



---

# ACCESIBILIDAD WEB

---

Diseño de Interfaces



2 DE FEBRERO DE 2022

1º DAW  
César Toribio

## Índice

¿Por qué la accesibilidad es clave? .....	2
Recomendaciones básicas de accesibilidad. ....	2
Asociar cada formulario con un label.....	2
Incluir texto alternativo a las imágenes. ....	3
Identificar el idioma de la página. ....	3
Usar etiquetas semánticas. ....	3
Usar mensajes de error autoexplicativos.....	3
Escribir el código en el orden de visualización.....	4
Escribir código que adapte a la tecnología del usuario.....	4
Dar explicaciones sobre los elementos no habituales .....	5
Asegurar que los elementos interactivos sean accesibles con el tabulador.....	6
Evitar usar Captcha donde no sea necesario. ....	6

## ¿Por qué la accesibilidad es clave?

Podemos decir que una herramienta o sitio web está codificado de manera accesible una vez que todos los usuarios pueden disfrutar de lo que el sitio tiene que ofrecer sin importar que tipo de discapacidad tengan, ya que hay métodos para enfocar muchas de ellas y así hacer que nuestro sitio sea accesible para más usuarios.

Además, los motores de búsqueda escanean nuestro sitio y analizan cuán accesible es, para beneficiarlo a la hora de realizar el posicionamiento.

Por tanto, la accesibilidad es un punto clave a la hora de desarrollar la página web, ya que además de permitir a personas con distintas discapacidades disfrutar del sitio, nos aportan beneficio de cara a la empresa, ya que muchos más usuarios van a poder acceder a nuestro sitio web.

Además de personas con discapacidad, la accesibilidad permite a personas que están en una situación fuera de lo corriente, por ejemplo, intentando ver un video en el metro, o con un brazo roto.

## Recomendaciones básicas de accesibilidad.

Estas son unas cuantas recomendaciones de accesibilidad básicas, que nos van a permitir empezar a trabajar en un sitio web accesible. Estas son:

- Accesibilidad relacionada con video.
- Asociar cada formulario con un label
- Incluir texto alternativo
- Identificar el lenguaje de la página
- Usar etiquetas semánticas para identificar la estructura de la página
- Usar mensajes de error auto explicativos
- Codificar en orden de lectura
- Escribir código adaptado a las tecnologías de los usuarios
- Dar explicaciones sobre elementos no habituales
- Asegurar que todos los elementos interactivos sean accesible mediante tabuladores
- Evitar usar CAPTCHA donde no sea necesario.

### Asociar cada formulario con un label.

Usar label a la hora de crear un formulario es muy útil, ya que permitirá que los usuarios usando un lector de páginas sepan a qué elemento corresponde ese input, incluso, se puede añadir el label y ocultarlo, tan solo para que diga que elemento es el que tenemos al alcance.

#### Rendered

Username

#### Code Snippet

```
<label  
for="username">Username</label>  
<input id="username" type="text"  
name="username">
```

### Incluir texto alternativo a las imágenes.

Es muy importante añadir texto alternativo a las imágenes mediante la etiqueta<alt>, ya que permitirá a los lectores de pantalla para personas ciegas o a los usuarios de móvil que tengan desactivadas las imágenes para ahorrar datos describir los contenidos de la imagen al usuario, por lo que es importante ser lo más descriptivo posible a la hora de escribir este texto, y que transmita lo más fidedignamente posible lo que busca transmitir la imagen.

### Identificar el idioma de la página.

Importante también para los lectores y para la manera en la que indexan los navegadores la página identificar el lenguaje en el que esta la página, y reflejar mediante un atributo de idioma "lang" los idiomas de las partes que contengan otro idioma, no solo para los usuarios con discapacidad que necesiten un lector, si no para que Google indexe correctamente la página web a los usuarios del idioma que busca atraer la página web.

```
<!DOCTYPE html>
<html lang="en"> == $0
  <head>...</head>
  <body>
    <header>...</header>
    <nav>...</nav>
    <section>...</section>
    <!-- Code injected by live-server -->
    <script type="text/javascript">
```

### Usar etiquetas semánticas.

Es indispensable utilizar etiquetas semánticas HTML5 para identificar la estructura del contenido, ya que además de facilitar la legibilidad del contenido para los usuarios y desarrolladores y permitir ordenar y estructurar el contenido, permite que los buscadores lo indexen mejor, por lo que es indispensable utilizar las etiquetas de HTML5, como <main>,<section>,<nav> y demás etiquetas que facilitan la estructura de la página.

### Usar mensajes de error autoexplicativos.

Se entiende perfectamente, si estamos, por ejemplo, diseñando un formulario, es imprescindible que guiemos al usuario lo máximo posible, y, que, si se incurre en algún error, por ejemplo, una contraseña no segura, se explique apropiadamente, o si queremos guiar a los usuarios a introducir datos con un formato concreto, hemos de explicárselo, ya que el usuario no va a adivinarlo, esto además de hacer la página más accesible, la hará más sencilla de usar para todos los usuarios.

## Escribir el código en el orden de visualización

Es importante que la estructura del código sea similar al orden de lectura de los elementos, independientemente de que se pueda lograr la misma visualización con orden diferente de etiquetas. En el siguiente ejemplo, el título va a antes de la imagen en el segundo trozo de código, mientras que en el primer snippet la imagen va antes:



### Space trainers

Space trainer for a classic and stylish look.

[Add to cart](#)

#### ✖ Image before heading may be missed

```

<h3>Space trainers</h3>
<p>Space...</p>
<a href="...">Add to cart</a>
```

[+ View complete code example](#)

#### ✔ Heading marks the start of the section

```
<h3>Space trainers</h3>

<p>Space...</p>
<a href="...">Add to cart</a>
```

[+ View complete code example](#)

## Escribir código que adapte a la tecnología del usuario

También es importante usar diseño responsive para que el contenido se ajuste a distintos niveles de zoom y tamaños de pantalla, ya que, si una persona con problemas visuales necesita ampliar mucho la letra, tiene que poder disfrutar igualmente de la página. También es importante disponer de imágenes en distintos tamaños para que si se ve la página desde un teléfono y no necesitamos que se vea en la mejor definición, podemos disponer de distintos tamaños de imagen para optimizar la carga.

Aquí se pueden utilizar herramientas como flexbox o las media queries que reorganizan el contenido a medida que se reduce el ancho de pantalla o cambia el tipo de puntero.

## Dar explicaciones sobre los elementos no habituales

Es importante también dar explicaciones sobre elementos que no se ven en todas las páginas web, por ejemplo, si tenemos algún elemento con el que el usuario ha de interactuar y puede no saber que ha de introducir en ese campo, podríamos poner una pequeña explicación, por ejemplo, si es un campo de texto, podemos ponerle el atributo placeholder que explica lo que es, o un label para informar al usuario de lo que ha de introducirse. También podemos usar los elementos aria y los elementos role.

### Example: Menu function and state identified using WAI-ARIA

```
<nav aria-label="Main Navigation" role="navigation">
  <ul>
    <li><a href="#">Home</a></li>
    <li><a href="#">Shop</a></li>
    <li class="has-submenu">
      <a aria-expanded="false" aria-haspopup="true"
href="#">SpaceBears</a>
      <ul>
        <li><a href="#">SpaceBear 6</a></li>
        <li><a href="#">SpaceBear 6 Plus</a></li>
      </ul>
    </li>
    <li><a href="#">MarsCars</a></li>
    <li><a href="#">Contact</a></li>
  </ul>
</nav>
```

Asegurar que los elementos interactivos sean accesibles con el tabulador.

#### </> Code Snippet

```
var buttonExample =
document.getElementById('example-button');

buttonExample.addEventListener('keydown',
function(e) {
    // Toggle the menu when RETURN is
    pressed
    if(e.keyCode && e.keyCode == 13) {

toggleMenu(document.getElementById('example-
button-menu'));
    }
});

buttonExample.addEventListener('click',
function(e) {
    // Toggle the menu on mouse click

toggleMenu(document.getElementById('example-
button-menu'));
});
```

Evitar usar Captcha donde no sea necesario.

Los captchas son el famoso método para comprobar que el usuario de nuestra página web no sea un robot de scraping, por lo que se le plantea al usuario un determinado número de pruebas, por ejemplo, seleccionar cuales de estas fotos contienen un semáforo.

Sin embargo, aunque estas pruebas sean muy útiles, para la gente que no tiene las mismas capacidades, puede ser difícil, ya que, si no tienen vista, pueden no ser capaces de aceptar este captcha, por lo que, si no es un sitio muy necesario, o lo puedes sustituir con otro método de verificación, deberías no ponerlo, ya que puede dificultar la accesibilidad, además, incluso los usuarios que no tienen dificultades hacen que sea más difícil navegar por la página.