

DISEÑO DE INTERFACES WEB

ACCESIBILIDAD WEB

SEBASTIAN POPESCU

2ª DESARROLLO DE APLICACIONES WEB

ÍNDICE

Contenido

I. Introducción.....	2
II. Principios de Diseño Accesible y su Implementación	3
1. Principio de Perceptibilidad	3
2. Principio de Operabilidad	4
3. Principio de Comprensibilidad	5
4. Principio de Robustez.....	6

I. Introducción

La accesibilidad web se fundamenta en la premisa de que toda persona, independientemente de sus capacidades, debe poder acceder y beneficiarse de la información y servicios que ofrece la red. Este concepto se enmarca dentro del diseño universal, que promueve la creación de entornos digitales inclusivos y sin barreras. Desde una perspectiva teórica, la accesibilidad se sustenta en cuatro pilares esenciales:

1. Perceptibilidad: Garantiza que toda la información —visual, auditiva o de otro tipo— sea presentada de forma que pueda ser percibida por todos los usuarios. Esto implica el uso de textos alternativos para imágenes, subtítulos en contenidos multimedia y una adecuada selección de contrastes y tipografías.

2. Operabilidad: Asegura que los elementos interactivos y de navegación sean accesibles y utilizables con diversos dispositivos de entrada, como el teclado o tecnologías de asistencia. Se busca que la interacción con el contenido sea sencilla y sin obstáculos.

3. Comprensibilidad: Se centra en que el contenido y la interfaz sean claros, coherentes y predecibles. Utilizar un lenguaje sencillo, estructuras semánticas apropiadas y mensajes de error claros son estrategias clave para facilitar la comprensión de la información.

4. Robustez: Consiste en que el contenido web esté construido siguiendo estándares y prácticas recomendadas (como las WCAG del W3C), de manera que pueda ser interpretado de forma fiable por una amplia variedad de navegadores y tecnologías de asistencia, garantizando su funcionamiento a lo largo del tiempo.

Estas directrices, formalizadas en las Directrices de Accesibilidad para el Contenido Web (WCAG), no solo constituyen un marco normativo y técnico, sino que también reflejan una responsabilidad ética y social. Al integrar estos principios, el desarrollo web se orienta hacia la creación de experiencias digitales que respeten y valoren la diversidad de usuarios, promoviendo la igualdad de oportunidades en el acceso a la información.

II. Principios de Diseño Accesible y su Implementación

1. Principio de Perceptibilidad

Para garantizar que toda la información se presente de forma que pueda ser percibida por todos los usuarios, se han implementado las siguientes acciones:

Texto Alternativo en Imágenes y Multimedia:

Se ha agregado el atributo alt descriptivo a todas las imágenes, de manera que los usuarios de lectores de pantalla puedan comprender el contenido visual.

```
<h3>Elementos visuales</h3>


```

Subtítulos y Transcripciones para Contenidos Multimedia:

Los vídeos y otros recursos multimedia incluyen subtítulos y, en lo posible, transcripciones completas.

```
<!-- Video con subtítulos -->
<h3>Video demostrativo con subtítulos</h3>
<video controls width="800" aria-describedby="descripcion-video">
  <source src="video.mp4" type="video/mp4">
  <track kind="subtitles" src="subtitles_es.vtt" srclang="es" label="Español" default>
  Tu navegador no soporta el elemento de video.
</video>
<p id="descripcion-video" class="sr-only">Video demostrativo sobre accesibilidad web. Cuenta con subtítulos en español.</p>
</section>
```

Contraste de Color Adecuado:

Se han seleccionado combinaciones de colores que cumplen con los estándares WCAG. Se han utilizado herramientas (como el validador de contraste) para asegurar que la legibilidad no se vea comprometida.

```
/* Ejemplo de Contraste de Color Adecuado */
body {
  background-color: #ffffff; /* Fondo claro */
  color: #000000; /* Texto oscuro para un alto contraste */
}
```

Escalabilidad del Texto:

Se han definido tamaños de fuente y unidades relativas para que el usuario pueda aumentar o disminuir el zoom sin perder la estructura del contenido.

```
<!-- Escalabilidad del Texto -->
<h3>Ejemplo: Escalabilidad del Texto</h3>
<p style="font-size: 1.2rem;">El uso de unidades relativas (rem, em, %) permite que el usuario aumente o disminuya el zoom sin romper el diseño.</p>
/section>
```

2. Principio de Operabilidad

Se han adoptado medidas para que todos los elementos interactivos sean fácilmente operables:

Navegación por Teclado:

Toda la web ha sido diseñada para ser navegable usando únicamente el teclado. Se han probado los flujos de navegación (mediante la tecla Tab) para verificar que no existan “trampas” de foco.

Enlace de Salto (Skip Link):

Se ha implementado un “skip link” para que los usuarios puedan saltar directamente al contenido principal.

Foco Visible:

Se han definido estilos CSS para que el elemento enfocado se distinga claramente del resto.

Interacción Sencilla y Predecible:

Se ha simplificado la interacción en formularios y menús, asegurando que cualquier acción requiera pasos mínimos y sea clara para el usuario.

```
<!-- Interacción Sencilla y Predecible -->
<h3>Ejemplo: Formulario Sencillo</h3>
<form onsubmit="return validarFormulario()">
  <!-- Etiquetado Correcto en Formularios -->
  <label for="nombre">Nombre:</label>
  <input type="text" id="nombre" name="nombre">
```

Control de Tiempo:

Donde se requiera interacción en tiempo limitado, se han previsto alternativas para pausar o extender dicho tiempo.

```
<!-- Control de Tiempo -->
<h3>Ejemplo: Control de Tiempo</h3>
<p>Si existe un límite de tiempo para alguna interacción, se debe ofrecer la opción de pausar o extender dicho tiempo.</p>
<button onclick="alert('Temporizador pausado')" aria-label="Pausar temporizador">Pausar Temporizador</button>
</section>
```

3. Principio de Comprensibilidad

Para que la información sea fácil de entender, se han realizado las siguientes mejoras:

Lenguaje Claro y Simple:

Todo el contenido se redacta en un lenguaje sencillo, evitando tecnicismos innecesarios. Se incluyen definiciones y ayudas contextuales donde se requiera.

Etiquetado Correcto en Formularios:

Cada campo de entrada en los formularios está correctamente etiquetado con elementos `<label>`, de forma que se establece una relación clara entre la etiqueta y el control.

```
<!-- Etiquetado Correcto en Formularios -->
<label for="nombre">Nombre:</label>
<input type="text" id="nombre" name="nombre" required autocomplete="name">

<label for="email">Correo electrónico:</label>
<input type="email" id="email" name="email" required autocomplete="email">
```

Mensajes de Error e Instrucciones:

Se muestran mensajes claros y orientativos en caso de error o para guiar al usuario en el proceso de llenado de formularios y en la navegación de la web.

```
<!-- Mensajes de Error e Instrucciones -->
<span id="errorFormulario" class="error" aria-live="polite"></span>

<button type="submit">Enviar</button>
</form>
<script>
// Ejemplo simple de control de error e instrucciones
function validarFormulario() {
  let nombre = document.getElementById('nombre').value;
  let email = document.getElementById('email').value;
  let errorSpan = document.getElementById('errorFormulario');

  if (nombre.trim() === '' || email.trim() === '') {
    errorSpan.innerHTML = 'Por favor, complete todos los campos requeridos.';
    return false;
  }
  // Si todo está bien, se podría enviar el formulario
  errorSpan.innerHTML = '';
  return true;
}
</script>
```

Consistencia en el Diseño:

Se ha mantenido una estructura y un estilo coherentes en todas las páginas para que el usuario se familiarice rápidamente con la interfaz.

4. Principio de Robustez

La robustez asegura que el contenido se pueda interpretar correctamente por diferentes dispositivos y tecnologías:

Uso de HTML Semántico:

Se emplean etiquetas HTML5 semánticas (como <header>, <nav>, <main>, <article> y <footer>) para estructurar el contenido de forma lógica y facilitar su interpretación por tecnologías de asistencia.

Validación del Código:

Se ha validado el código HTML y CSS utilizando herramientas como el validador de W3C, lo que contribuye a la compatibilidad entre navegadores y dispositivos.

Implementación de ARIA:

Cuando resulta necesario (por ejemplo, en componentes dinámicos o personalizados), se han utilizado roles y atributos ARIA para mejorar la accesibilidad.

Compatibilidad con Tecnologías de Asistencia:

Se han realizado pruebas con lectores de pantalla (como NVDA o VoiceOver) para verificar que la información se transmita correctamente y que la estructura semántica sea respetada.