

# Informe con el análisis del primer test de accesibilidad

## Grupo N

Destacar que nuestra aplicación pasaba los correspondientes test de los navegadores Web. Subirlo a aplicaciones como GitHub Pages y pasarle el test de TAW fue imposible debido a que nuestra aplicación no tiene sentido sin su apartado de BackEnd.

En general, hemos implementado múltiples medidas para incrementar la accesibilidad a nivel de:

### Escritura

Títulos de página únicos: de cara la navegación cada página muestra estará identificada con un pequeño título acorde a la sección donde se encuentra el usuario.

Se han usado títulos y subtítulos, ej:home, para la agrupación de pequeñas secciones dentro de cada página.

Los enlaces que se muestran en nuestra página, ej:créditos, muestran claramente a donde te redirigen y su funcionalidad de cara a que el usuario sepa en todo momento por qué camino va.

Todas las imágenes poseen su etiqueta 'alt' la cual describe de manera resumida de qué trata la imagen, muy útil para las personas con vista reducida o ceguera.

Se dan instrucciones claras al usuario en todo momento, ej:login, para que el usuario sepa que ha de insertar en cada campo rellenable, y que sepa con qué formato.

El contenido es claro y conciso, todas nuestras páginas contienen texto reducido el cual específico y resumen su información de manera que el usuario pueda captarla de la manera más rápida y eficaz posible.

### Diseño

Las imágenes, los títulos y los textos cumplen la diferencia de contraste mínima con el fondo de manera que el usuario no tenga problemas a la hora de distinguirlos.

El color no es la única manera de mostrar información en ninguna de las páginas. Pues de esta manera evitamos que el usuario con problemas de vista pueda perder parte de la información mostrada por pantalla.

Todos los elementos interactivos son perceptibles ya que cuando se les pasa el ratón por encima, se les preseleccionan, se activa un efecto hover, o también están identificados en forma de enlace: color del texto en azul + subrayado.

Las opciones de navegación están identificadas de manera clara y concisa. Esta se ha mostrado con el 'navbar' el menú donde el usuario puede navegar entre las diferentes páginas o con botones que hacen de enlaces dentro del contenido de algunas páginas.

En los formularios cada campo de texto a llenar está mostrado de manera clara, ej: añadir cuentas, con un pequeño texto el cual indica la función de cada uno.

Se le dan al usuario un feedback claro en cada acción que éste realiza de manera que en todo momento es consciente de sus acciones y de que está modificando. Esto se puede ver a lo largo de toda la página.

Se respeta el espacio asociado a cada sección de manera que cada sección dentro de cada página sea reconocible fácilmente y el usuario pueda hacer su modelo de una manera más fácil.

Existen controles para los elementos interactivos que empiezan de manera automática, ej: el carrousel de home, contienen una serie de botones e indicaciones para que el usuario sepa cómo manejar estos componentes.

Para los componentes que son dinámicos se hace una pausa mínima de 5 segundos, además estos se pueden parar y reiniciar al gusto del usuario. Además estos muestran información que a la hora de cambiar transitan de manera cómoda a la vista.

El contenido en todo momento es responsive de manera que el usuario puede acceder a nuestra página desde cualquier dispositivo.

## Developing

Cada textField tiene su label en cada formulario, para que este tenga un pequeño texto identificativo.

Todas las imágenes poseen su texto identificativo con el atributo alt.

Nuestra página web tiene identificado el idioma en el código HTML con el atributo 'lang'.

No se ha recurrido al uso del CAPTCHA ya que es molesto para el usuario común en nuestra aplicación web, y ya que un ordenador no podrá crear cuentas pues está capado.

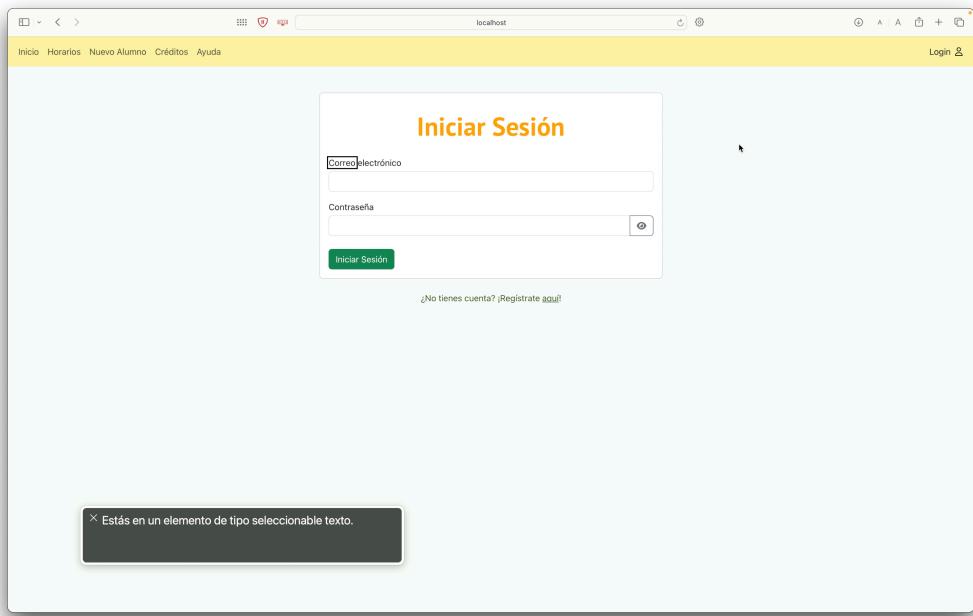
Las correspondientes estructuras html están creadas con su correspondiente etiqueta HTML de manera que los navegadores puedan ejecutarla sin problemas y que forme parte de un código legible.

El orden de lectura es reflejado por el orden del código ya que como se ha mencionado antes se ha desarrollado el código de manera que sea un código limpio y legible.

Se ha recurrido al uso del atributo tabindex para que a la hora de recorrer la página con el teclado se omitan o accedan a diferentes secciones de la página.

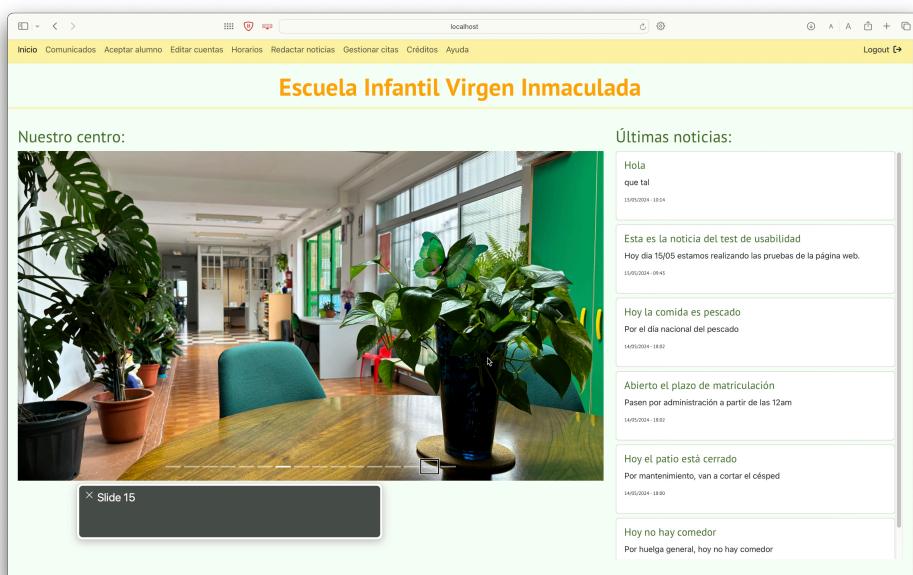
Se ha recurrido al uso de los atributos ARIA de manera que nuestro árbol de accesibilidad represente mejor la estructura de la página y que nuestra página sea accesible.

Cuando hicimos el test en el que vendamos los ojos a un compañero que nunca había usado la app para que la probara usando el lector de pantalla, nuestro compañero tuvo que enfrentarse a tareas como aceptar un alumno en el sistema, pedir cita o iniciar sesión. En ese proceso vimos que teníamos bastantes cosas por mejorar, como que había elementos tabulables no necesarios que ralentizaban el uso de la aplicación. En las siguientes etapas del desarrollo evitamos eso. Un ejemplo claro es el login, donde ahora no es tabulable los títulos de los campos.



**Aquí vemos un ejemplo de elemento innecesario tabulable que ralentizaba la navegación.**

Otro lugar en el que hemos mejorado la accesibilidad es en las presentaciones de fotografías. Ahora el lector de pantalla puede leer en qué consisten las imágenes que van rotando. Además, para ajustarnos al criterio 2.2.2 de la WCAG, hemos añadido un botón para poder parar la presentación.



**Carrousel antes**

# Escuela Infantil Virgen Inmaculada

Nuestro centro:



Pausar el pase de fotografías

Últimas noticias:

publica again  
sies

09/06/2024 - 13:47

Esta sí es  
publica

09/06/2024 - 13:47

## Carrousel ahora, mucho más accesible

En esta imagen también podemos comprobar cómo hemos cambiado algunos colores de la interfaz para ayudar a las personas con dificultades visuales.

Por otro lado, aunque no hemos podido pasarle a la aplicación el test de TAW, si hemos pasado otros test (con bastante éxito por cierto).

Inspector web: localhost — home

Accesibilidad

Diagnóstica de problemas de accesibilidad comunes que afectan a lectores de pantalla y otra tecnología de ayuda.

21 superada, 1 advertencia 95 %

- testMenuRoleForRequiredChildren
 

Asegúrate de que los elementos de las funciones "menu" y "menubar" tengan los elementos requeridos que son propiedad de acuerdo con WAI-ARIA.

  - domAttributes: ["role"] (1)
- testGridRoleForRequiredChildren
 

Asegúrate de que los elementos de la función "grid" tengan los elementos requeridos que son propiedad de acuerdo con WAI-ARIA.

  - domAttributes: ["role"] (1)
- testAriaLabelledBySpelling
 

Asegúrate de que "aria-labelledby" esté correctamente escrito.

  - domAttributes: ["aria-labelledby"] (1)
- testMultipleBanners
 

Asegúrate de usar un solo banner en la página.

  - domAttributes: ["role"] (1)
- testLinkLabels
 

Asegúrate de que los enlaces tengan etiquetas accesibles para la tecnología de ayuda.

  - domAttributes: ["aria-label", "aria-labelledby", "title"] (3)
- testRowGroupRoleForRequiredChildren
 

Asegúrate de que los elementos de la función "rowgroup" tengan los elementos requeridos que son propiedad de acuerdo con WAI-ARIA.

  - domAttributes: ["role"] (1)
- testTableRoleForRequiredChildren
 

Asegúrate de que los elementos de la función "table" tengan los elementos requeridos que son propiedad de acuerdo con WAI-ARIA.

  - domAttributes: ["role"] (1)

Resultados

- Ejecución 1
- Ejecución 2

Filtro

Automático: home

Auditoría de accesibilidad de Safari

No obstante, la mayor dificultad que sufrieron nuestros compañeros a la hora de probar nuestra aplicación fue el hecho de familiarizarse al lector de pantalla integrado con el ordenador, ya que no lo habían hecho nunca antes.

En definitiva, este test de accesibilidad nos ha servido mucho para poder aprender sobre el uso de esta herramienta y poder adaptar nuestra aplicación para una mayor eficiencia al usarla con el lector de pantalla.