

APELLIDOS, NOMBRE: FERNANDEZ DOMINGUEZ, PABLO	
Practica: 2ª Evaluación	
CURSO 2021/2022	FECHA DE ENTREGA:24/02/2022

DISEÑO DE INTERFACES WEB – DAW2

ACCESIBILIDAD

INDICE

Concepto de accesibilidad.....	2
Recomendaciones del Consorcio World Wide Web (W3C).....	2
WAI-ARIA.....	2
Landmark roles.....	2
Aria-label y aria-labelledby.....	3
Contenido web perceptible (recomendaciones de WebAim).....	3
Contenido no textual.....	3
Medios basados en el tiempo (videos/audio).....	3
Adaptable.....	3
Pautas de Accesibilidad al Contenido en la Web (WCAG).....	4
Métodos para realizar revisiones preliminares y evaluaciones de adecuación o conformidad de documentos Web.	4
Online.....	5
Plugins.....	5
Locales.....	5
Probadores de color.....	5
Mi pagina accesible:.....	5
Bibliografía.....	6

Concepto de accesibilidad

La accesibilidad es una característica que las interfaces web cumplen en mayor o menor grado. El objetivo inicial de su análisis es reducir las barreras de comunicación e interacción entre la interfaz y el usuario, considerando las limitaciones de personas discapacitadas y también necesidades de usuarios habituales. Se dice que una web es accesible cuando está diseñada con herramientas para facilitar su acceso a usuarios con limitaciones, ya sean por discapacidades o por circunstancias especiales. Por ejemplo, un usuario sin discapacidad auditiva también se puede beneficiar de una web que utiliza subtítulos para los vídeos, ya que es posible que no pueda reproducir su sonido en un determinado contexto.

Recomendaciones del Consorcio World Wide Web (W3C).

- Tener suficiente contraste entre los colores de cada apartado, diferenciando el primer plano y el fondo
- No utilice el color solo para transmitir información. Si bien el color puede ser útil para transmitir información, el color no debe ser la única forma de transmitir la información. Cuando utilice el color para diferenciar elementos, proporcione también una identificación adicional que no dependa de la percepción del color. Por ejemplo, use un asterisco además del color para indicar los campos de formulario obligatorios y use etiquetas para distinguir áreas en los gráficos.
- Remarcar los elementos interactivos para que estos sean mas visibles
- Proporcionar opciones de navegación claras y consistentes como las “migas de pan”.
- Incluir labels correctamente asociado dentro de los formularios
- Dar respuestas bien identificadas (error o respuestas satisfactorias).
- Utilizar cabeceras y espaciado para relacionar el contenido
- Crear diseños para diferentes tamaños de dispositivos
- Incluir imágenes y contenido multimedia con alternativas (subtítulos o alt de una imagen)
- Controles en un diseño de carrusel

WAI-ARIA

Es la forma de facilitar que los usuarios que utilicen lectores de pantalla no estén obligados a recorrerla de forma línea pudiendo ojear cada parte.

Landmark roles

La especificación [WAI-ARIA](#) define un tipo especial de aria roles, los landmark roles, que se usan para identificar áreas separadas de tu página y transmitir la naturaleza de la mismas. De esta manera añadimos características útiles de navegación global, consistentes en cualquier documento (X)HTML, que transmiten información de la estructura de la página e información semántica sobre estas zonas.

Las principales son las siguientes:

- Banner
- Complementary
- Contentinfo
- Form
- Main
- Navigation
- Search
- Region

Aria-label y aria-describedby

Es una manera de añadir información a las etiquetas de nuestra página.

Aria-label, en este atributo añades la información descriptiva del elemento.

Aria- describedby, en este atributo indicas el ID del elemento descriptivo, como puede ser un label en un formulario o un titulo en un contenedor.

Contenido web perceptible (recomendaciones de WebAim)

Se pone a disposición de los sentidos: vista, oído y/o tacto.

Contenido no textual

- Haciendo referencia a las imagines, estas deben tener un texto alternativo, utilizando la etiqueta **alt**.
- Las imágenes que no transmiten contenido, como una de fondo, reciben un texto alternativo vacío : **alt=""**, debiendo estar siempre la etiqueta alt aunque en este caso sin contenido en ella.
- Las entradas de un formulario deben ser accesibles, con un texto descriptivo, un label relacionado con la etiqueta for, un fieldset que agrupe los checkbox...
- Todo el contenido multimedia debe tener texto accesible.

Medios basados en el tiempo (videos/audio)

- Se debe proporcionar una transcripción para videos como audio.
- Integración de subtítulos programados (el uso de archivos vtt donde programas en que rango de tiempo se debe mostrar un texto en concreto, los cuales enlazamos con la etiqueta <track>)

Adaptable

El contenido debe poder mostrarse de diferentes maneras.

- El uso correcto de las etiquetas de html5, como un uso correcto de los encabezados, los tipos de listas, las etiquetas de texto especial (string), etiquetas semánticas como header, nav, main, article, section, footer
- Uso correcto de las etiquetas para mostrar información, una tabla en la que los datos sean tabulares, y cada celda corresponda con su encabezado.
- Un orden correcto de lectura y navegación

Pautas de Accesibilidad al Contenido en la Web (WCAG).

- **Asociar los claves con cada campo del formulario con la etiqueta for**

Utilice un atributo en el elemento vinculado al atributo del elemento form o mediante atributos WAI-ARIA. En situaciones específicas, puede ser aceptable ocultar los elementos visualmente, pero en la mayoría de los casos se necesitan etiquetas para ayudar a todos los lectores a comprender la entrada requerida.

Ejemplo: Uso y atributos for id

Prestados

Nombre de usuario

</> Fragmento de código

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

- **Incluir alternativas de texto en aquellas imágenes que aporten información**

Asegúrese de que el texto alternativo para las imágenes se agregue a todas las imágenes informativas y funcionales. Utilice texto alternativo vacío, para imágenes decorativas, o inclúyalos en el CSS en su lugar. Las alternativas de texto generalmente son proporcionadas por los responsables del contenido escrito.

- **Identificar los idiomas y la opción de alternar entre ellos**

Indique el idioma principal de cada página utilizando el atributo de la etiqueta, por ejemplo. Utilice el atributo en elementos específicos cuando el idioma del elemento difiera del resto de la página.

Remarcar texto para transmitir significado y una estructura Utilice el marcado adecuado para encabezados, listas, tablas, etc. HTML5 proporciona elementos adicionales para estructurar mejor su contenido. Los roles WAI-ARIA pueden proporcionar un significado adicional, por ejemplo, para identificar la funcionalidad de búsqueda. Trabaje con diseñadores y escritores de contenido para acordar significados y luego usarlos de manera consistente.

Ejemplo: Uso de HTML para proporcionar estructura y significado**Prestados****Superbear salva el día**

7 ago 2015

El oso favorito de la ciudad demuestra una vez más su temple al rescatar a un gato joven de un árbol. Los testigos dicen que los esfuerzos de Superbear no fueron apreciados por el felino, quien infligió algunas heridas menores por arañazos a su rescatador.

Artículos relacionados

- [Oso recibe llave de la ciudad](#)
- [Superbear significa alcalde](#)

</> Fragmento de código

```
<section>
  <article>
    <h2>Superbear saves the day</h2>
    <time datetime="2015-08-07">7 Aug
2015</time>
    <p>The city's favorite bear yet again
proves his mettle by rescuing a young cat from
a tree. Witnesses say that Superbear's efforts
were not appreciated by the feline, who
inflicted some minor scratch wounds on his
rescuer.</p>
    <aside>
      <h3>Related Articles</h3>
      <ul>
        <li><a href="#">Bear receives key to
city</a></li>
        <li><a href="#">Superbear stands for
mayor</a></li>
      </ul>
    </aside>
  </article>
</section>
```

- **Ayudar a los usuarios para corregir sus errores**

Proporcione instrucciones claras, mensajes de error y notificaciones para ayudar a los usuarios a completar formularios en su sitio.

Cuando se produce un error:

- Ayudar a los usuarios a encontrar dónde está el problema
- Proporcionar explicaciones específicas y comprensibles
- Sugerir correcciones

Sea lo más indulgente posible con el formato al procesar la entrada del usuario. Por ejemplo, acepte números de teléfono que incluyan espacios y elimine los espacios según sea necesario.

Ejemplo: campo de número de teléfono australiano con validación de perdón**Prestados**Teléfono

Por ejemplo, (02) 1234 1234

</> Fragmento de código

```
<label for="phone">Phone</label>
<input id="phone" name="phone" type="tel"
  pattern="^(\(?0[1-9]{1}\)?)?[0-9 -]*$"
  aria-describedby="phone-desc">
<p id="phone-desc">For example, (02) 1234
1234</p>
```

- **Reflejar el orden correcto en el código**

Asegúrese de que el orden de los elementos en el código coincida con el orden lógico de la información presentada. Una forma de verificar esto es eliminar el estilo CSS y revisar que el orden del contenido tiene sentido.

Ejemplo: Reflejar el orden de lectura lógico en el código



Entrenadores espaciales

Entrenador espacial para un aspecto clásico y elegante.

[Añadir al carrito](#)

❗ **Es posible que se pierda la imagen antes del encabezado**

```

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

[+ Ver ejemplo de código completo](#)

✔ **El encabezado marca el comienzo de la sección**

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

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

[+ Ver ejemplo de código completo](#)

- **Escribir código que se adapte a la tecnología del usuario**

Utilice el diseño responsivo para adaptar la pantalla a diferentes estados de zoom y tamaños de ventanilla, como en dispositivos móviles y tabletas.

Cuando el tamaño de la fuente se incrementa en al menos un 200%, evite el desplazamiento horizontal y evite cualquier recorte de contenido. Utilice la mejora progresiva para ayudar a garantizar que la funcionalidad y el contenido principales estén disponibles independientemente de la tecnología que se utilice.

Ejemplo: Uso de consultas multimedia para adaptar la navegación

```
/* On narrow viewports, make the
navigation full width */
@media screen and (min-width: 25em) {
  #nav {
    float: none;
    width: auto;
  }
  #main {
    margin-left: 0;
  }
}
```

```
/* On wider viewports, put the navigation
on the left */
@media screen and (min-width: 43em) {
  #nav {
    float: left;
    width: 24%;
  }
  #main {
    margin-left: 27%;
  }
}
```

- **Proporcionar significado para elementos interactivos no estándar (WAI-ARIA)**

Utilice WAI-ARIA para proporcionar información sobre la función y el estado de los widgets personalizados, como acordeones y botones personalizados. Se requiere código adicional para implementar el comportamiento de dichos widgets, como expandir y contraer contenido o cómo responde el widget a los eventos del teclado.

Ejemplo: Función de menú y estado identificados mediante 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>
```

- **Asegúrese de que todos los elementos interactivos sean accesibles desde el teclado**

Piense en el acceso al teclado, especialmente al desarrollar elementos interactivos, como menús, información del mouse, acordeones plegables o reproductores multimedia. Se utiliza para agregar un elemento que normalmente no recibe enfoque en el orden de navegación cuando se utiliza para la interacción. Utilice secuencias de comandos para capturar y responder a eventos de teclado.

- **Evite captcha siempre que sea posible**

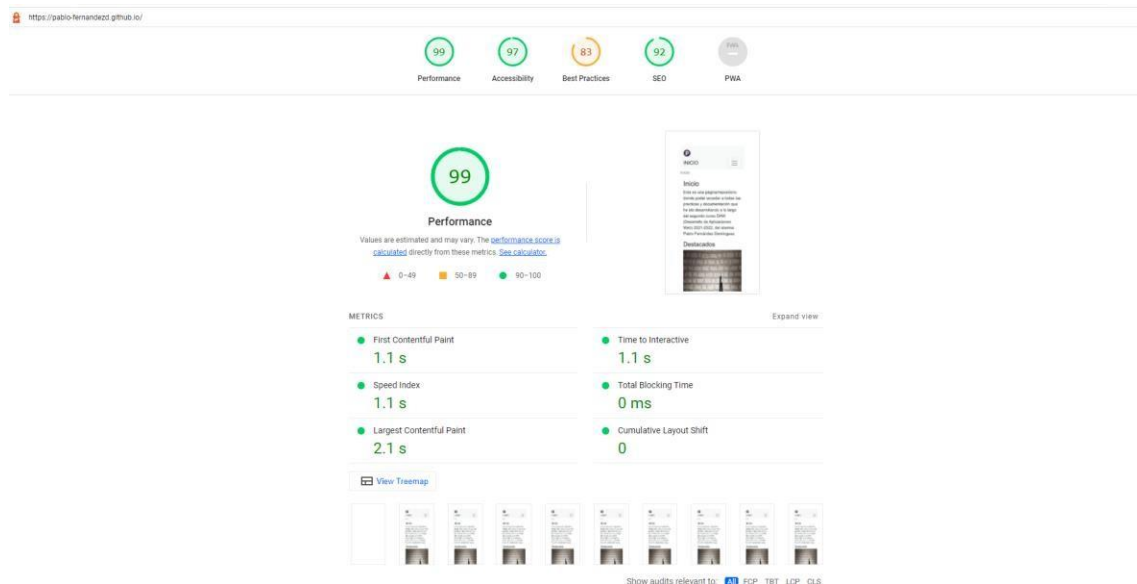
Los CAPTCHA crean problemas para muchas personas. Hay otros medios para verificar que la entrada del usuario fue generada por un humano que son más fáciles de usar, como la detección automática o las interacciones de la interfaz. Si CAPTCHA realmente necesita ser incluido, asegúrese de que sea fácil de entender e incluya alternativas para los usuarios con discapacidades, tales como:

- Proporcionar más de dos formas de resolver los CAPTCHAs
- Proporcionar acceso a un representante humano que pueda eludir CAPTCHA
- No requiere CAPTCHAs para usuarios autorizados.

Métodos para realizar revisiones preliminares y evaluaciones de adecuación o conformidad de documentos Web.

Existen numerosas herramientas para asegurar y poder corregir los diferentes fallos a la hora de hacer nuestra web accesible:

Uno muy reconocido tanto para rendimiento como accesibilidad es [lighthouse](https://github.com/GoogleChrome/lighthouse):



Existen otras herramientas según donde se ejecutan:

Online

[TAW](#)

[HERA](#)

[WAVE](#)

[SITEIMPROVE](#)

Plugins

Los navegadores disponen de un sinfín de herramientas o plugins que nos permiten añadir funcionalidad extra en nuestro cliente web.

[Lighthouse](#)

[Accesibility Developer tools](#)

Locales

Para poder analizar una web de forma más eficiente existen herramientas locales como:

AccessSniff: Se instala mediante el gestor de paquetes npm con npm i Access-sniff, este programa no he conseguido probarlo ya que ocurrían suficientes errores.

[HTMLCodeSniffer](#)

Probadores de color

[WEBAIM](#)

Mi pagina accesible:

General:

<https://pablo-fernandezd.github.io/DINT/Tema 4/>

Con imágenes y videos:

<https://pablo-fernandezd.github.io/DINT/Tema 4/accesiblevid>

Formulario:

<https://pablo-fernandezd.github.io/DINT/Tema 4/accesibform>

Bibliografía

[Lighthouse](#)

[W3schools](#)

[Webaim](#)