

```
<!--FPS 2ºDAW-Diseño de Interfaces Web -->
```

# Accesibilidad{

```
<Por="Ylenia Acuña y Llara Ramos"/>
```

}



## Contenidos II

01

Recordamos...

etiquetas ARIA

02

Etiqueta ROLE

03

Niveles WCAG 2.1

04

Lectores de pantalla

05

Accesibilidad

Angular /Bootstrap

06

Práctica FORMULARIOS

## aria-label {

Proporciona una etiqueta textual a un elemento para los usuarios de tecnologías de asistencia, especialmente útil cuando no hay una etiqueta visual.

```
<button aria-label="Buscar">  
  <i class="fas fa-search"></i>  
</button>
```

## aria-labelledby }

Asociar un elemento con el texto de otro elemento que funciona como su etiqueta. Esto es útil cuando el texto de la etiqueta no está físicamente junto al elemento que etiqueta.

```
<div id="user-name">  
  <input type="text" id="name" aria-labelledby="nameLabel"> </input>  
  <span id="nameLabel">Nombre de Usuario:</span>  
</div>
```



# aria-hidden {

Se utiliza para **ocultar elementos** que no deben ser percibidos por los usuarios de lectores de pantalla.

```
<div>
```

```
  <h2> Biblioteca Virtual <span aria-  
hidden="true">📖</span> </h2>
```

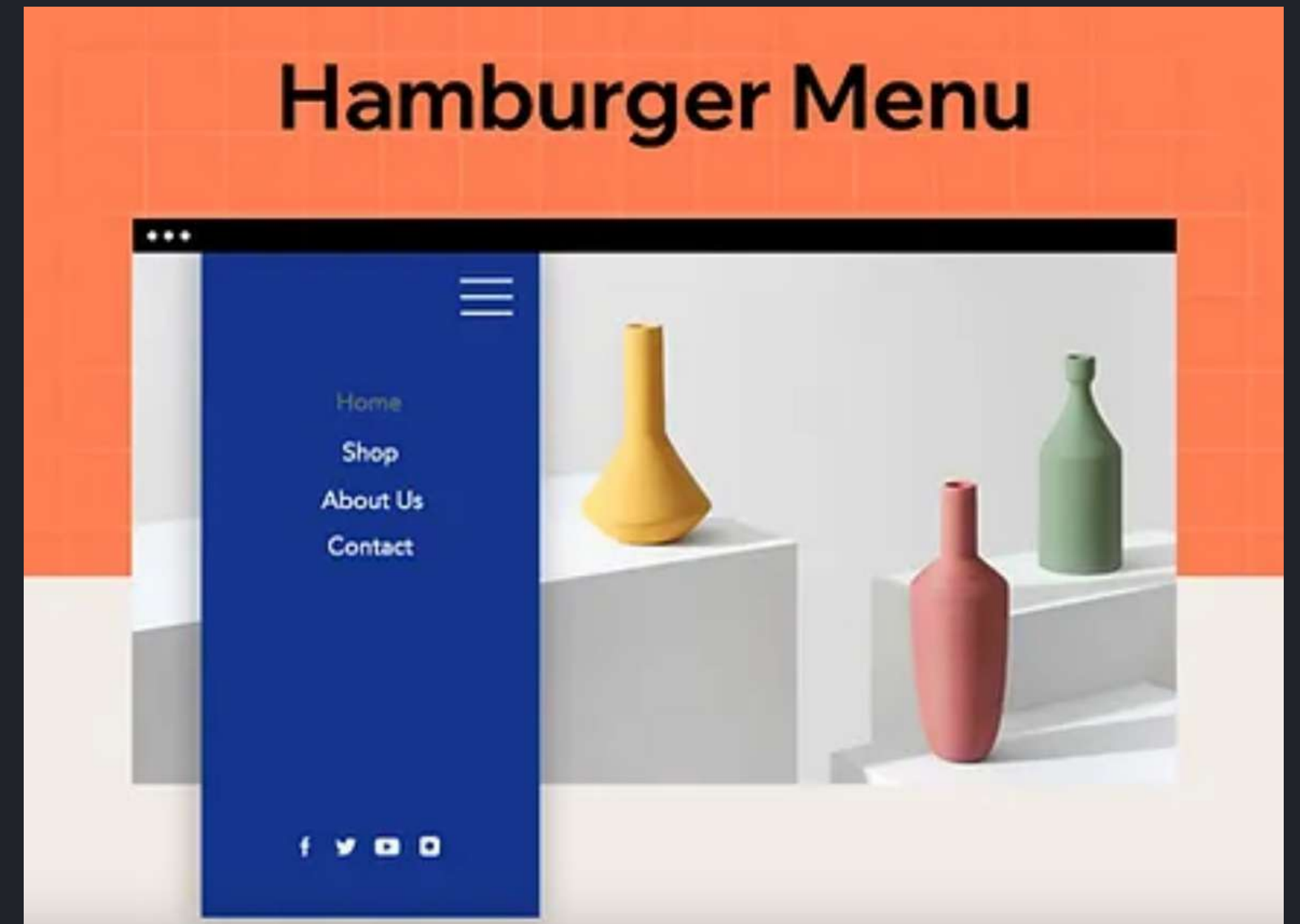
```
  <p> Explora nuestra colección de libros  
electrónicos y recursos en línea.</p>
```

```
</div>
```



# aria-expanded {

Se utiliza para indicar si los **contenidos expandibles están actualmente expandidos o colapsados**, lo cual es crucial para la accesibilidad de usuarios que dependen de lectores de pantalla.





# aria-expanded {

```
<button id="menuButton" aria-expanded="false" onclick="toggleMenu()">Menú </button>
```

```
<div id="menuContent" hidden>
```

```
<ul>
```

```
<li><a href="#">Inicio</a></li>
```

```
<li><a href="#">Sobre nosotros</a></li>
```

```
<li><a href="#">Servicios</a></li>
```

```
</ul>
```

```
</div>
```

```
<script>
```

```
function toggleMenu() {
```

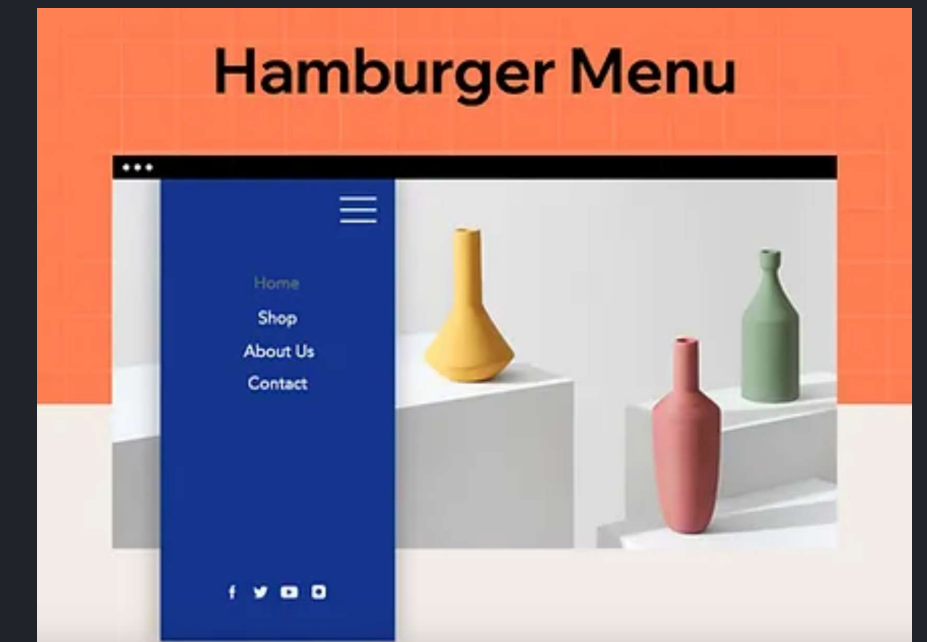
```
var menu = document.getElementById("menuContent");
```

```
var isExpanded = menu.getAttribute("aria-expanded") === "true";
```

```
document.getElementById("menuButton").setAttribute("aria-expanded", !isExpanded); menu.hidden = isExpanded;
```

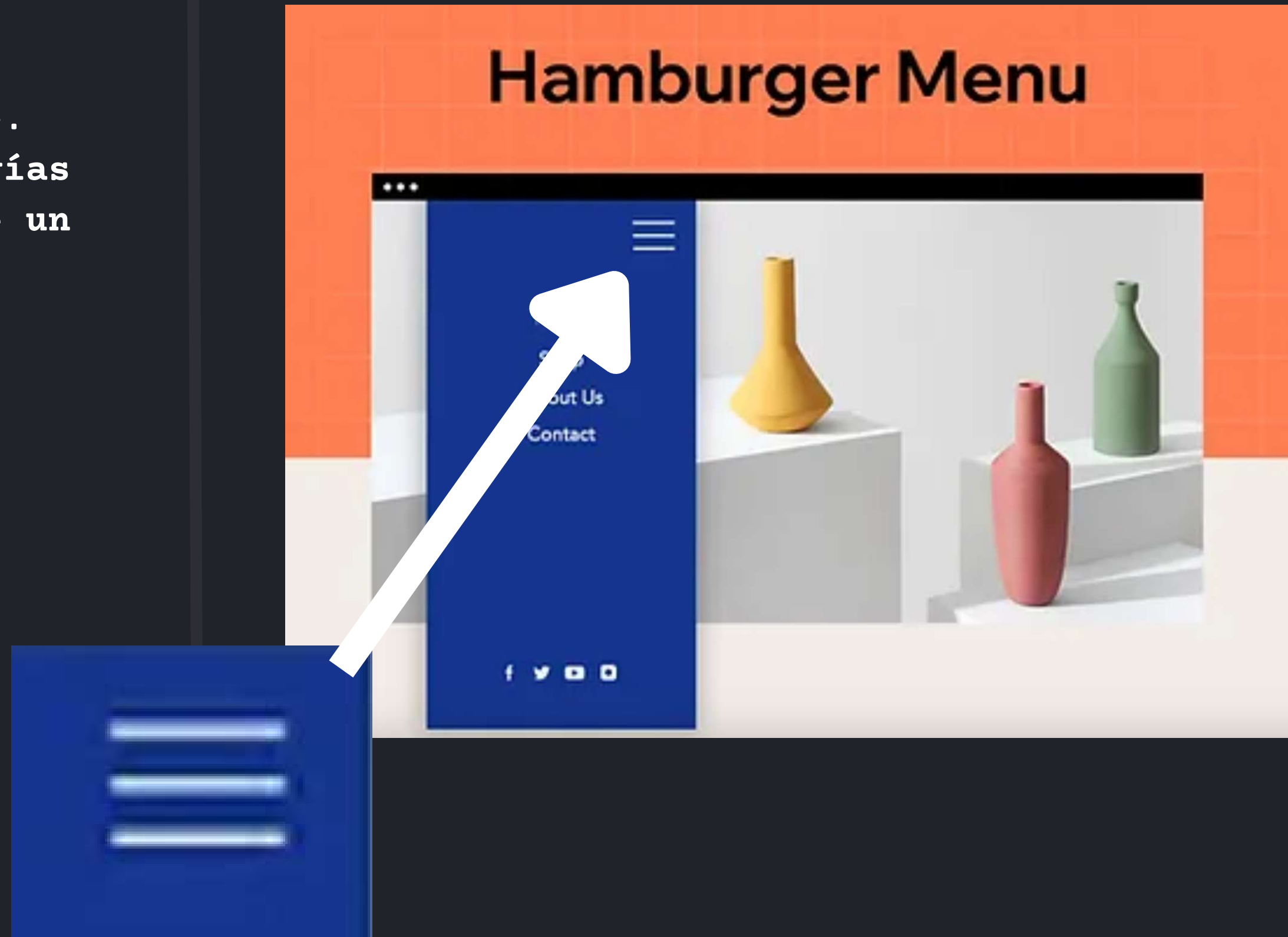
```
}
```

```
</script>
```



## `aria-haspopup {`

Controla la visibilidad de un menú desplegable o un elemento emergente. Este atributo **indica a las tecnologías de asistencia que el elemento tiene un menú emergente que puede ser desplegado.**



# aria-haspopup {

```
<button aria-haspopup="true" onclick="toggleMenu()">Opciones</button>
<div id="optionsMenu" hidden>
  <ul>
    <li><a href="#">Perfil</a></li>
    <li><a href="#">Configuración</a></li>
    <li><a href="#">Cerrar sesión</a></li>
  </ul>
</div>

<script>
function toggleMenu() {
  var menu = document.getElementById("optionsMenu");
  menu.hidden = !menu.hidden;
}
</script>
```





# aria-disabled {

Este atributo indica a las tecnologías de asistencia que, **aunque el elemento está presente visualmente, su interacción está desactivada.**



Diagrama de una interfaz de usuario con un campo de texto y un botón. El campo de texto está etiquetado como "Nombre de usuario" y contiene una línea horizontal, indicando que está vacío. El botón, etiquetado como "Enviar", está desactivado, lo que se evidencia por su color gris opaco y la falta de un efecto de sombra.



Diagrama de una interfaz de usuario con un campo de texto y un botón. El campo de texto está etiquetado como "Nombre de usuario" y contiene el texto "Ylenia I". El botón, etiquetado como "Enviar", está activo, lo que se evidencia por su color azul oscuro y la presencia de un efecto de sombra.

# aria-disabled {

```
<form id="myForm">
  <input type="text" id="username" name="username" placeholder="Nombre de usuario">
  <button type="submit" id="submitButton" aria-disabled="true" disabled>Enviar</button>
</form>
```

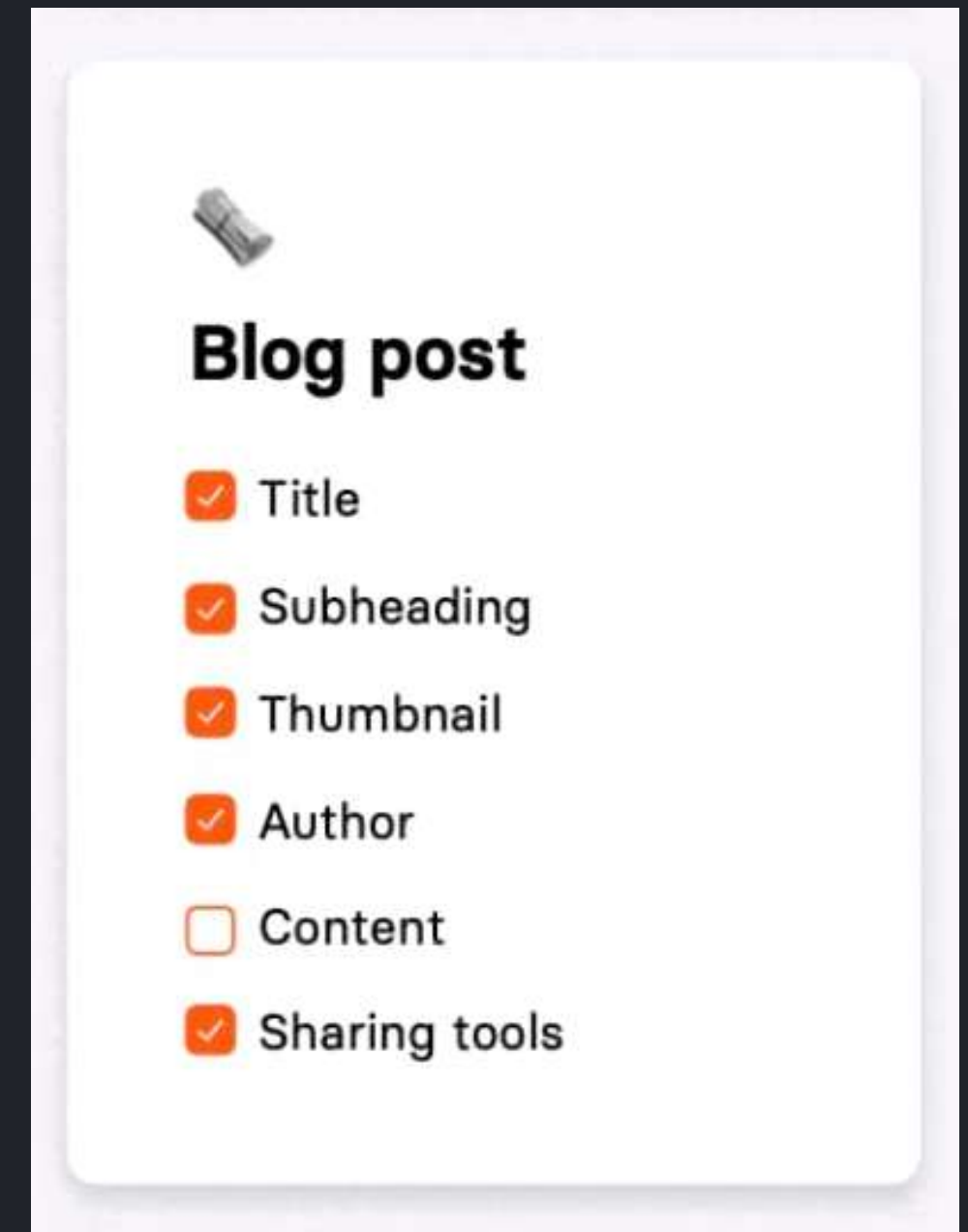
```
<script>
  document.getElementById('username').addEventListener('input', function(e) {
    const button = document.getElementById('submitButton');
    if (e.target.value === '') {
      button.setAttribute('aria-disabled', 'true');
      button.disabled = true;
    } else {
      button.setAttribute('aria-disabled', 'false');
      button.disabled = false;
    }
  });
</script>
```



# aria-checked {

Este atributo se utiliza para elementos que tienen un **estado de "marcado"** que puede ser cambiado, como **checkboxes** o **radio buttons**. Un elemento con `aria-checked="true"` comunica que la opción ha sido seleccionada, mientras que `aria-checked="false"` significa lo contrario.

```
<div class="checklist"
  <h2 Blog post </h2>
  <ul>
    <li aria-checked="true"><span class="checked">✓</span> Title</li>
    <li aria-checked="true"><span class="checked">✓</span> Subheading</li>
    <li aria-checked="true"><span class="checked">✓</span> Thumbnail</li>
    <li aria-checked="true"><span class="checked">✓</span> Author</li>
    <li aria-checked="false"><span class="unchecked">X</span> Content</li>
    <li aria-checked="true"><span class="checked">✓</span> Sharing tools</li>
  </ul>
</div>
```





# aria-pressed {

Se utiliza para indicar el estado de un **botón que puede ser presionado o no**, similar a un interruptor de encendido/apagado.

- **Botones de acción:** como un botón de "Silenciar" o "Seguir/Dejar de seguir"
- **Botones de selección de herramientas:** como en un editor de texto donde puedes activar o desactivar el negrita o la cursiva.
- **Controles de tipo interruptor**

```
<button aria-pressed="false" onclick="toggleButton(this)">Me gusta</button>
```

```
<script>
```

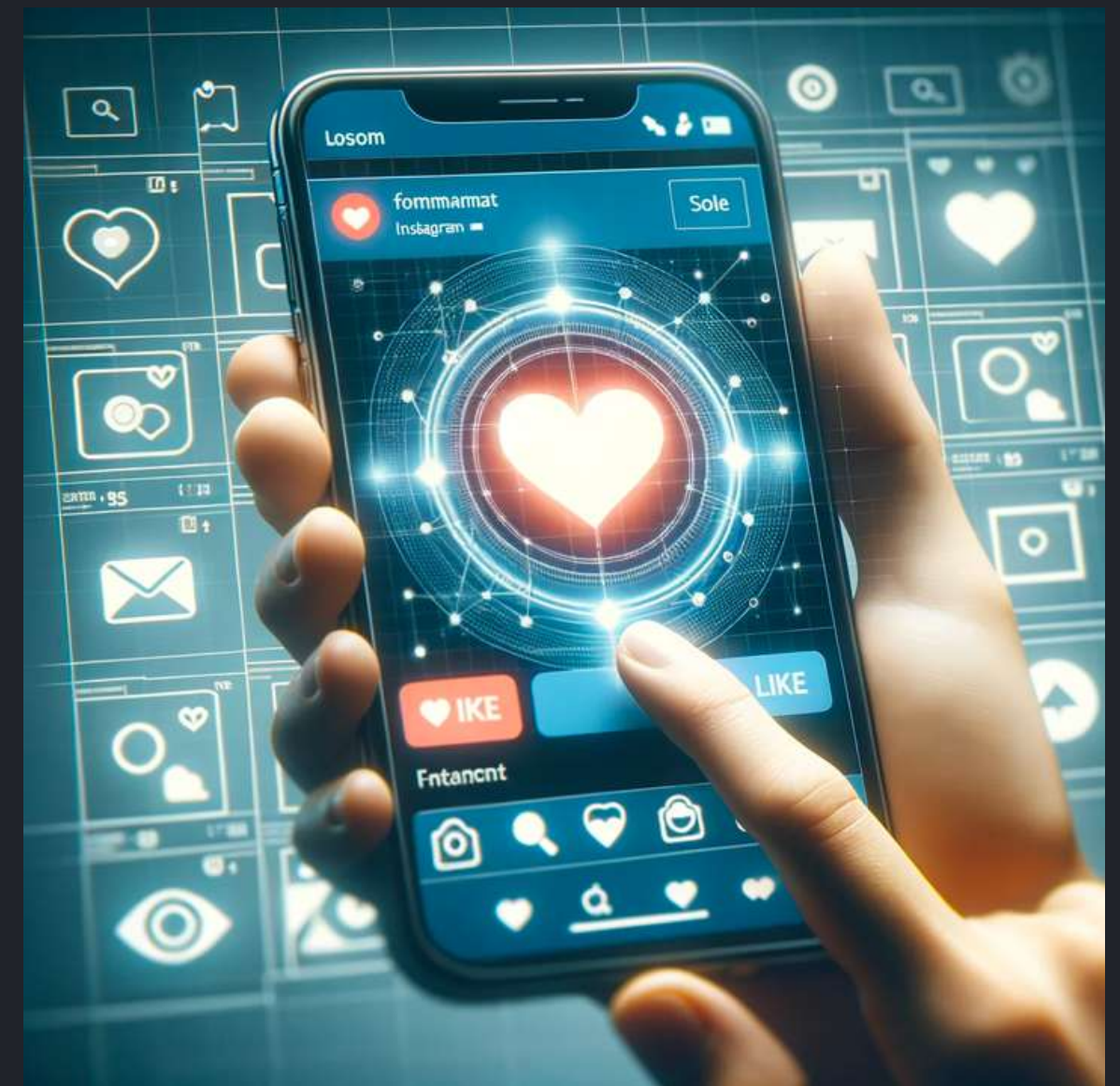
```
function toggleButton(button) {
```

```
    var isPressed = button.getAttribute("aria-pressed") === "true";
```

```
    button.setAttribute("aria-pressed", String(!isPressed));
```

```
}
```

```
</script>
```

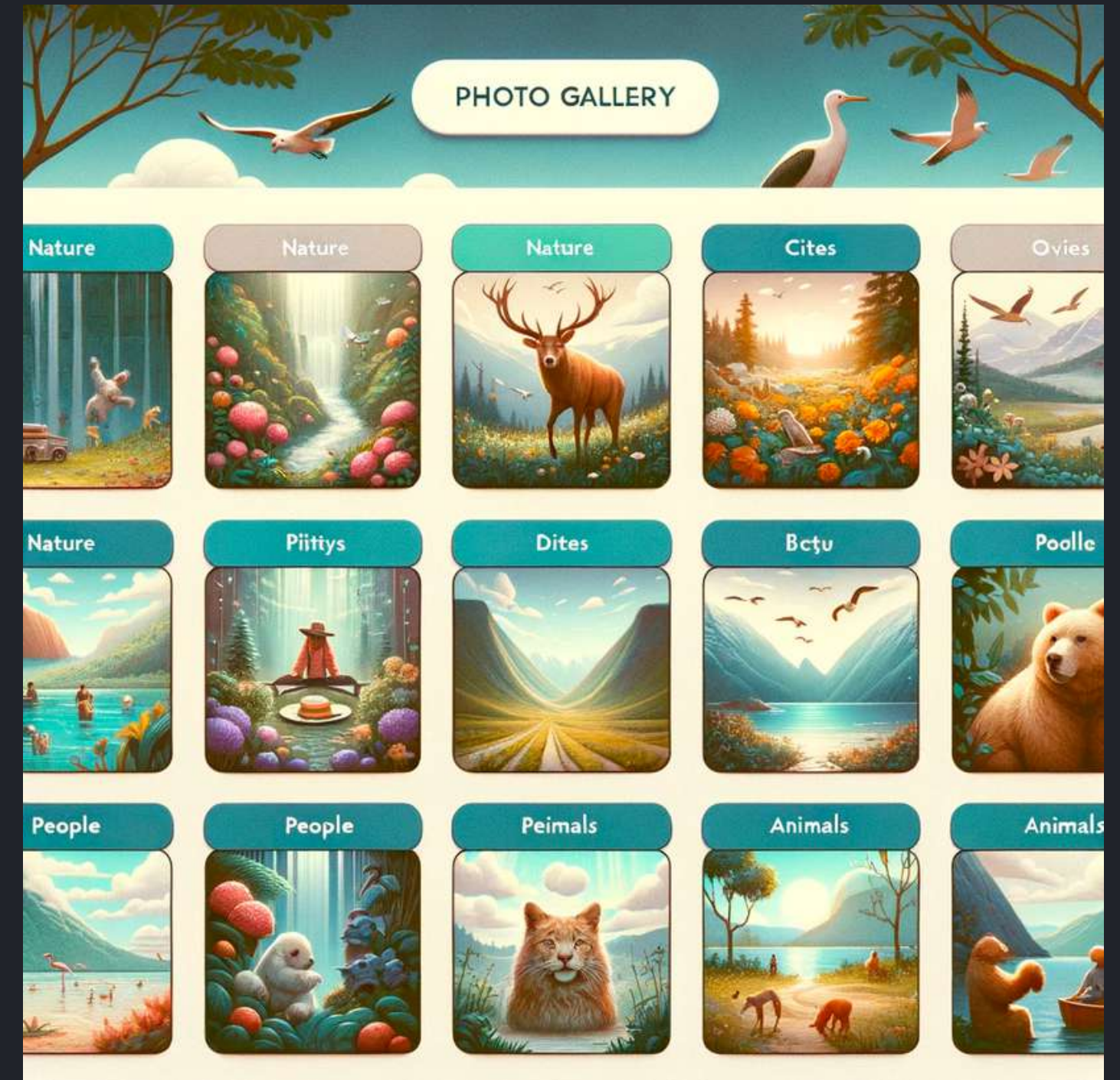




# aria-selected {

Se utiliza para indicar que un elemento (como una opción en una lista, una pestaña en un conjunto de pestañas, o un elemento en una cuadrícula) **está seleccionado**.

- **Navegación por pestañas:** Indica qué pestaña está actualmente seleccionada en un grupo de pestañas.
- **Listas y cuadrículas:** Denota el elemento actualmente seleccionado en una lista o cuadrícula.
- **Elementos de selección:** Como ítems de un menú desplegable, opciones en un menú lateral, etc.

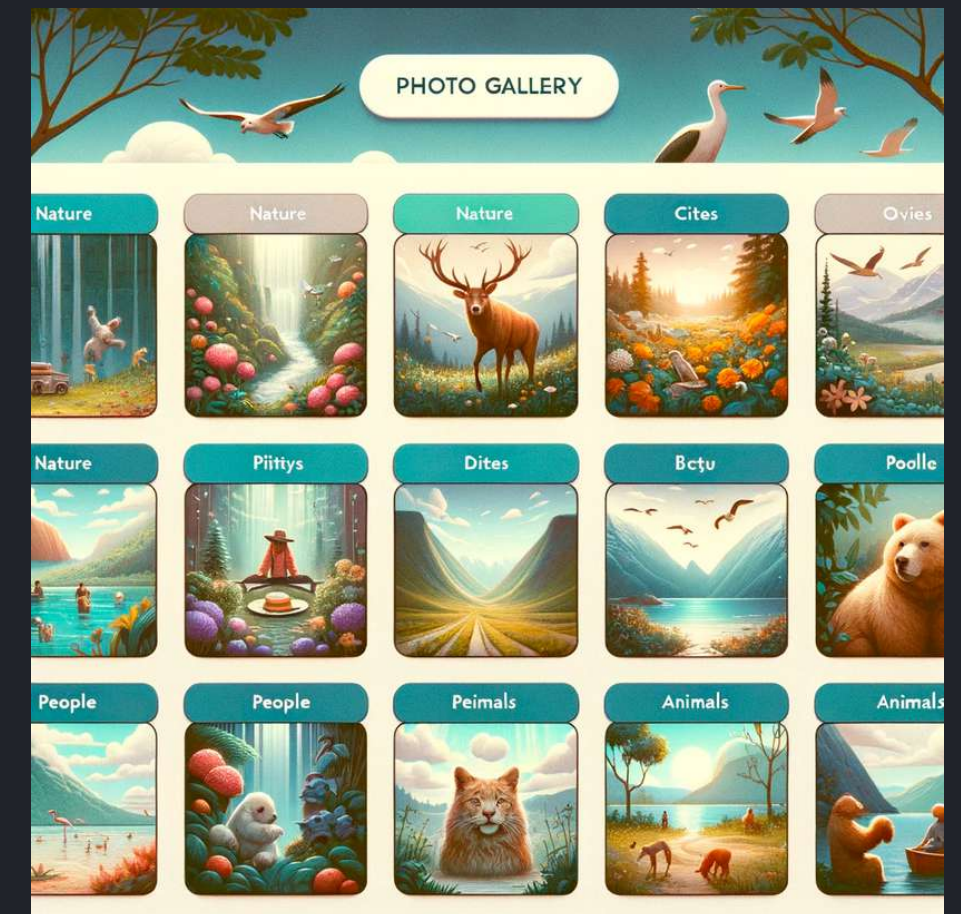




# aria-selected { html

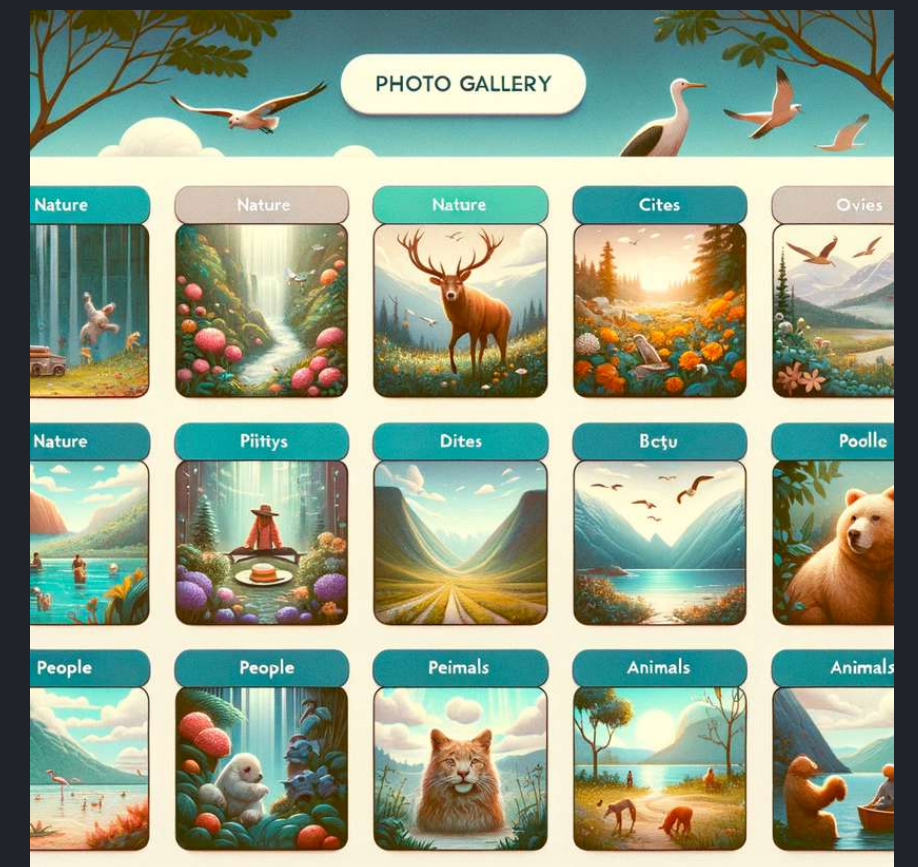
```
<div class="button-container">
  <!-- Añade aria-selected="true" al botón inicialmente seleccionado -->
  <button onclick="filterCategory('all')" aria-selected="true">Todos</button>
  <button onclick="filterCategory('nature')" aria-selected="false">Naturaleza</button>
  <button onclick="filterCategory('cities')" aria-selected="false">Ciudades</button>
  <button onclick="filterCategory('people')" aria-selected="false">Personas</button>
</div>
```

```
<div class="gallery-container">
  <div class="nature"></div>
  <div class="nature"></div>
  <div class="cities"></div>
  <div class="cities"></div>
  <div class="people"></div>
  <div class="people"></div>
</div>
```



# aria-selected { Script

```
function filterCategory(category) {  
  // Actualiza el display de los elementos basado en la categoría  
  const elements = document.querySelectorAll('.gallery-container div');  
  elements.forEach(el => {  
    el.style.display = category === 'all' || el.classList.contains(category) ?  
    '' : 'none';  
  });  
  
  // Actualiza aria-selected en los botones  
  const buttons = document.querySelectorAll('.button-container button');  
  buttons.forEach(btn => {  
    const isSelected = btn.textContent.toLowerCase() === category ||  
(btn.textContent.toLowerCase() === 'todos' && category === 'all');  
    btn.setAttribute('aria-selected', isSelected.toString());  
  });  
}
```



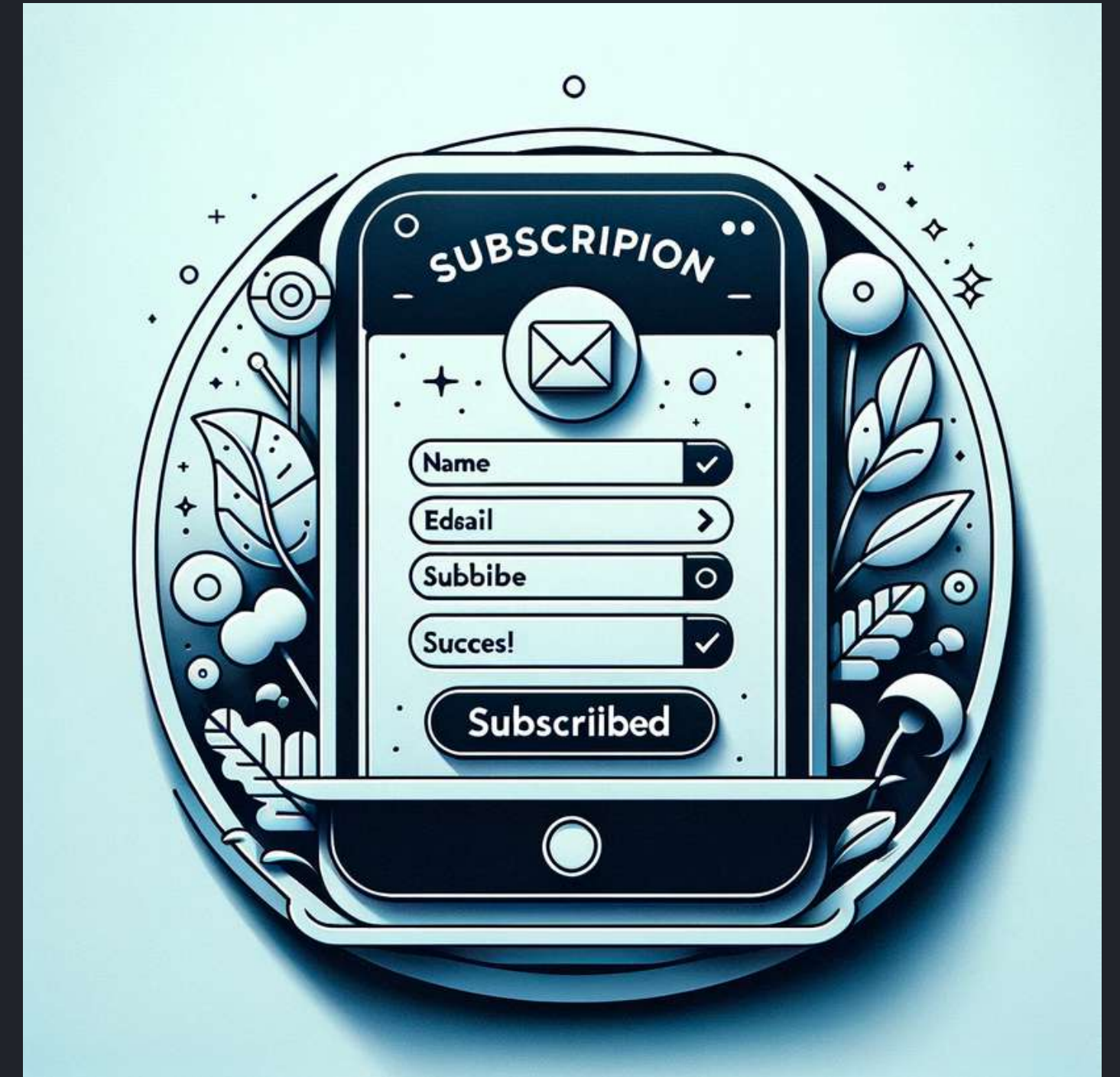


# aria-selected {

Un ejemplo común y simple del uso de aria-live es **mostrar un mensaje de confirmación después de que un usuario realiza una acción, como enviar un formulario**. En este caso, `aria-live="polite"` se utiliza para informar al usuario sobre el estado del envío sin interrumpir su flujo de trabajo actual.

Los valores más comunes para `aria-live` son:

- **off** (el valor predeterminado): Los cambios no se anuncian.
- **polite**: Los cambios se anuncian solo si no interrumpen la tarea actual del usuario.
- **assertive**: Los cambios se anuncian inmediatamente, posiblemente interrumpiendo al usuario.



# aria-selected { html

```
<h2>Formulario de Suscripción</h2>
<form id="subscriptionForm" onsubmit="submitForm(); return
false;">
  <label for="email">Correo Electrónico:</label>
  <input type="email" id="email" required>
  <button type="submit">Suscribirse</button>
</form>

<!-- El elemento para mostrar el mensaje de estado con
aria-live -->
<div id="statusMessage" aria-live="polite" style="margin-
top: 20px;"></div>
```





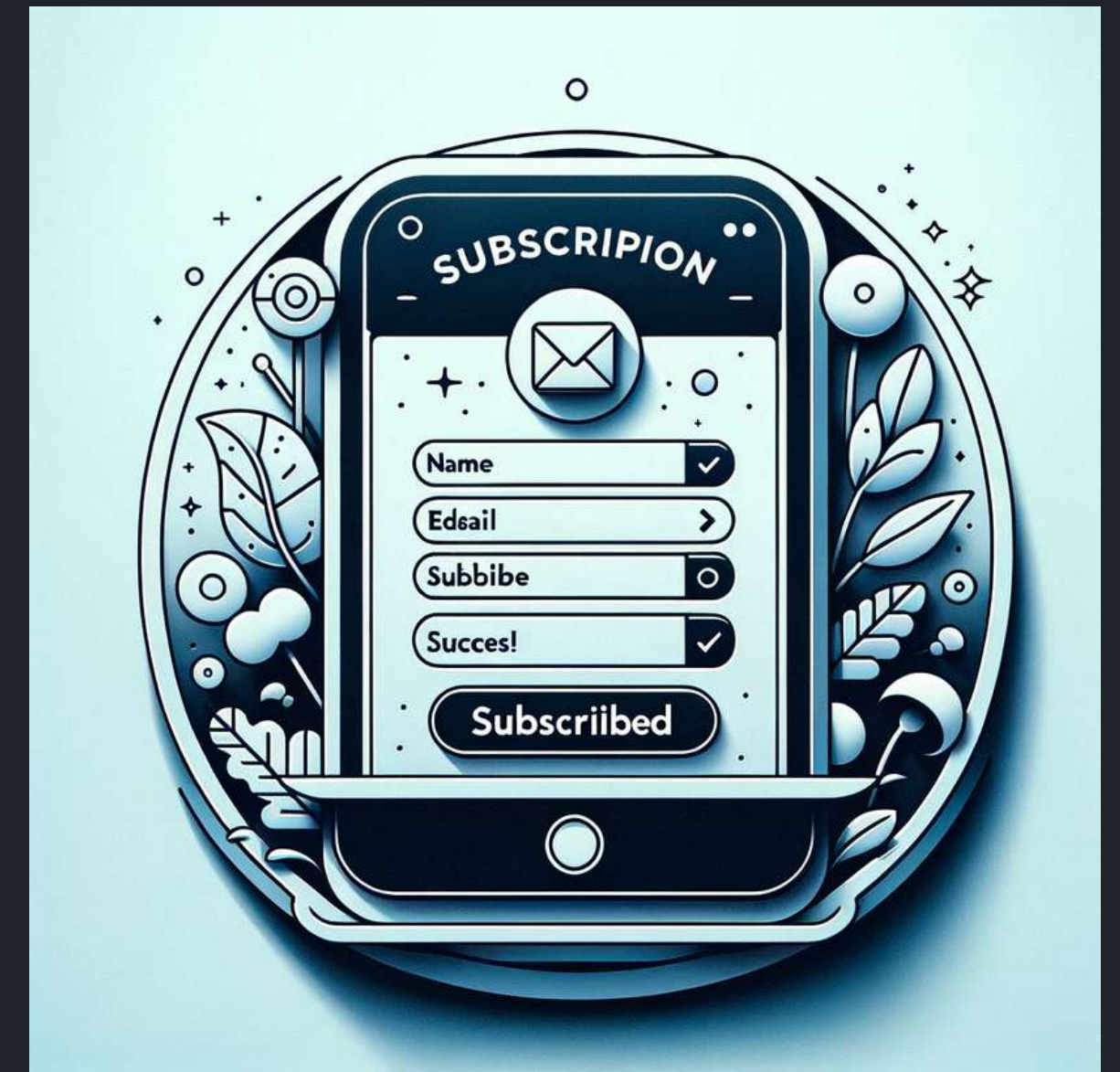
# aria-selected { Scrip

```
<script>
function submitForm() {
  // Obtener el elemento del mensaje de estado
  var statusMessage = document.getElementById('statusMessage');

  // Simula el procesamiento del formulario
  statusMessage.innerText = 'Enviando...';

  // Simula una respuesta del servidor después de un retraso
  setTimeout(function() {
    statusMessage.innerText = '¡Suscripción exitosa!';
  }, 2000);
}
</script>

</body>
</html>
```






# aria-describedby {

se utiliza para dar a los usuarios de tecnologías de asistencia más contexto sobre un elemento, como:

- Proporcionar más detalles sobre la función o el propósito de un botón.
- Ofrecer instrucciones adicionales para campos de entrada en formularios.
- Explicar más profundamente el contenido o la data representada por elementos interactivos.

```
<form>
  <label for="username">Nombre de usuario:</label>
  <input type="text" id="username" aria-describedby="ayuda para el usuario">
  <div id="usernameHelp">Tu nombre de usuario debe contener entre 4 y 12 caracteres,
sin espacios ni caracteres especiales.</div>

  <button type="submit">Registrarse</button>
</form>
```



\* Nombre:

alber

\* Correo: \* Campo obligatorio

Correo...

\* Mensaje: \* Campo obligatorio

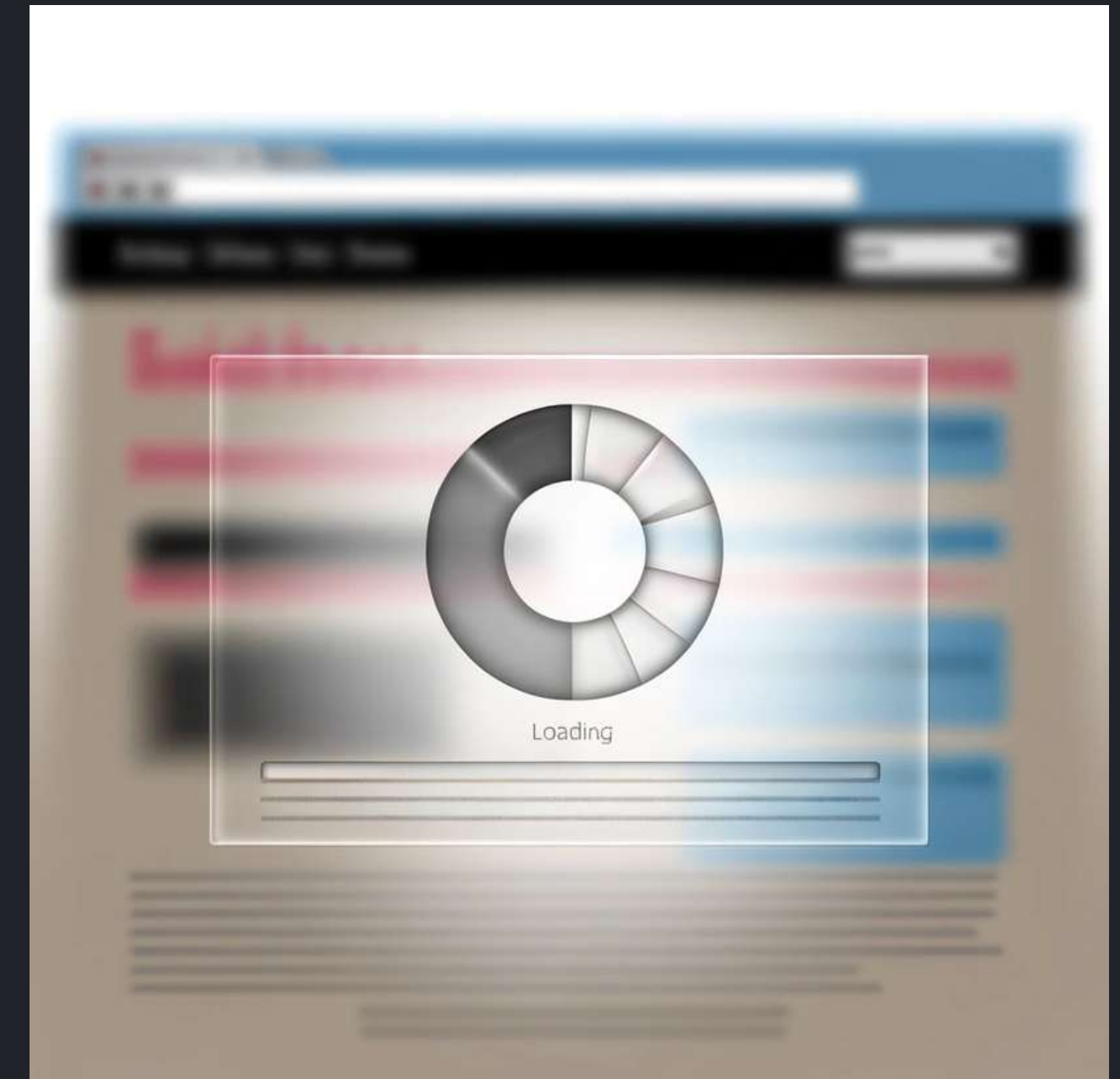
Mensaje...

Enviar

# aria-busy {

Se utiliza para **indicar a las tecnologías de asistencia que un elemento está siendo actualizado y que los cambios en este elemento deben ser ignorados temporalmente**. Esto es especialmente útil en aplicaciones web donde partes del contenido se actualizan dinámicamente sin recargar toda la página.

- **Indicar actividad o carga:** Se utiliza para indicar que un área específica de la página está siendo actualizada o cargada, y que los usuarios deben esperar hasta que el proceso haya terminado para interactuar con esa parte de la interfaz.
- **Mejorar la accesibilidad de aplicaciones dinámicas:** Ayuda a los usuarios de lectores de pantalla a entender cuándo el contenido está cambiando y podría no estar listo para ser leído o interactuado.



## aria-busy {

```
<div id="feedContainer" aria-busy="false"> <!-- Contenido del feed aquí -->
</div>
<button onclick="updateFeed()">Actualizar Feed</button>

<script>
function updateFeed() {
  const feedContainer = document.getElementById('feedContainer');
  // Marcar el contenedor del feed como "ocupado" mientras se carga el contenido
  feedContainer.setAttribute('aria-busy', 'true');
  fetch('/get-new-feed-items') // Simulación de una solicitud para obtener nuevos ítems
    .then(response => response.json())
    .then(data => {
      // Actualizar el contenido del feed aquí... Por ejemplo, añadir nuevos ítems al
      // contenedor del feed. Una vez completada la actualización, marcar el feed como no ocupado
      feedContainer.setAttribute('aria-busy', 'false');
    });
}
</script>
```

## Atributo ARIA-role {

Este atributo se utiliza para definir roles semánticos específicos para elementos HTML, lo cual ayuda a las tecnologías de asistencia, como los lectores de pantalla, a entender el propósito de cada elemento dentro de una página web. Al conocer el rol de un elemento, las tecnologías de asistencia pueden proporcionar información más detallada o interacciones específicas a los usuarios.

### Todos los atributos role

## role="navigation"

Indica que el elemento es una sección de navegación principal.

## role="button"

Se utiliza en un elemento <div>, <span>, o <a> para indicar que actúa como un botón.

## role="alert"

Señala que un elemento debe ser tratado como una alerta y que su contenido debe ser anunciado de manera inmediata por las tecnologías de asistencia.

## role="search"

Identifica una sección de la página destinada a la búsqueda, ayudando a los usuarios a localizar rápidamente el mecanismo de búsqueda en una página.

}

# 3 niveles de ACCESIBILIDAD web según WCAG 2.1 {

## NIVEL A, Accesibilidad BÁSICA

- Proporcionar **textos alternativos para las imágenes no decorativas**.
- Garantizar que todos los elementos de la interfaz sean **operables a través del teclado**.
- Usar etiquetas y atributos adecuados en el código HTML para asegurar una **correcta estructuración del contenido**.

## Administraciones públicas

## NIVEL AA Accesibilidad MEJORADA

Todo lo anterior +

- Asegurar un **contraste de color** suficiente entre el texto y el fondo.
- Proporcionar **títulos de página** que sean descriptivos y útiles, así como **etiquetas para formularios, campos y botones**.

## NIVEL AAA Accesibilidad EXCEPCIONAL

Todo lo anterior +

- Ofrecer varias formas de ayuda para entender y navegar por el contenido, como un **glosario de términos especializados**.
- Proporcionar **subtítulos en vivo para cualquier contenido de audio en directo**.
- Permitir que los usuarios ajusten el **espaciado del texto** para mejorar la legibilidad.

}



¿Cómo se que nivel alcanza mi web? {



Crterios de  
cada nivel



AUDITORIAS



Análisis auto.  
Lighthouse

}

# Otras herramientas de accesibilidad {



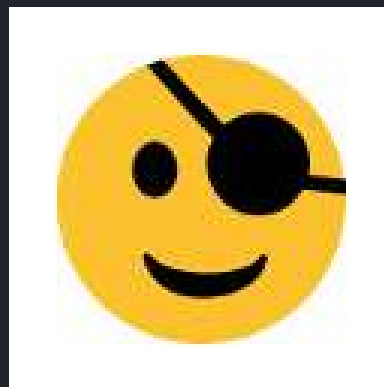
Accessibilty a  
powerful web  
assintant



Web Disability  
simulator



Wave evaluation  
tool



ARIA DevTools



Tab11y



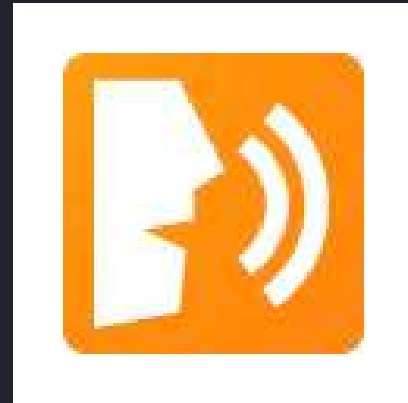
HTML Heading  
Highlighter



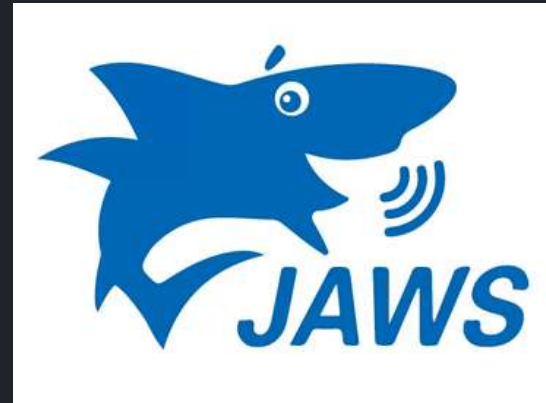
HeadingsMap

}

# LECTORES DE PANTALLA {



Screen reader  
extension Chrome



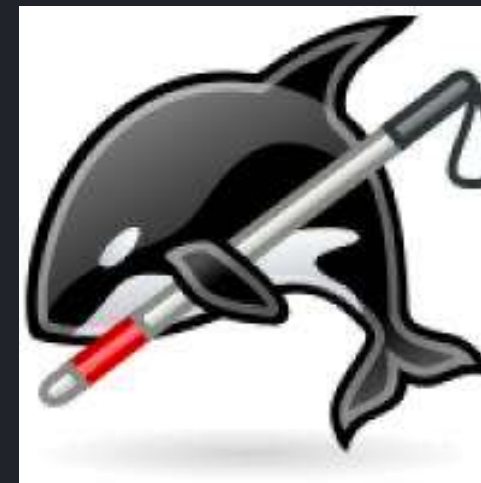
JAWS (Windows)  
de pago



NVDA (Windows)  
gratis



VoiceOver (macOS)



Orca (Linux)

}

# Accesibilidad de Angular y Bootstrap {

## Angular Material

Al utilizar Angular Material, asegúrese de utilizar sus componentes correctamente y de que no estén implementados de forma inaccesible. Los componentes de Angular Material tienen conocimiento incorporado de las mejores prácticas de accesibilidad.



## Bootstrap

CUIDADO Bootstrap no agrega ninguno de los atributos de rol y aria necesarios para los verdaderos menús ARIA. Los autores deberán incluir ellos mismos estos atributos más específicos.



}

# Practica formularios {

| Default  | Val-Error  | Val-Success  |
|--|--|--|
| <div>Default</div> <div>Placeholder</div> <div>help or description text</div>  | <div>Default</div> <div>Placeholder</div> <div>help or description text</div>  | <div>Default</div> <div>Placeholder</div> <div>help or description text</div>  |
| <div>Focus</div> <div>Placeholder</div> <div>help or description text</div>    | <div>Focus</div> <div>Placeholder</div> <div>help or description text</div>    | <div>Focus</div> <div>Placeholder</div> <div>help or description text</div>    |
| <div>Hover</div> <div>Placeholder</div> <div>help or description text</div>    | <div>Hover</div> <div>Placeholder</div> <div>help or description text</div>    | <div>Hover</div> <div>Placeholder</div> <div>help or description text</div>    |
| <div>Disabled</div> <div>Placeholder</div> <div>help or description text</div> | <div>Disabled</div> <div>Placeholder</div> <div>help or description text</div> | <div>Disabled</div> <div>Placeholder</div> <div>help or description text</div> |



## Diseña un formulario UX accesible

First Name:

Last Name:

Email:   
(Your email address will be your username)

Re-type Email:

Password:   
(Min. 8 characters, 1 number, case-sensitive)

Re-type Password:

Address:

City:

State:

Zip Code:

Phone:    
No spaces or dashes

Date of Birth:    

Gender:  

Security Question:  

Security Answer:   
(Not case-sensitive)



**Personal Information**

First Name:

Last Name:

Date of Birth:    

Gender:  

**Account Information**

Email:   
(Your email address will be your username)

Re-type Email:

Password:   
(Min. 8 characters, 1 number, case-sensitive)

Re-type Password:

Security Question:  

Security Answer:   
(Not case-sensitive)

**Contact Information**

Address:

City:

State:

Zip Code:

Phone:    
No spaces or dashes



# PRÁCTICA: etiqueta <fieldset> y <legend> {

## ¿Qué es?

El elemento `fieldset` (grupo de campos) permite organizar en grupos los campos de un formulario.

## Como organizarse.

```
<form>
```

```
  <fieldset>
```

```
    <legend>Información Personal</legend>
```

```
    Nombre: <input name='nombre' type='text' tabindex='1'>
```

```
    Apellidos: <input name='apellidos' type='text' tabindex='2'>
```

```
  </fieldset>
```

```
  <fieldset>
```

```
    <legend>edad</legend>
```

```
    <input type='checkbox' tabindex='20'
```

```
      name='edad' value='20-39' > 20-39
```

```
    <input type='checkbox' tabindex='21'
```

```
      name='edad' value='40-59' > 40-59
```

```
    <input type='checkbox' tabindex='22'
```

```
      name='edad' value='60-79' > 60-79
```

```
  </fieldset>
```

```
</form>
```

```
}
```

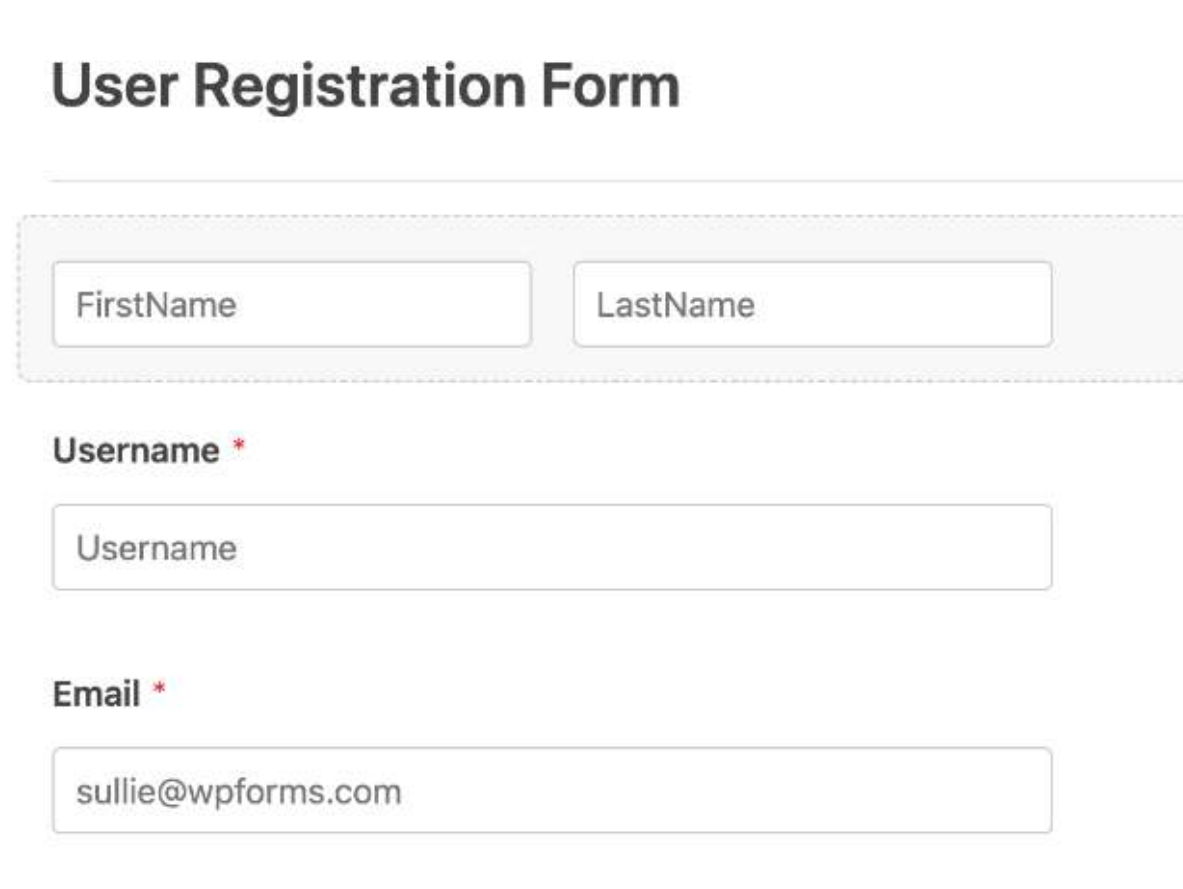
# PRÁCTICA: atributo placeholder="" {

## ¿Qué es?

El atributo placeholder se define como un consejo o indicación corta que puede consistir en una sola palabra o en una pequeña frase. Es aceptado en los elementos input y textarea. Esta indicación es mostrada por el navegador cuando el elemento no tiene valor asignado o este sea un valor vacío.

## Como organizarse.

```
<input type="email" name="email" placeholder="Por ejemplo, juan@dominio.com">
```



A screenshot of a 'User Registration Form' with a light gray background. The form contains several input fields with placeholder text. At the top, there's a title 'User Registration Form'. Below it, a dashed border encloses two input fields: 'FirstName' and 'LastName'. Underneath, there's a 'Username' label with a red asterisk, followed by an input field with the placeholder 'Username'. Below that is an 'Email' label with a red asterisk, followed by an input field containing the text 'sullie@wpforms.com'.

}

# PRÁCTICA: Orden tabulación{

## taba11y

Extensión para Chrome de orden de tabulación de la página.



}

# Recursos muy TOP

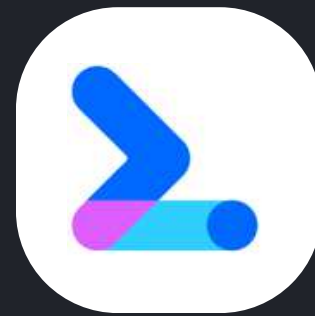
## PHIND

ChatGPT para  
programadores



## web.dev

Curso  
accesibilidad



## Dribbble

Inspiración  
diseños web



## Uiverse

HTML y CSS  
GRATIS de comp.



## USERWAY

Witget para la  
accesibilidad



## @tarreddev

Perfil  
instagram



## 100dayscss

Curso CSS



## @elrincondeldev

Perfil  
instagram



}



# Recursos muy TOP

## Htmlrev

Platillas web  
gratis: html,  
angula,  
bootstrap

HTMLrev

## Uxcel

Aprende UX  
(práctico)



## Dribbble

Inspiración  
diseños web



## POST INTERESANTES

- Galeria de imágenes TOP, con código
- Menú UX/UI, con código
- Diseño estilo cuadrícula
- El css y html de los elementos de Figma
- Cards responsive (código).

}

# Extensiones Visual Studio Core TOP



## Error lens

Te permite ver los errores de tu código en la misma línea.



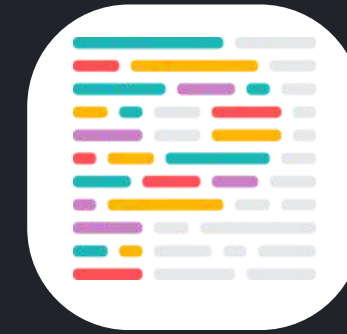
## Thunder Client

Vas a poder probar APIs sin tener que salir de visual studio code



## Prettier

Formatear código de forma rápida y limpia



## aterial Icon Theme

Iconos personalizados para tus archivos en visual studio code



## Console ninja

Previsualizar las imágenes sin salir del código



## Console ninja

Muestra el output de los consol log



```
<!--FPS 2ºDAW-Diseño de Interfaces Web -->
```

Gracias, mañana  
UX/UI {

```
<Por="Ylenia Acuña y Llara Ramos"/>
```

}