

Resumen: Accesibilidad web

1. Definición de Accesibilidad Web

Desarrollo de aplicaciones que puedan ser utilizadas por **todos los usuarios**, incluyendo aquellos con:

- **Discapacidades** (visuales, auditivas, motoras, cognitivas).
- **Limitaciones contextuales** (conexiones lentas, dispositivos antiguos, pantallas pequeñas).

Objetivo: Eliminar barreras para garantizar **inclusión digital**.

1.1. Barreras y Soluciones

1.1.1. Barreras del Entorno

Problema	Solución
Navegadores antiguos	Alternativas para funciones no soportadas (ej.: menús sin JavaScript).
Navegadores de texto	Equivalente textual para elementos no textuales (imágenes, vídeos o sonidos).
Conexiones lentas	Optimizar recursos (compresión de imágenes, lazy loading).
Pantallas no estándar	Diseño responsive.
Monitores monocromos	Evitar dependencia del color (usar íconos + texto).

1.1.2. Barreras por Perfil de Usuario

- **Ceguera:**
 - Texto alternativo (**alt**) en imágenes.
 - Contenido multimedia con transcripciones.
 - Navegación por teclado.
- **Baja visión:**
 - Tamaños de fuente relativos (ej.: **rem**).

- Alto contraste (mínimo 4.5:1).
 - **Daltonismo:**
 - No usar solo color para transmitir información (añadir símbolos o patrones).
 - **Auditivas:**
 - Subtítulos en videos.
 - Alternativas a comandos de voz.
 - **Motoras:**
 - Áreas interactivas grandes (mínimo 44×44px).
 - Soporte para navegación por teclado.
 - **Neurológicas/Cognitivas:**
 - Lenguaje claro y estructurado.
 - Evitar destellos (riesgo de epilepsia).
 - Permitir desactivar animaciones.
-

1.3. Buenas Prácticas Clave

- **HTML Semántico:**
 - Uso de `<header>`, `<nav>`, `<main>`, `<button>` (mejor que `<div>`).
 - **CSS Accesible:**
 - Unidades relativas (`rem`, `em`).
 - Contraste verificable con herramientas como [WebAIM Contrast Checker](#).
 - **Formularios Accesibles:**
 - Etiquetas asociadas (`<label for="id">`).
 - Mensajes de error claros.
 - **Multimedia:**
 - Subtítulos (`<track>`), transcripciones para audio.
-

1.4. Caso de Éxito

Ejemplo: Sitio de Gobierno

- **Problema inicial:** Inaccesible para usuarios con discapacidad visual.
 - **Mejoras:**
 - Texto alternativo en imágenes.
 - Navegación por teclado.
 - Contraste ajustado.
 - **Resultado:** ↑ 40% en satisfacción de usuarios con necesidades especiales.
-

1.5. Conclusión

La accesibilidad web **no es opcional**; es un derecho y una responsabilidad ética. Implementarla:

- ✓ **Beneficia a todos los usuarios** (ej.: navegación por teclado útil en móviles).
 - ✓ **Mejora el SEO** (los bots priorizan contenido accesible).
 - ✓ **Cumple regulaciones** (WCAG, ADA, Leyes locales).
- "El poder de la web está en su universalidad"** (Tim Berners-Lee).
-

2. Pautas de Accesibilidad Web (WCAG)

2.1. Evolución de las WCAG

- **WCAG 1.0 (1999):** 14 pautas iniciales enfocadas en:
 - Alternativas para contenido multimedia
 - Independencia de dispositivos
 - Claridad en navegación y estructura
- **WCAG 2.0/2.1 (2008/2018):** 4 principios fundamentales:
 1. **Perceptible:**
 - Texto alternativo para imágenes/videos
 - Contraste mínimo (4.5:1)

- Adaptable a diferentes formatos (braille, audio)
2. **Operable:**
- Navegación completa por teclado
 - Tiempo ajustable para tareas
 - Evitar contenido que cause convulsiones (parpadeos)
3. **Comprensible:**
- Lenguaje claro y predecible
 - Ayudas para corrección de errores
4. **Robusto:**
- Compatibilidad con tecnologías actuales y futuras
-

2.2. Técnicas Clave para Cumplir WCAG

2.2.1. Técnicas Fundamentales

- Alternativas textuales para multimedia
- Mapa del sitio y breadcrumbs
- Feedback claro en formularios
- Evitar dependencia del color

2.2.2. Técnicas por Tecnología

Tecnología	Ejemplos
HTML	Uso de etiquetas semánticas (<code><header></code> , <code><nav></code>), <code>alt</code> en imágenes, <code>lang</code> para idioma
CSS	Tamaños relativos (<code>rem</code>), pseudo-clases (<code>:focus</code>), separación estructura/estilo
JavaScript	Soporte para teclado, validación con alertas accesibles
WAI-ARIA	Atributos como <code>aria-label</code> , <code>aria-required</code> para lectores de pantalla

2.2.3. Para Servidores

- Redirecciones server-side (evitar JavaScript).
-

2.3. Niveles de Prioridad y Conformidad

Nivel	Cobertura	Requisitos
A	Prioridad 1	Mínimo exigible (ej.: texto alternativo)
AA	Prioridades 1+2	Estándar legal (ej.: contraste 4.5:1)
AAA	Prioridades 1+2+3	Máxima accesibilidad (ej.: contraste 7:1)

Logos de conformidad: Disponibles en [W3C](#).

2.4. Tabla de Verificación

Herramienta para evaluar accesibilidad:

- Ejemplo: [Checklist WCAG](#)
 - Incluye puntos como:
 - ¿Todas las imágenes tienen **alt** ?
 - ¿Es navegable solo con teclado?
 - ¿Hay subtítulos en videos?
-

Conclusión

Las WCAG son el estándar global para garantizar que la web sea **inclusiva**. Implementarlas:

✓ **Beneficia a usuarios con discapacidades.**

✓ **Mejora la experiencia para todos** (ej.: navegación por teclado útil en móviles).

✓ **Cumple con legislaciones** (ej.: ADA, EN 301549).

"La accesibilidad no es un extra, es la base de un buen diseño web."

3. Cómo Hacer Imágenes Accesibles

3.1. Texto Alternativo (alt text)

- Descripción concisa:

```

```

- **Imágenes decorativas:** Usar `alt=""` y `aria-hidden="true"` para ignorarlas.

3.2. Buenas Prácticas

✓ Optimización:

- Reducir peso de imágenes (formato WebP/AVIF).
- Evitar texto incrustado en imágenes.

✓ Contexto:

- Incluir texto descriptivo cerca de la imagen en el HTML.

```
<p>Visita nuestra sede  en Madrid.</p>
```

✓ Contraste:

- Mínimo 4.5:1 si la imagen contiene texto.

3.3. Errores Comunes

✗ Información crítica solo en imágenes:

- **Incorrecto:**

```

```

- **Correcto:**

```
<p>Contacto: <a href="tel:+123456789">+123456789</a></p>
```

✗ Uso de `longdesc`:

- Obsoleto. Mejor usar texto visible o enlaces a descripciones detalladas.

3.4. Atributo `title` (Opcional)

- Solo para información adicional no repetitiva:

```

```

Conclusión

- **Texto alternativo** es esencial para lectores de pantalla.
- **Imágenes decorativas** deben ser ignoradas por accesibilidad.
- **Evitar dependencia de imágenes** para información clave.

"Una imagen accesible es aquella que todos pueden entender, incluso sin verla."

4. Cómo Hacer Enlaces Accesibles

1. Texto Descriptivo

- **Evitar** frases genéricas como "Haz clic aquí".
- **Usar** texto que indique el destino del enlace:

```
<!-- Incorrecto →  
<a href="about.html">Haz clic aquí</a>  
<!-- Correcto →  
<a href="contact.html">Contáctanos</a>
```

4.2. Atributos y Estilos

- **title opcional:** Para información adicional (no repetitiva):

```
<a href="contact.html" title="Visita nuestra página de contacto">Contáctanos</a>
```

- **Destacar sin depender del color:** Subrayado, negrita o bordes:

```
<!-- Accesible →  
<a href="services.html" style="color: blue; text-decoration: underline;">Servicios</a>
```

4.3. Buenas Prácticas

- ✓ **Tamaño adecuado:** Mínimo 44×44px para dispositivos táctiles.
- ✓ **Orden lógico:** Estructura clara en menús:

```
<nav>
  <ul>
    <li>
      <a href="home.html">Inicio</a>
    </li>
    <li>
      <a href="about.html">Sobre nosotros</a>
    </li>
  </ul>
</nav>
```

- ✓ **Evitar enlaces vacíos:** No usar `href="#"`.

4.4. Pruebas de Accesibilidad

- Verificar con **lectores de pantalla** (ej.: NVDA, JAWS).
- Asegurar que los enlaces sean **navegables con teclado** (Tab/Shift+Tab).

Errores Comunes

✗ Enlace sin contexto:

```
<a href="data.pdf">Descargar</a> <!-- Mejor: "Descargar informe financiero 2023" →
```

✗ Solo color para diferenciar:

```
<a href="login.html" style="color: red;">Iniciar sesión</a> <!-- Falta subrayado →
```

Conclusión

Los enlaces accesibles deben ser:

1. **Descriptivos** (texto claro).

2. **Visualmente distinguibles** (sin depender solo del color).
3. **Fáciles de interactuar** (tamaño adecuado, navegación por teclado).

"Un enlace accesible es una puerta abierta para todos los usuarios."

5. Cómo Hacer Tablas Accesibles

5.1. Estructura Básica de Tablas Accesibles

- **Encabezados claros:** Usar `<th>` con `scope="col"` (columnas) o `scope="row"` (filas).

```
<table>
  <tr>
    <th scope="col">Producto</th>
    <th scope="col">Precio</th>
  </tr>
  <tr>
    <td>Laptop</td>
    <td>€999</td>
  </tr>
</table>
```

- **Descripción con `<caption>`:**

```
<table>
  <caption>Lista de precios 2023</caption>
  ...
</table>
```

5.2. Técnicas Avanzadas

- **Atributo `headers`:** Para tablas complejas, asociar celdas con encabezados:

```
<th id="producto">Producto</th><td headers="producto">Laptop</td>
>
```

- **Estructura semántica:** Agrupar contenido con `<thead>`, `<tbody>`, `<tfoot>`:

```
<table><thead>...</thead><tbody>...</tbody><tfoot>...</tfoot></table>
```

- **Evitar celdas vacías:** Usar `colspan` / `rowspan` para fusionar celdas:

```
<td colspan="2">Total: €1998</td>
```

5.3. Mejoras de Accesibilidad

- **Abreviaturas:** Usar `<abbr>` para clarificar términos:

```
<th><abbr title="Unidades disponibles">Stock</abbr></th>
```

- **Diseño responsive:**

- Añadir `overflow-x: auto` para desplazamiento horizontal en móviles.
- Usar CSS Grid/Flexbox para adaptar tablas a pantallas pequeñas.

5.4. Errores Comunes

✗ Tablas sin encabezados:

```
<table><tr><td>Dato 1</td><td>Dato 2</td></tr> <!-- Falta <th> →</table>
```

✗ Descripciones obsoletas: Evitar `summary` (HTML4) y usar `<caption>`.

Ejemplo Completo

```
<table>
  <caption>Inventario de productos</caption>
  <thead>
    <tr>
      <th scope="col">ID</th>
      <th scope="col">Nombre</th>
      <th scope="col">
```

```

        <abbr title="Unidades en stock">Stock</abbr>
      </th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">001</th>
      <td>Teclado</td>
      <td>50</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="3">Última actualización: 15/11/2023</td>
    </tr>
  </tfoot>
</table>

```

Conclusión

Las tablas accesibles deben ser:

1. **Estructuradas:** Con encabezados (`<th>`) y grupos semánticos (`<thead>` , `<tbody>`).
2. **Descriptivas:** Usar `<caption>` y `<abbr>` para contexto.
3. **Adaptables:** Responsive para móviles y compatibles con lectores de pantalla.

"Una tabla accesible es aquella que todos pueden entender, incluso sin ver su diseño."

6. Cómo Hacer Formularios Accesibles

6.1. Etiquetas Correctas

✓ Asociar `<label>` con campos:

```
<!-- Método 1: Atributo for →  
<label for="nombre">Nombre:</label><input type="text" id="nombre" required><!-- Método 2: Anidación →  
<label>  
  Apellido:  
  <input type="text" required></label>
```

✓ **aria-label** para elementos sin texto visible:

```
<button aria-label="Cerrar sesión">X</button>
```

6.2. Estructura y Agrupación

- Agrupar campos relacionados con `<fieldset>` y `<legend>`: CopyDownloadRun

```
<fieldset>  
  <legend>Información de Contacto</legend>  
  <label for="email">Email:</label>  
  <input type="email" id="email" required>  
</fieldset>
```

6.3. Navegación y Operabilidad

✓ **Teclado:**

- Asegurar que todos los campos sean navegables con **Tab**.
- Usar `tabindex` para orden personalizado (evitar valores > 0).

✓ **Botones descriptivos:**

```
<button type="submit">Registrarse</button> <!-- Mejor que "Enviar" →
```

6.4. Validación y Feedback

✓ **Campos obligatorios:**

```
<input type="text" required><span class="required">(Requerido)</span>
```

✓ **Mensajes de error claros:**

- Usar `aria-live="polite"` para actualizaciones dinámicas.

6.5. Diseño Accesible

- ✓ **Contraste mínimo** (4.5:1) entre texto/fondo.
- ✓ **Tamaño adecuado** de campos (mínimo 44×44px para táctiles).
- ✓ **Ejemplos de formato:**

```
<input type="tel" placeholder="Ej: +34 600 000 000">
```

6.6. Ejemplo Completo

```
<form>
  <fieldset>
    <legend>Datos Personales</legend>
    <label for="nombre">Nombre completo:</label>
    <input type="text" id="nombre" required aria-describedby="nombre-help"><span id="nombre-help">(Campo obligatorio)</span>
  </fieldset>

  <label for="email">Email:</label><input type="email" id="email" required
>

  <button type="submit" aria-label="Enviar formulario de contacto">Enviar
</button>
</form>
```

Errores Comunes

✗ Etiquetas no asociadas:

```
<label>Nombre:</label> <!-- Falta 'for' o anidación →
<input type="text">
```

✗ Botones ambiguos:

```
<button type="submit">OK</button> <!-- ¿OK para qué? →
```

Conclusión

Un formulario accesible debe ser:

1. **Comprensible:** Etiquetas claras y agrupación lógica.
2. **Navegable:** Operable solo con teclado.
3. **Tolerante a errores:** Validación con feedback claro.

"La accesibilidad en formularios no es un extra, es la base de una experiencia digital inclusiva."

7. Herramientas de evaluación, análisis y testeo de accesibilidad web

Listado de algunas de las herramientas de evaluación y testeo de accesibilidad web más destacadas:

- Taw: Aplicación web que permite hacer un análisis de accesibilidad de forma gratuita tan solo introduciendo la URL.
- Accessibilitychecker.org: Informe bastante completo sobre accesibilidad.
- Color Contrast Checker: aplicación para comprobar el contraste entre fondo y texto.
- Wave: Informe muy visual sobre la página a evaluar. Prestaciones parecidas a la extensión para Chrome creada por la misma empresa.
- Tenon.io: nos permite analizar un sitio web o un trozo de código.
- Textise: simulador online para ver una página web en modo «Sólo texto».
- Colour Contrast Analyser: aplicación para comprobar el contraste entre fondos y textos.
- WAI HTML Table Linearizar Entry Form: aplicación para evaluar tablas.
- W3C Markup Validator: Validador HTML.
- CSS Validator: Validador CSS.
- JAWS: lector de pantalla

Extensiones para instalar:

- Web Developer, extensión para Chrome.

- Helperbird Toolbar, extensión para Chrome y Firefox.
- WAVE Evaluation Tool, extensión para Chrome y Firefox.
- Screen Reader, extensión para Chrome.

Plugins de WordPress que ayudan a crear sitios web accesibles:

- **WP Accessibility:** Ayuda con una variedad de problemas comunes de accesibilidad que ocurren en los temas de WordPress.
- **Accessibility widget:** Aumentar el tamaño de los contenidos de la web.
- **Zoom widget:** Widget para hacer zoom.
- **Accessibility Access Keys:** Permite asignar a las teclas funciones de acceso directo a contenidos.
- **Dropdown Menus:** Permite navegar por los menús de una página con el teclado.
- **Accessible tag cloud:** Versión accesible del widget que crea una nube de etiquetas.
- **GSpeech plugin :** Plugin que convierte a audio cualquier texto seleccionado en la Web.