

# Unidad 3. Accesibilidad Web

Implementando Webs Accesibles

Técnicas de Accesibilidad

# Implementando Webs Accesibles

- HTML Semántico
- Atributo Alt
- Atributo Role (WAI-ARIA)
- Atributos Aria (WAI-ARIA)
- Atributo Tabindex
- Resumen de Técnicas
- Más consejos y técnicas

# Implementado Webs Accesibles

- HTML Semántico
- Es importante estructurar una página usando las etiquetas semánticas correctas, de modo que sea posible discernir cuál es el contenido principal de la misma o cuál el secundario.

# Implementado Webs Accesibles

- HTML Semántico

`<header></header>`

`<nav></nav>`

`<section>`

`<article></article>`

`<article></article>`

`</section>`

`<aside>`

`</aside>`

`<footer></footer>`

# Implementando Webs Accesibles

- HTML Semántico – strong y em
- También es importante utilizar la etiqueta **strong** para resaltar el texto y la etiqueta **em** para ponerlo en cursiva. Los lectores de pantalla darán al usuario la opción de navegar hasta cada uno de los encabezados disponibles.

```
<strong>Texto destacado con significado semántico</strong>  
<em>Texto en cursiva con significado semántico</em>
```

# Implementando Webs Accesibles

- HTML Semántico - Listas

Un lector de pantalla organizará los siguientes elementos y dará al usuario la opción de leer la lista no ordenada:

```
<ul>
  <li>Primer elemento no ordenado</li>
  <li>Segundo elemento no ordenado</li>
</ul>
```

El lector también informará al usuario de que una lista es secuencial:

```
<ol>
  <li>Primer elemento ordenado</li>
  <li>Segundo elemento ordenado</li>
</ol>
```

# Implementando Webs Accesibles

- HTML Semántico - Tablas

También es una buena práctica describir el contenido de las tablas mediante la etiqueta `caption`:

```
<table>
  <caption>Edades</caption>
  <tr>
    <td>Manuel</td>
    <td>24</td>
  </tr>
  <tr>
    <td>Edu</td>
    <td>35</td>
  </tr>
</table>
```

Del mismo modo, también deben usarse las etiquetas `thead`, `tbody` y `tfoot` para identificar la cabecera, el cuerpo y el pie de las tablas HTML.

# Implementando Webs Accesibles

- Atributo Alt

```

```

El atributo **alt** permite que las personas ciegas puedan recibir una descripción de la imagen y, además, también permitirá mostrar una descripción de la misma en caso de que no sea posible mostrar la imagen. Es casi imprescindible si quieres tener una buena puntuación SEO, ya que es uno de los muchos factores de peso que los buscadores utilizan.



# Implementando Webs Accesibles

- Atributo Role
- El atributo role permite **asignar diferentes roles a cada uno de los elementos de la página**. Los roles permiten identificar en qué consisten los diferentes elementos de la página. Sin embargo, no todos los elementos necesitan tener asignado un rol, sino solo aquellos más relevantes. Los lectores de pantalla son capaces de inferir el rol adecuado de cada elemento en la mayor parte de los casos en base al código HTML existente.

# Implementando Webs Accesibles

- ## Atributo Role

- **complementary**: El rol `complementary` sirve para identificar una sección de la página que se relaciona con el contenido principal de la misma, pero que también podría tener sentido de forma independiente. Un ejemplo de ello sería una barra lateral con enlaces o información relacionada con el artículo principal. Siempre que sea posible, debe usarse la etiqueta semántica `aside` en estos elementos.
- **list**: El rol `list` sirve para identificar una lista de elementos. Suele usarse junto con el rol `listitem`, usado para identificar los elementos de la lista.
- **listitem**: El rol `listitem` se utiliza para identificar los elementos que se encuentran en el interior de los elementos que tienen el rol `list`, usados a modo de contenedor.

# Implementando Webs Accesibles

- Atributo Role

- **main:** El rol `main` se usa para identificar el contenido principal de un documento. Se trata del contenido central de la página, sin el cual no tendría sentido su existencia.
- **navigation:** El rol `navigation` se usa para identificar aquellos grupos de enlaces que se utilizan para navegar por el sitio web o por los contenidos de la propia página.
- **region:** El rol `region` se utiliza para identificar áreas del documento que los autores han identificado como relevantes, proporcionando un mecanismo de navegación sencillo cuando ningún otro rol se adapta.

# Implementando Webs Accesibles

- ## Atributo Role

- **article**: El rol `article` sirve para indicar la sección de la página que podría conformar la página por sí misma dentro del documento. También se suelen incluir en su interior contenidos relacionados como información acerca del autor, comentarios o noticias.
- **heading**: El rol `heading` se usa para definir la cabecera de una página o de una sección, agregando más información estructural a la página.
- **banner**: El rol `banner` se usa para representar contenido genérico dentro de una página, que suele estar situado en la cabecera de la misma. En el interior del elemento con este rol podría estar el nombre del sitio web, el logotipo del mismo o ciertos elementos destacados.

# Implementando Webs Accesibles

- Atributo Role

- **form**: El rol `form` se utiliza en aquellos elementos que engloban a otros que proporcionan la misma funcionalidad que un formulario.
- **button**: El rol `button` se usa, tal y como su nombre indica, para representar aquellos elementos en los que el usuario puede hacer clic. En un lector de pantalla, estos elementos se representarán como si fuesen botones.
- **textbox**: El rol `textbox` se usa para identificar aquellos elementos en los que es posible introducir texto.
- **checkbox**: El rol `checkbox` se usa para indicar elementos interactivos que es posible seleccionar o marcar. Estos elementos deben contener también el atributo `aria-checked` para identificar así su estado.

# Implementando Webs Accesibles

- **Atributo Role**

- **table:** El rol `table` se utiliza para identificar los elementos que tienen estructura tabular no interactiva, con sus contenidos organizados en filas y columnas, sin que necesariamente tenga que tratarse de un elemento de tipo `table`.
- **cell:** El rol `cell` sirve para indicar aquellos elementos que forman parte de un conjunto de elementos que se encuentran en un mismo contenedor, sin que este contenga información acerca de las cabeceras de las filas o de las columnas del mismo.
- **row:** El rol `row` se usa para identificar un conjunto de celdas organizadas en filas con estructura tabular, que no siempre han de ser tablas HTML necesariamente. Un elemento con rol `row` puede contener más celdas, cuadrículas o grids de celdas o cabeceras de columnas, además de una cabecera de fila.
- **rowgroup:** El rol `rowgroup` se usa para indicar aquellos elementos con estructura tabular que contienen grupos de filas o elementos `row`.

# Implementando Webs Accesibles

- **Atributo Role**
  - **search:** El rol `search` se usa para identificar aquellos elementos de la página que permiten buscar contenidos en la misma, ya sean contenidos de la propia página o de otras páginas.
  - **contentinfo:** El rol `contentinfo` se usa para indicar aquella información que se suele repetir al final de cada página, como el copyright de la misma o ciertos menús con enlaces. Es habitual aplicar este rol a la etiqueta `footer`.
  - **document:** El rol `document` se suele utilizar para indicar a los lectores de pantalla que deben leer el contenido que hay en su interior. Suele usarse en widgets relativamente complejos.
  - **feed:** El rol `feed` se usa para indicar una lista de artículos. Los artículos del feed se agregan o eliminan de la lista cuando el usuario hace scroll por los mismos. En muchos casos estos feeds pueden ser infinitos.

# Implementando Webs Accesibles

- **Atributo Role**
  - **img**: El rol `img` se usa para identificar todos aquellos elementos que puedan ser considerados como imágenes. Estos elementos pueden contener imágenes, emojis, texto o cualquier otro elemento visual similar a una imagen.
  - **figure**: El rol `figure` se usa para indicar aquellos elementos visuales o esquemáticos que no es posible englobar en otros roles, como bloques de código, conjuntos de imágenes o esquemas; todo aquello que no sea información textual en general.
  - **grid**: El rol `grid` se usa con aquellos elementos que contienen una o más filas con celdas. Deberá ser posible navegar por cada una de las celdas con el teclado.
  - **gridcell**: El rol `gridcell` se usa con los elementos que estén en el interior de un elemento con rol `grid`. Se usan para emular el funcionamiento de las etiquetas `td` de las tablas HTML.



# Implementando Webs Accesibles

- Atributo Role
- No es necesario que agregar roles a las etiquetas HTML que ya identifican semánticamente su contenido, como por ejemplo las etiquetas **nav**, **form**, **button**, **table**, **ul** o **li**.

# Implementando Webs Accesibles

- Atributo Role
- No es necesario que agregar roles a las etiquetas HTML que ya identifican semánticamente su contenido, como por ejemplo las etiquetas **nav**, **form**, **button**, **table**, **ul** o **li**.

# Implementando Webs Accesibles

- Atributo Role

```
<nav>
  <ul>
    <li><a href="/">Inicio</a></li>
    <li><a href="/noticias">Noticias</a></li>
    <li><a href="/contacta">Contacta</a></li>
  </ul>
</nav>
```

Si por algún motivo te vieses obligado a usar una etiqueta `div` en lugar de una etiqueta `nav`, tendrías que usar el rol `navigation` en dicha etiqueta:

```
<div role="navigation">
  <ul>
    <li><a href="/">Inicio</a></li>
    <li><a href="/noticias">Noticias</a></li>
    <li><a href="/contacta">Contacta</a></li>
  </ul>
</div>
```

En resumen; el atributo `role` se usa para dar significado a aquellos elementos de la interfaz que carecen del mismo.

- <https://developer.mozilla.org/es/docs/Web/Accessibility/ARIA/Roles>

# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)
- Las técnicas **WAI-ARIA** (Web Accessibility Initiative – Accessible Rich Internet Applications [***Aplicaciones Enriquecidas Accesibles de Internet***]) proporcionan semántica, de tal forma que se pueden transmitir comportamientos de la interfaz de usuario e información estructural a las tecnologías de apoyo (por ejemplo, lectores de pantalla). La especificación de ARIA establece componentes que definen roles, estados y propiedades de los elementos de la interfaz de usuario.

# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)

- **aria-label:** Este atributo se utiliza para describir textualmente al elemento en el que se incluye:

```
<p aria-label="Descripción del texto aquí">...</p>
```

Este atributo es habitual en campos de texto que carecen de etiquetas, como por ejemplo un campo de búsqueda. En ocasiones es posible que un usuario pueda inferir la utilidad de algún elemento, pero puede que un lector de pantalla lo tenga más complicado, por lo que el atributo **aria-label** resulta útil en muchos casos.

# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)

- **aria-labelledby**: Este atributo establece una conexión entre un elemento y el elemento que lo etiqueta. Por ejemplo, podemos tener un campo de texto `input` y una etiqueta `label`, en cuyo caso los lectores de texto serán capaces de establecer una conexión gracias al atributo `for` de la etiqueta. Sin embargo, cuando queremos establecer conexiones usando otras etiquetas HTML, debemos usar el atributo `aria-labelledby` para establecer la conexión:

```
<h2 id="titulo">Título de la película</h2>

<div aria-labelledby="titulo">
Sinópsis de la película
</div>
```

# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)
  - **aria-hidden**: En ocasiones podrías querer que los lectores de pantalla ignoren algunos elementos que no contienen información relevante. Para ello puedes usar el atributo `aria-hidden`, que acepta los valores `true` y `false`. Si se usa el atributo con el valor `true` en algún elemento, los lectores de pantalla lo ignorarán:

```
<div aria-hidden="true">Información poco importante</div>
```

# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)

- **aria-describedby**: Este atributo establece una conexión entre un elemento y otro que se corresponde con su descripción:

```
<button aria-describedby="descripcion-boton" >Enviar formulario</button>  
  
<div id="descripcion-boton">Haz clic en el botón para enviar el formulario</div>
```

- **aria-describedbyat**: Este atributo permite especificar una URL en la que debería describirse el contenido del elemento sobre el que se aplica el atributo:

```

```



# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)

- **aria-level:** Este atributo permite establecer un orden jerárquico para los diferentes elementos de una estructura, que habitualmente será de árbol, aunque también se suele aplicar en listas de pestañas u otros elementos secuenciales. Partiendo de 1, el nivel debería aumentar a medida que aumenta la profundidad de los elementos:

```
<div id="nodo-1" aria-level="1">Nodo 1</div>
  <div id="nodo-1-1" aria-level="2">Nodo 1,1</div>
    <div id="nodo-1-1-1" aria-level="3">Nodo 1,1,1</div>
    <div id="nodo-1-1-2" aria-level="3">Nodo 1,1,2</div>
  <div id="nodo-1-2" aria-level="2">Nodo 1,2</div>
<div id="nodo-2" aria-level="1">Nodo 2</div>
  <div id="nodo-2-1" aria-level="2">Nodo 2,1</div>
  <div id="nodo-2-2" aria-level="2">Nodo 2,2</div>
```

# Implementando Webs Accesibles

- Atributos ARIA

- **aria-busy**: Sirve para indicar si un elemento y sus descendientes están cargando datos en el momento actual. Acepta los valores `true` o `false`, siendo `false` su valor por defecto:

```
<div id="usuarios" aria-busy="true">
...
</div>
```

- **aria-flowto**: Permite saltarse el orden lógico o visual de los elementos e indicar el siguiente elemento al que deben saltar los lectores de pantalla. Acepta como valor el id del siguiente elemento o una lista de ids, dando al usuario la opción de escoger el siguiente elemento de entre los de la lista en este último caso:

```
<ul>
  <li id="a" aria-flowto="c">Elemento 1</li>
  <li id="b">Elemento 2</li>
  <li id="c">Elemento 3</li>
</ul>
```

# Implementando Webs Accesibles

- Atributos ARIA (WAI-ARIA)
  - **aria-expanded**: Este atributo se utiliza para indicar si un elemento o conjunto de elementos están expandidos o colapsados. Suele usarse, por ejemplo, en los submenús de los menús desplegables. Acepta los valores `true`, `false` o `undefined`, siendo este último su valor por defecto:

```
<ul role="navigation">
  <li><a href="/inicio">Inicio</a></li>
  <li>
    <a href="/informacion">Información</a>
    <ul class="nav-submenu" aria-expanded="true">
      <li><a href="/programacion">Programación</a></li>
      <li><a href="/aplicaciones">Diseño</a></li>
    </ul>
  </li>
</ul>
```

- [https://www.neoguias.com/accesibilidad-web/#Usa\\_los\\_atributos\\_aria](https://www.neoguias.com/accesibilidad-web/#Usa_los_atributos_aria)

# Implementando Webs Accesibles

- Atributo Tabindex
- El atributo **tabindex** se usa *para cambiar el orden mediante el cual los elementos mostrados en pantalla se seleccionan cuando se pulsa el tabulador*. Por defecto, solamente los enlaces y los formularios pueden ser seleccionados de esta forma, pudiendo saltar rápidamente de un elemento a otro. Sin embargo, podrías querer que los usuarios puedan seleccionar otros elementos.
- El atributo tabindex acepta un número como valor que se corresponde con el orden de selección. Si agregas `tabindex="0"` a un elemento, será posible seleccionarlo siguiendo el flujo de selección habitual:

# Implementando Webs Accesibles

- Atributo Tabindex

```
<div tabindex="0">  
  ...  
</div>
```

Si agregas `tabindex="-1"` a un elemento, ya no será posible seleccionarlo o saltar a él cuando se pulse el tabulador:

```
<div tabindex="-1">  
  ...  
</div>
```

# Implementando Webs Accesibles

- Resumen de Técnicas: **T. Fundamentales:**
  - Incluir alternativas de texto al contenido multimedia.
  - Permitir pausar el contenido multimedia.
  - Incluir alternativas legibles cuando la lectura dependa del contraste de colores.
  - Ordenación coherente del contenido.
  - Incluir glosarios, mapa del sitio, tabla de contenidos.
  - Incluir títulos descriptivos.
  - Añadir enlaces para ir al principio de la página.
  - Identificar la localización del usuario dentro de la web (breadcrumbs).
  - Evitar parpadeos.
  - Alinear los textos de manera similar.
  - Ofrecer feedback de confirmación o negación al realizar una operación, etc.

# Implementando Webs Accesibles

- Resumen de Técnicas: **T. HTML:**
  - Utilizar elemento **title** para dar un título coherente a la página.
  - Incluir la etiqueta **meta description**.
  - Incluir el **botón de submit en los formularios**.
  - Definir el **idioma** correspondiente con el atributo **lang**.
  - Incluir el atributo **alt** en las imágenes.
  - Utilizar el atributo **caption** y **summary** en las **tablas**.
  - **Maquetación sin tablas**.
  - Utilizar los **encabezados h1-h6**.
  - Crear un **orden de tabulación** coherente en **formularios** y **enlaces**.
  - Utilizar **etiquetas estructurales o elementos semánticos**.

# Implementando Webs Accesibles

- Resumen de Técnicas: **T. CSS:**
  - Facilitar mecanismos para que se pueda modificar la hoja de estilos CSS: colores, fuentes, etc.
  - Separar la estructura de los estilos y no utilizar estilos en línea.
  - Utilizar tamaño de letra con medidas relativas: **unidades em.**
  - Incluir el foco en los elementos mediante la clase **:focus.**
  - Utilizar CSS para presentar el texto, controlar el espaciado, etc.



# Implementando Webs Accesibles

- Resumen de Técnicas: `JavaScript`:
  - Aumentar los tiempos de acción.
  - Soporte para utilizar tanto ratón como teclado.
  - Validación de datos con alertas.
  - Usar el DOM para manipular la página.
  - Utilizar scripts para modificar el aspecto de la página, como el fondo.

# Implementando Webs Accesibles

- HTML Semántico – h1 a h6
- Los elemento de cabecera implementan seis niveles de cabeceras de documentos; **<h1>** es la de mayor importancia y **<h6>** es la de menor impotancia. Un elemento de cabecera describe brevemente el tema de la sección que introduce.

# Implementando Webs Accesibles

- Más consejos y técnicas
  - <http://accesibilidadweb.dlsi.ua.es/?menu=guiabreve-1>
  - <https://www.w3.org/WAI/GL/WCAG20-TECHS/>
  - [http://www.codexexemplar.org/curso/curso\\_2\\_6.php](http://www.codexexemplar.org/curso/curso_2_6.php)
  - <http://sidar.org/traducciones/wcag20/es/#guidelines>
- Logos de conformidad
  - <https://www.w3.org/WAI/WCAG1-Conformance.html.es>