PRÁCTICA ACCESIBILIDAD WEB – TEORÍA

LUIS VALLES PASTOR 2ºDAW

VISIÓN GENERAL:

Comprender la accesibilidad, su alcance y su impacto pueden hacer de ti un mejor programador web.
También descubrirás que muchas de estas técnicas te ayudarán a crear interfaces más agradables y fáciles de usar para todos los usuarios, no solos aquellos con discapacidades.
Cuando decimos que un sitio es accesible, queremos decir que el contenido del sitio está disponible y, literalmente, cualquiera puede manejar su funcionalidad.

Problemas de accesibilidad:
El texto tiene poco contraste (utilizar negrita), lo cual dificulta la lectura para usuarios con baja visión.
En un formulario asociar cada etiqueta con su campo, desde el punto de vista físico como en cuanto a la semántica. La posibilidad de activar un checkbox desde su propia etiqueta.
<input id="promo" type="checkbox"></input>
<labelfor="promo">Receivepromotionaloffers?</label>

WCAG - Pautas de accesibilidad a contenido web.
Principios:

- Perceptible: A los sentidos:visual, motriz, auditiva y cognitiva.
- Manejable: Por teclado, mouse, pantalla táctil, diseño responsive…
- Comprensible: Contenido relacionado con su etiqueta.
- Sólido: Sitio web consumible desde distintos navegadores.

Cada tipo de discapacidad puede ser circunstancial, temporal o permanente.

FOCO:

El foco se refiere a qué control en la pantalla (un elemento de entrada, como un campo, una casilla de verificación, botón o un vínculo) recibe actualmente la entrada desde el teclado, y desde el portapapeles si pegas contenido. Por lo tanto, una estrategia de foco bien implementada te asegura que todos los que usen tu app tengan una mejor experiencia. En general, no hay necesidad de enfocar algo con lo que el usuario no puede interactuar.
Cómo usuario, puedes controlar qué elemento tiene actualmente el foco usando Tab, Shift+Tab.
Debe seguir un correcto orden de tabulación, lo sigue según su posición en el DOM, puedo variar según el posicionamiento del elemento en CSS.
Puedes usar document.activeElement desde la consola para encontrar qué elemento tiene actualmente el foco.

Cuando descubras qué elemento fuera de pantalla tiene el foco, puedes configurarlo en display: none o visibility: hidden, y luego volver a configurarlo en display: block o visibility: visible antes de mostrarlo al usuario.

En ciertas ocasiones querrás modificar el orden de tabulación, para estos casos, puedes usar el atributo HTML tabindex para establecer explícitamente la posición de pestaña de un elemento.
tabindex="0": El elemento puede tomar el foco si se presiona la tecla Tab.
tabindex="-1": El elemento puede tomar el foco mediante una llamada a su método focus().
El tabindex itinerante funciona configurando tabindex en -1 para todos los elementos secundarios excepto el que se encuentra activo en el momento (0). El método .focus() llama al que posea valor 0.

SEMÁNTICA:

Un dispositivo affordance es cualquier objeto que le ofrezca a su usuario la posibilidad de realizar una acción. Mientras mejor esté diseñado el affordance, más obvio o intuitivo será su uso.
El lector de pantalla brinda información sobre:
- El rol o tipo de elemento, si se especifica (debería).
- El nombre del elemento, si lo tiene (debería).
- El valor del elemento, si lo tiene (puede o no tenerlo).
- El estado del elemento, p. ej., si está habilitado o inhabilitado (si corresponde).

Todas las imágenes deberían tener un atributo alt. Las imágenes importantes deberían tener texto de alt descriptivo que describa en forma concisa qué es una imagen, mientras que las imágenes decorativas deberían tener atributos de alt vacíos.

HTML5 introdujo nuevos elementos que ayudan a definir la estructura de semantic de la página: header, footer, nav, article, section, main y aside. Los elementos estructurales de semantic reemplazan los múltiples bloques div repetitivos, y brindan una forma más limpia y descriptiva.

ESTILOS:

WAI ARIA:
Trata de un conjunto de atributos para ayudar a mejorar la semántica de un sitio web, da sentido a ciertos elementos que no son nativos de HTML.
Trata de expresar los semantics que el HTML no puede expresar por su cuenta.
Ejemplo: Añade los atributos role y aria-checked.

```
<li tabindex="0" class="checkbox" role="checkbox" checked aria-checked="true">
  Receive promotional offers
</li>
```

Es importante comprender que no es necesario definir semantics predeterminados.
Un elemento <input type="checkbox"> HTML estándar no necesita un atributo de ARIA role="checkbox" adicional para anunciarse correctamente.

Roles.

¿Qué puede hacer ARIA?
aria-label: Agrega texto descriptivo.
aria-labelledby: es un ejemplo de un "atributo de relación", permite especificar la ID de otro elemento del DOM como etiqueta de un elemento.
aria-owns: presenta un elemento separado del DOM como un elemento secundario del actual.
aria-activedescendant: Así como el elemento activo de una página es el que tiene el foco, la configuración de un descendiente activo nos permite decirle a la tecnología asistencial que debería presentarse al usuario un elemento como elemento en foco cuando el elemento principal es el que tiene el foco.
aria-describedby: hacer referencia a elementos que no son visibles de los usuarios de tecnología asistencial.
aria-posinset ("posición en set") y aria-setsize ("tamaño de set") tratan de definir una relación entre elementos relacionados en un set, como una lista.
ARIA puede hacer que ciertas partes de la página estén "vivas", informando inmediatamente: aria-live="true".
Tiene tres valores permisibles: polite, assertive y off.

- polite: alerta al usuario sobre el cambio. Es muy bueno usarlo si hay algo importante, pero no urgente.
- assertive: alerta al usuario sobre el cambio de inmediato.
- off: suspende aria-live temporalmente.

aria-hidden: informa del estado.

· Extensiones:
Aumentar contraste (tema oscuro): A través de un tema programado dentro del sitio Web o a través del uso de plugins de navegadores.
Rastreo ocular.
Uso de subtítulos y/o transcripciones en elementos multimedia (audios, vídeos), lectores de pantalla, VoiceOver,ChromeVox lite, NVDA, JAWS.
Aumento de zoom/tamaño.
Realizar el sitio Web con un menú desplegable desde la izquierda.
Uso del elemento dialog: Si el usuario presiona Esc, cierra la ventana modal. Esto resulta muy útil porque permite que el usuario cierre la ventana modal sin la necesidad de buscar un botón para cerrar.