

The background is a dark blue-grey color with a subtle pattern of small, light-grey dots arranged in a grid. Overlaid on this is a large, dark blue circle with a thick white border. Inside this circle, there are several white molecular structures, which are clusters of small circles connected by thin lines, resembling chemical or network diagrams. The text "DISEÑO DE INTERFACES" is centered within the circle in a bold, white, sans-serif font.

# **DISEÑO DE INTERFACES**

La accesibilidad es esencial para los desarrolladores y organizaciones que quieren crear sitios y herramientas web de calidad para no excluir a ninguna persona del uso de sus productos y servicios

La accesibilidad web significa que las páginas, herramientas o tecnologías están diseñadas y desarrolladas para que personas con discapacidad puedan utilizarlas, esta abarca a todas las discapacidades:

- Auditivo
- Cognitivo
- Neurológico
- Físico
- Habla
- Visual

Aunque también beneficia a las personas sin discapacidad, entre muchas ventajas encontramos:

- Personas que utilicen teléfonos, relojes, etc... (pantallas pequeñas)
- Personas mayores con habilidades cambiantes
- Personas con discapacidad temporal (brazo roto)
- Personas con situaciones situacionales (Sitio con luz que no se pueda ver bien la pantalla)
- Personas que utilizan conexión lenta de internet



## Asociar una etiqueta con cada control de formulario

Use un `for` atributo en el `<label>` elemento vinculado al `id` atributo del elemento de formulario, o use atributos WAI-ARIA. En situaciones específicas, puede ser aceptable ocultar `<label>` elementos visualmente, pero en la mayoría de los casos se necesitan etiquetas para ayudar a todos los lectores a comprender la entrada requerida.

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

## Incluir texto alternativo para las imágenes

Asegúrese de que se agregue texto alternativo para las imágenes a todas las imágenes informativas y funcionales. Use texto alternativo vacío, `alt=""` para imágenes decorativas, o inclúyalos en el CSS en su lugar. Las alternativas de texto suelen ser proporcionadas por los responsables del contenido escrito.

### Tipos de Imágenes

**Imágenes Formativas:** Representan información, la alternativa de texto debe ser una descripción que transmita información esencial por la imagen

**Imágenes Decorativas:** No es importante para la página por o que tendremos que dejar el campo nulo (`alt=""`)

**Imágenes Funcionales:** El texto alternativo debe describir la funcionalidad que represente esta imagen

**Imágenes de texto:** La alternativa de texto debe contener el mismo texto de la imagen

**Imágenes Complejas:** Gráficos o diagramas, explicar texto completo con información muy detallada como texto

**Grupos de Imágenes:** Si varias imágenes transmiten la misma información el texto debe describir todo el grupo

**Mapas de Imagen:** La alternativa de texto se puede hacer cuando pulses click en un área que describa su propósito o enlace

Es importante para personas que usan lectores de pantalla, software de entrada de voz, navegan por sitios web habilitados para voz, usuarios de web móvil, optimización de motores de búsqueda

## Identificar el idioma de la página y los cambios de idioma

Indique el idioma principal de cada página mediante el `Lang` atributo en la `html` etiqueta, por ejemplo `<html lang="en">`. Utilice el `Lang` atributo en elementos específicos cuando el idioma del elemento difiera del resto de la página.

## Utilice el marcado para transmitir significado y estructura

Use el marcado apropiado para encabezados, listas, tablas, etc. HTML5 proporciona elementos adicionales, como `<nav>` y `<aside>`, para estructurar mejor su contenido. Los roles WAI-ARIA pueden proporcionar un significado adicional, por ejemplo, utilizar `role="search"` para identificar la funcionalidad de búsqueda. Trabaje con diseñadores y escritores de contenido para acordar significados y luego utilícelos de manera consistente.

Ejemplo: uso de HTML para proporcionar estructura y significado

```
<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>
```

Ejemplo: campo de búsqueda usando WAI-ARIA

```
<form action="#" method="post">
  <div role="search">
    <label for="search">Search for</label>
    <input type="search" id="search" aria-describedby="search-help">
    <div id="search-help">Search records by customer id or name</div>
    <button type="submit">Go</button>
  </div>
</form>
```

Regiones de página: Identificar y marcar las regiones

Etiquetado de regiones: Para permitir a los usuarios que las distinguan y puedan acceder a ellas

Encabezados: Se agregan para poder etiquetar secciones de páginas web según su importancia

Estructura del contenido: Utilizar elementos apropiados y significativos

Es importante para las personas con discapacidades cognitivas de aprendizaje, personas que usan lectores de pantalla, usuarios de teclado, personas que utilizan software que solo muestra contenido principal, personas con discapacidad visual, usuarios de móvil, personas que utilizan ciertos complementos del navegador

## Ayudar a los usuarios a evitar y corregir errores

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

```
<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>
```

## Refleja el orden de lectura en el orden del 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 tenga sentido.

## Escribir código que se adapte a la tecnología del usuario

Utilice un diseño receptivo para adaptar la pantalla a diferentes estados de zoom y tamaños de ventana gráfica, como en dispositivos móviles y tabletas. Cuando el tamaño de fuente aumenta al menos en 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 principal estén disponibles independientemente de la tecnología que se utilice.

Ejemplo: uso de consultas de medios para adaptar la navegación

```
@media screen and (min-width: 25em) {  
  #nav {  
    float: none;  
    width: auto;  
  }  
  #main {  
    margin-left: 0;  
  }  
}  
  
@media screen and (min-width: 43em) {  
  #nav {  
    float: left;  
    width: 24%;  
  }  
  #main {  
    margin-left: 27%;  
  }  
}
```



## Proporcionar significado para elementos interactivos no estándar.

Utilice WAI-ARIA para proporcionar información sobre la función y el estado de los widgets personalizados, como acordeones y botones personalizados. Por ejemplo, `role="navigation"` y `aria-expanded="true"`. 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 y estado del menú 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 cuando desarrolle elementos interactivos, como menús, información sobre el mouseover, acordeones plegables o reproductores multimedia. Úselo `tabindex="0"` para agregar un elemento que normalmente no recibe el foco, como `<div>` o `<span>`, en el orden de navegación cuando se usa para la interacción. Use secuencias de comandos para capturar y responder a los eventos del teclado.

```
var buttonExample = document.getElementById('example-button');

buttonExample.addEventListener('keydown', function(e) {
  // Toggle the menu when RETURN is pressed
  if(e.keyCode && e.keyCode == 13) {
    toggleMenu(document.getElementById('example-button-menu'));
  }
});

buttonExample.addEventListener('click', function(e) {
  // Toggle the menu on mouse click
  toggleMenu(document.getElementById('example-button-menu'));
});
```



## Evite CAPTCHA siempre que sea posible

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

- Proporcionando más de dos formas de resolver los CAPTCHA
- Proporcionar acceso a un representante humano que puede omitir CAPTCHA
- No requiere CAPTCHA para usuarios autorizados.

