

- Tarea: Accesibilidad Web
 - Objetivos:
 - Problemas de accesibilidad y medidas a tomar:
 - a. Se usan atributos que no existen en la versión de HTML declarada.
 - b. No se proporciona un texto equivalente para los elementos no textuales.
 - c. Problemas con el menú lateral: texto ilegible en caso de no cargar imágenes.
 - d. El formulario está maquetado con tablas.
 - e. Existe una pequeña animación en Flash.
 - f. El video de YouTube no está subtulado.
 - g. No es posible utilizar el teclado para navegar.
 - Herramientas recomendadas:

Tarea: Accesibilidad Web

Objetivos:

- Reconocer la importancia de la comunicación visual y sus principios básicos.
 - Analizar y seleccionar los colores y tipografías adecuados para su visualización en pantalla.
 - Analizar alternativas para la presentación de la información en documentos web.
-

Problemas de accesibilidad y medidas a tomar:

a. Se usan atributos que no existen en la versión de HTML declarada.

Medidas:

1. Revisar y corregir el DOCTYPE del archivo HTML para asegurar que sea el adecuado para el contenido.

Ejemplo: Usar `<!DOCTYPE html>` para HTML5.

2. Actualizar los atributos para usar solo los permitidos por la versión de HTML declarada.
 3. Usar validadores de HTML como [W3C Validator](#) para identificar y corregir errores.
-

b. No se proporciona un texto equivalente para los elementos no textuales.

Medidas:

1. Añadir el atributo `alt` en todas las imágenes con una descripción adecuada.
 - Ejemplo: ``
 2. Para elementos como íconos o gráficos SVG, usar `aria-label` o `aria-hidden="true"` si no aportan información.
 3. Proporcionar transcripciones o descripciones textuales para otros elementos multimedia, como gráficos o diagramas.
-

c. Problemas con el menú lateral: texto ilegible en caso de no cargar imágenes.

Medidas:

1. Eliminar dependencia de la imagen de fondo para garantizar la legibilidad.
 - Cambiar el color del texto y del fondo para cumplir con las pautas de contraste mínimo de WCAG 2.1 (contraste mínimo 4.5:1).
 - Ejemplo: Texto blanco (#FFFFFF) sobre un fondo azul oscuro (#000080).
2. Usar herramientas de validación de contraste como [WebAIM Contrast Checker](#) para asegurarse de que los colores sean accesibles.
3. Proporcionar un `fallback` en CSS para el fondo si la imagen no se carga:

```
.menu {  
  background-color: #000080; /* Fondo alternativo */  
  color: #FFFFFF; /* Texto legible */  
}
```

d. El formulario está maquetado con tablas.

Medidas:

1. Reestructurar el formulario utilizando etiquetas semánticas de HTML5, como `<form>`, `<label>`, `<input>`, `<fieldset>` y `<legend>`.
2. Asegurar que cada campo tenga una etiqueta asociada:

```
<label for="nombre">Nombre:</label>
<input id="nombre" type="text">
```

3. Usar `aria-labelledby` o `aria-describedby` cuando sea necesario para mejorar la accesibilidad.

e. Existe una pequeña animación en Flash.

Medidas:

1. Eliminar la animación en Flash, ya que no es compatible con muchos navegadores y no es accesible.
2. Sustituirla por una animación HTML5 usando `<canvas>` o CSS3.
 - Ejemplo de animación CSS:

```
@keyframes bounce {
  0%, 100% { transform: translateY(0); }
  50% { transform: translateY(-10px); }
}
.animacion {
  animation: bounce 1s infinite;
}
```

f. El video de YouTube no está subtulado.

Medidas:

1. Añadir subtítulos al video directamente en YouTube o mediante un archivo de subtítulos en formato **.srt** o **.vtt**.
2. Utilizar el atributo **track** en HTML5 para incluir subtítulos:

```
<video controls>
  <source src="video.mp4" type="video/mp4">
  <track src="subtitulos.vtt" kind="subtitles" srclang="es"
label="Español">
</video>
```

3. Asegurar que los subtítulos sean precisos y accesibles.
-

g. No es posible utilizar el teclado para navegar.

Medidas:

1. Habilitar la navegación por teclado asegurándose de que todos los elementos interactivos (enlaces, botones, etc.) sean accesibles mediante **tab** y estén ordenados lógicamente.
 - Usar el atributo **tabindex** para ajustar el orden del foco si es necesario.
 - Evitar valores negativos en **tabindex** salvo que sea imprescindible.
2. Garantizar que todos los elementos interactivos tengan un estado visible al recibir el foco, utilizando estilos CSS como **:focus**.
 - Ejemplo:

```
button:focus {
  outline: 2px solid #0000FF;
}
```

3. Proveer alternativas accesibles para interacciones complejas, como menús desplegados o carouseles, para que puedan activarse con el teclado.
4. Validar la navegación utilizando herramientas de accesibilidad como:
 - **NVDA** (lector de pantalla).
 - **Lighthouse** en Google Chrome DevTools.
5. Considerar usar librerías como **FocusTrap** para garantizar una experiencia de navegación controlada en componentes interactivos.

Herramientas recomendadas:

1. **Validación de HTML:** [W3C Validator](#).
2. **Comprobación de contraste de color:** [WebAIM Contrast Checker](#).
3. **Accesibilidad general:** [Lighthouse](#).
4. **Herramientas de navegación por teclado:** [Deque Systems Axe](#).