libro se recomiendo el uso del tipo de letra Comic Sans, odiada por la mayoría de los expertos en tipografía.
* Utiliza texto real y no imágenes: esto permite a los usuarios con problemas utilizar hojas de estilo propias con sustitución de la tipografía empleada.
* El contenido debe ser claro de leer. Para ello, las frases deben ser cortas (no más de 10-15 palabras por frase) y los párrafos deben ser cortos.
* Los colores de alto contraste pueden dificultar la lectura: un texto negro sobre un fondo blanco puede aparecer como borroso para muchas personas con dislexia. Se aconseja que el color del texto sea gris sobre un blanco oscuro. Además, esto cansa menos la vista para todas las personas.
* El texto justificado es muy bonito, pero causa problemas. Mejor no usarlo nunca.
* Las imágenes pueden ayudar y distraer a la vez, no hay una solución general: a muchos disléxicos les ayudan a comprender el contenido, pero a otros les puede distraer.
* Las animaciones son, sin duda, una gran distracción, mejor no usarlas. Si se tienen que usar, no se debe utilizar la opción de "autoplay" y el usuario debe tener el control para detenerlas en cualquier momento. Por eso mismo, los anuncios (banners) son un problema, una gran distracción.
* Cuidado con los fondos, en especial, las imágenes de fondo. Son otra gran distracción y dificultan la lectura.

Algunos usuarios con una dislexia muy severa emplean lectores de pantalla, como los usuarios ciegos. Así que, una página web correctamente accesible para un usuario ciego también ayudará a los usuarios con dislexia severa que utilicen un lector de pantallas.

Algunos usuarios prefieren imprimir la página web y leerla impresa, así que hay que comprobar que la página web se imprime correctamente. Por ejemplo, se puede crear una hoja de estilo CSS específica para impresión.

Hay varios tipos de letra diseñadas para facilitar la lectura de textos por parte de personas con dislexia:

* De las comunes, las mejores: Helvetica, Courier, Arial, y Verdana.
* Open Dislexic: [OpenDyslexic (gumroad.com)](https://app.gumroad.com/d/9b580823f23fa095b7504905a9cce058)
* <https://www.sylexiad.com/fonts/purchase/sylexiad-sans>
* [Dyslexie Font - An award winning Dyslexia font](https://www.dyslexiefont.com/en/home/)

### OTROS REQUISITOS DE ACCESIBILIDAD RELACIONADOS CON EL ESTILO

Criterio de conformidad 1.4.4 Cambio de tamaño del texto. (Nivel AA)

A excepción de los subtítulos y las imágenes de texto, todo el texto puede ser ajustado sin ayudas técnicas hasta un 200 por ciento sin que se pierda el contenido o la funcionalidad.

No es suficiente con que el texto se pueda redimensionar, sino que también ha de hacerlo de forma que siga siendo legible. Por ello, la maquetación del sitio debe adaptarse adecuadamente a los diferentes tamaños del texto, de forma que no se produzcan desbordamientos o solapamientos del contenido que dificulten o impidan su percepción.

Aunque existen otras técnicas para cumplirlo, lo más recomendable es que definas el tamaño del texto y de los elementos que lo contienen mediante medidas relativas (% o em) y no mediante medidas absolutas (px o pt).

Criterio de conformidad 1.4.8 Presentación visual. (Nivel AAA)

En la presentación visual de bloques de texto, se proporciona algún mecanismo para lograr lo siguiente:

* Los colores de fondo y primer plano pueden ser elegidos por el usuario.
* El ancho no es mayor de 80 caracteres o signos.
* El texto no está justificado (alineado a los márgenes izquierdo y derecho a la vez).
* El espacio entre líneas (interlineado) es de, al menos, un espacio y medio dentro de los párrafos y el espacio entre párrafos es, al menos, 1.5 veces mayor que el espacio entre líneas.
* El texto se ajusta sin ayudas técnicas hasta un 200 por ciento de modo tal que no requiere un desplazamiento horizontal para leer una línea de texto en una ventana a pantalla completa.

Hay que respetar siempre la separación del contenido y la presentación, de manera que los estilos deberían estar siempre definidos en las CSS del sitio.

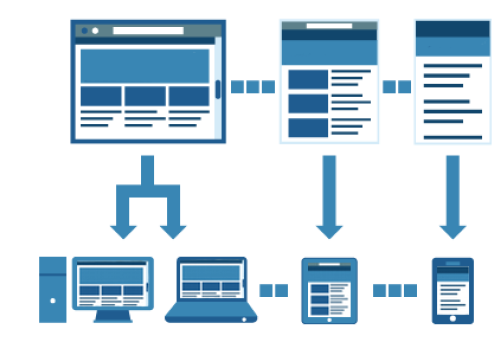
### DISEÑO WEB ADAPTABLE

El diseño web adaptable o adaptativo, conocido en inglés como responsive web design (RWD), tiene como objetivo realizar un diseño de una página web que sea capaz de adaptar la apariencia de la página web al dispositivo que se esté utilizando para visualizarla.

Gracias al diseño web adaptable una misma página web se puede visualizar en diferentes dispositivos, como ordenadores, portátiles, tabletas y teléfonos móviles, sin tener que crear diferentes versiones de la página: la página (el código HTML) es la misma, lo único que cambia es la apariencia que se define mediante diferentes presentaciones (diferentes CSS).

Esta flexibilidad se consigue mediante el uso de un código HTML único pero que se presenta de manera diferente gracias a:

* La separación entre el contenido y la presentación: todos los estilos están definidos en las CSS.
* Plantillas basados en grids: la información se organiza en ejes verticales y horizontales. Tendremos definidos diferentes layouts en diferentes CSS, por ejemplo uno de 3 columnas, uno de 2 columnas y uno de 1 columna.



* Grids fluidos, es decir, el uso de medidas relativas que permitan que el contenido se pueda adaptar realmente. Permite utilizar todo el espacio disponible y evitar el desplazamiento horizontal.
* Media Queries, permite cargar dinámicamente las diferentes CSS que hemos definido en función del tamaño de pantalla, su resolución o su orientación.
* Configuración del viewport: mediante este metadato indicamos que nuestra web es flexible para adaptarse a los diferentes anchos y resoluciones de pantalla, le indicamos al navegador que no aumente o reduzca la página, que use el zoom por defecto.
* Imágenes y vídeos de tamaño flexible, que también se adapten al espacio disponible. En el caso de los vídeos podemos usar HTML5 y el elemento video con diferentes sources según la resolución.

En cuanto a la metodología se viene defendiendo un enfoque Mobile First, basado en el principio de Mejora Progresiva. Pensar primero en la versión móvil y después ir añadiendo complejidad mediante Media Queries, que serán ignoradas por los navegadores que no las soporten.

Un sitio desarrollado con la técnica Responsive Design no implica que dicho sitio vaya a ser accesible y a cumplir con las pautas de accesibilidad, sin embargo es un gran punto de partida.

### INTERACCIÓN ACCESIBLE

La interacción del usuario con las páginas web es una parte muy importante de las páginas web actuales, ya que desde hace tiempo las páginas web no son simplemente contenedoras de información, sino que permiten a los usuarios realizar actividades de todo tipo como contestar un correo electrónico, redactar un documento o participar en un juego online.

La programación de la interacción de un usuario con una página web se realiza con JavaScript, el lenguaje estándar para la programación de las páginas web en el lado del cliente.

JavaScript es una tecnología compatible con la accesibilidad y, por tanto, compatible con los productos de apoyo. Podemos usar JavaScript en nuestras páginas, siempre y cuando se implemente de forma accesible.

Para ello debemos seguir una serie de pautas:

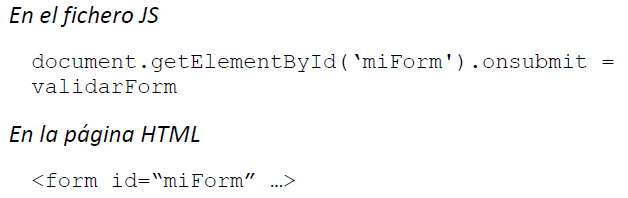
* Usar correctamente los manejadores de eventos JavaScript y el atributo “tabindex”.
* Utilizar funciones DOM para insertar, eliminar o modificar contenido dinámicamente.
* Implementar JavaScript de forma no intrusiva, separando por tanto el contenido, la presentación y el comportamiento de la página.
* Principio de “mejora progresiva”

### MEJORA PROGRESIVA

La “mejora progresiva” se basa en la separación del contenido, la presentación y el comportamiento, que implementamos con JavaScript no intrusivo. Y asimismo se basa en la mejora de la funcionalidad de nuestra página de forma progresiva.

Implementaremos las páginas como si no fueran a soportar JavaScript y sobre ellas añadiremos una capa de programación JavaScript no intrusivo como mejora.

La forma no intrusiva y correcta es separar el comportamiento incluyendo toda la programación en ficheros JS.



¿En qué circunstancias o contextos no se van a ejecutar los scripts de mis páginas?

Si el navegador no soporta los scripts o su soporte es antiguo, por ejemplo en navegadores antiguos, especialmente de dispositivos móviles o navegadores para televisión, o en navegadores solo texto.

Si se ha desactivado por el usuario o, por ejemplo, por el firewall corporativo por motivos de seguridad.

Si se ha implementado de forma propietaria y no con funciones DOM, como vimos, y por tanto el navegador no comprende el script.

Entre un 1% y un 2% por ciento de los usuarios navegan sin soporte para JavaScript. Puede parecer un porcentaje muy pequeño, pero ten en cuenta que uno de estos usuarios que no ejecutará tus scripts es muy importante para tu sitio, es el robot de los buscadores. En el caso de Googlebot solo comprenderá los que sean muy simples. Por tanto, es muy probable que los robots de los buscadores no sigan tus enlaces si dependen de JavaScript, ni indexen el contenido si depende de JavaScript.

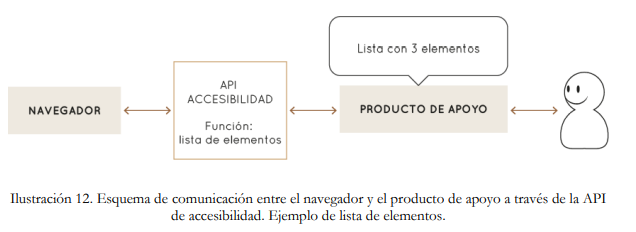
Por último comentar que existe un enfoque alternativo a la “mejora progresiva”: la “degradación elegante”.

La degradación elegante parte de implementar la solución teniendo en cuenta a los usuarios con los dispositivos y navegadores más modernos, y pensar después hacia atrás, en dar una solución alternativa a los usuarios con navegadores más antiguos, para ofrecerles una funcionalidad mínima y suficiente.

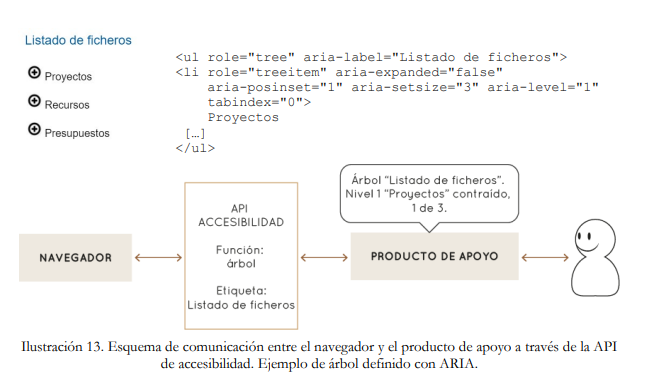
### WAI-ARIA

WAI-ARIA o ARIA (Accessible Rich Internet Applications 1.0) es una especificación del W3C, una taxonomía de roles, estados y propiedades que ayudan a definir los elementos de la interfaz para que lo definido en la página con html sea descriptivo a un lector de pantalla (y por lo tanto al usuario)

EJEMPLO sin ARIA:



EJEMPLO con ARIA: El lector de pantalla NVDA diría “Árbol Listado de ficheros Nivel 1 Proyectos contraído, 1 de 3”



ARIA nos va a permitir incluir información semántica sobre la estructura de la página, los componentes de la interfaz, su comportamiento y la relación entre los mismos, de manera que esta información pueda ser transmitida a los usuarios que utilizan productos de apoyo, como un lector de pantalla, y facilitarles así la comprensión de la página y su interacción con la misma.

Para incluir esta información semántica sobre la interfaz y su comportamiento, WAI-ARIA proporciona una ontología de roles, estados y propiedades.

Por tanto, se podría definir ARIA como un conjunto de atributos que se añaden a las etiquetas HTML para que los agentes de usuario (navegadores y productos de apoyo) comprendan que estas etiquetas tienen un comportamiento diferente al habitual y actúen en consecuencia.

Pasos:

1. Tabindex para poder recibir el foco: Tabindex en todos los elementos
2. Roles:

Incluir información sobre el rol o función de un elemento de nuestra página mediante WAI-ARIA es tan sencillo como añadir a la etiqueta del elemento role=”[nombre\_del\_rol]”.

Existen dos tipos de roles:

Listado de roles:

* <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/ARIA_Techniques>
* <https://developer.mozilla.org/en-US/docs/Web/Accessibility/ARIA/Roles>

1. Definen elementos de la interfaz (árboles, alertas, sliders, etc.)
2. Definen la estructura de la página (cabecera, navegación, pie)

Dentro de los que definen la estructura de la página se distinguen los landmark roles, que se usan para identificar áreas separadas de la página y transmitir la naturaleza de las mismas. De esta manera añadimos características útiles de navegación global que transmiten información de la estructura de la página e información semántica sobre estas zonas.

1. Estados y propiedades

A los elementos dinámicos que cambian de estado (caso de los checkbox, listas expandidas o no,…) además del rol se les puede asociar el estado (desplegado, seleccionado,…)