

TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE TIJUANA

SUBDIRECCIÓN ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE:

Agosto - diciembre 2025

CARRERA:

Ingeniería en Sistemas Computacionales

MATERIA y SERIE:

TEMAS AVANZADOS DE DESARROLLO DE SOFTWARE, DSF-2104

TÍTULO ACTIVIDAD:

Accesibilidad Revisión y entrega final

UNIDAD A EVALUAR:

Unidad II y III

NOMBRE Y NÚMERO DE CONTROL DEL ALUMNO:

Herrera Barba Luis Pablo, 21211965.

Gómez Prieto Ricardo, 21211954,

Rodríguez Gallardo Alan Paul, C20212322

Gutiérrez Solís Maraly Guadalupe 18210482

Álvarez De Uña Iván Jared 20400695

Chavez Hernández Emmanuel Isaí 23211005

NOMBRE DEL MAESTRO (A):

Maribel Guerrero Luis

Introducción

La accesibilidad digital es un componente esencial en el desarrollo de aplicaciones modernas, ya que permite que cualquier usuario, sin importar sus capacidades visuales, auditivas, motoras o cognitivas, pueda interactuar correctamente con una interfaz tecnológica. Actualmente, muchas organizaciones y empresas exigen que sus sistemas cumplan con los lineamientos de accesibilidad establecidos por estándares como WCAG 2.1, no solo para garantizar inclusión, sino también para mejorar la experiencia general del usuario.

Durante este proyecto se analizaron y aplicaron diversas funciones orientadas a facilitar el uso del sistema por parte de públicos con diferentes necesidades. Entre estas se encuentran opciones visuales, controles adaptativos y herramientas cognitivas que fortalecen la usabilidad y aseguran que la interfaz pueda ser comprendida y manipulada con mayor facilidad.

La implementación de estas características busca sentar las bases para un sistema más profesional, alineado con prácticas reales utilizadas en la industria, donde la accesibilidad es un requisito cada vez más solicitado. Además, este trabajo permite comprender cómo pequeños ajustes en diseño y funcionalidad pueden generar un impacto significativo en la inclusión de usuarios con diversas limitaciones o condiciones especiales.

1. INFORME DE ACCESIBILIDAD (Accessibility Report)

1.1 Descripción general

Durante el desarrollo del sistema se integraron diversas funciones orientadas a mejorar la accesibilidad digital. Aunque el proyecto aún se encuentra en fase de mejora continua, ya se han incorporado herramientas clave que permiten que distintos tipos de usuarios interactúen con la interfaz de manera más cómoda y adaptable.

Estas funciones se implementan siguiendo principios recomendados en accesibilidad web y tomando en cuenta tanto necesidades reales de usuarios como prácticas que actualmente se solicitan en entornos profesionales y en empresas que requieren interfaces inclusivas.

1.2 Funciones de accesibilidad implementadas

Las funciones integradas hasta el momento son:

Visuales

- Ajuste de tamaño de texto (A-, A, A+).
- Cambio de tipografía.
- Modo de alto contraste.
- Inversión de colores.
- Zoom de pantalla integrado.
- Indicadores visuales para resaltar elementos interactivos.

Motoras / Físicas

- Puntero grande.
- Teclado en pantalla.
- Resaltado de enlaces para facilitar la selección.

Cognitivas

- Lectura en voz alta del contenido.
- Modo enfoque para reducir elementos distractores.
- Menús simplificados.

Estas funciones permiten que personas con capacidades distintas puedan utilizar la aplicación en mejores condiciones.

1.3 Justificación de las funciones seleccionadas

Las características incorporadas se eligieron considerando principalmente:

- **Personas con baja visión** (necesitan contraste, aumentó de texto y zoom).
- **Usuarios con daltonismo o sensibilidad a colores** (inversión y ajustes visuales).
- **Personas con dificultades motoras** (puntero grande y teclado en pantalla).
- **Usuarios con carga cognitiva reducida** (modo enfoque, lectura en voz alta).
- **Usuarios con poca experiencia digital o adultos mayores** (tipografías legibles y controles más visibles).

Estas funciones se escogieron porque representan mejoras aplicables de inmediato y están alineadas con prácticas actuales de accesibilidad utilizadas en aplicaciones profesionales.

1.4 Métodos de pruebas implementadas

Uso de Extensión web Wave

WAVE (Web Accessibility Evaluation Tool), desarrollado por WebAIM, usa una combinación de análisis automatizado y revisión asistida manualmente. Su metodología se basa en:

1. Análisis automatizado basado en reglas (rule-based automated testing)

WAVE ejecuta una serie de **reglas heurísticas** para detectar automáticamente errores y alertas relacionados con WCAG 2.x.

Ejemplos:

- falta de texto alternativo
- encabezados mal estructurados
- bajo contraste de color
- formularios sin etiquetas

Esto se parece a una auditoría automática, pero **no evalúa la accesibilidad completa**, porque muchas cosas requieren juicio humano.

2. Presentación visual de elementos para revisión manual

WAVE destaca visualmente en la página diferentes íconos y estructuras para que el evaluador:

- valide si el contenido cumple realmente con WCAG
- interprete elementos que el automático no puede determinar
- Es decir, WAVE **no hace una auditoría manual**, pero facilita la revisión manual mostrando dónde mirar.

3. Evaluación contextual

WAVE analiza la estructura del DOM y el contexto de cada elemento para detectar patrones problemáticos, usando:

- Análisis semántico ligero
- Heurísticas de usabilidad y accesibilidad
- Evaluación de relaciones ARIA/HTML

Uso de GoogleDevTools

Google Lighthouse utiliza principalmente un método de pruebas automatizado basado en reglas (rule-based automated testing), ejecutado dentro del navegador Chrome. Para accesibilidad, Lighthouse usa un motor llamado axe-core (desarrollado porDeque Systems), que aplica reglas automáticas mapeadas a WCAG 2.x.

1. Auditoría automatizada basada en reglas (Rule-Based Automated Auditing)

Lighthouse ejecuta una batería de reglas predefinidas (basadas en axe-core) para evaluar:

- uso correcto de roles ARIA
- accesibilidad del formulario
- contraste de color
- etiquetas en botones
- estructura semántica
- navegación por teclado
- atributos obligatorios faltantes

Estas reglas son totalmente automáticas y no requieren intervención humana.

2. Simulación en tiempo de ejecución (Runtime Analysis)

Lighthouse ejecuta el sitio en un **contexto simulado**, evaluando:

- renderizado del DOM
- árbol de accesibilidad del navegador
- estados del UI
- presencia de elementos interactivos
- tiempos de carga y cambios en el DOM

Esto permite detectar problemas que no se identifican solo leyendo el código.

3. Análisis dinámico con interacción mínima

La herramienta no realiza interacciones profundas como un usuario real, pero sí:

- activa scripts
- carga recursos
- analiza eventos
- inspecciona nodos interactivos

1.5 Resultados obtenidos y mejoras pendientes

Anexo

- Reportes de Google LightHouse
- Reportes de Wave

Revisar Apartado 2 y 3 del documento

2. Matriz de Cumplimiento WCAG

Criterio WCAG	Nivel	Estado	Evidencia / Cómo se cumple
1.1.1 Texto alternativo	A	Cumplido	El logotipo del TecNM incluye atributo ALT.
1.3.1 Estructura y relaciones	A	Cumplido	Se utilizan encabezados H1 y H2 en página de inicio.
1.4.3 Contraste (mínimo)	AA	Parcial	Se agregó modo de alto contraste, aunque algunos botones necesitan ajustes.
1.4.4 Tamaño del texto	AA	Cumplido	El sistema cuenta con control A-, A, A+.
1.4.11 Contraste de elementos no textuales	AA	Parcial	El puntero grande y menú resaltado funcionan, pero faltan ajustes en algunas gráficas.

1.4.12 Espaciado de texto	AA	Cumplido	Los párrafos mantienen espacio adecuado incluso con texto aumentado.
2.1.1 Teclado	A	Cumplido	Navegación por tab en menú y botones principales.
2.1.2 Sin trampas de teclado	A	Cumplido	Ningún componente atrapa el enfoque.
2.4.7 Foco visible	AA	Parcial	Algunos botones no muestran claramente el foco.
3.2.3 Navegación coherente	AA	Cumplido	El menú lateral permanece consistente en todas las vistas.
3.3.2 Etiquetas e instrucciones	A	Cumplido	Formularios contienen labels asociados.
4.1.2 Nombre, rol, valor	A	Cumplido	Botones y componentes del menú tienen roles y texto visible.

3. Registro de Pruebas de Accesibilidad (Testing Log)

Registro de Pruebas de Accesibilidad #1 (Testing Log)

Aplicación evaluada: Página de inicio de sesión (localhost)

Fecha de prueba: 30 de noviembre de 2025

Herramienta principal: Lighthouse 12.8.2 (incluido en Chrome DevTools)

Entorno de prueba: Emulado en escritorio, Chromium 142.0.0.0

Casos de prueba aplicados

1. Evaluación automatizada de accesibilidad mediante Lighthouse.
2. Verificación de estructura semántica (encabezados, landmarks, etiquetas).
3. Validación de formularios (etiquetas, roles, navegación por teclado).
4. Comprobación de contraste de color y tamaño de objetivos táctiles.
5. Revisión de soporte para lectores de pantalla (atributos lang, alt, ARIA).
6. Validación de navegación por teclado (orden de tabulación, trampas de foco).

Herramientas utilizadas

- **Lighthouse** (integrado en Chrome DevTools) – Evaluación automatizada.
- **Chrome DevTools** – Para revisión de elementos, consola y panel de Issues.
- **Validación manual** de elementos críticos (formularios, encabezados, landmarks).

Problemas encontrados y soluciones propuestas

1. Falta del atributo lang en <html>

- **Problema:** El elemento <html> no tiene atributo lang, lo que afecta la pronunciación de lectores de pantalla.
- **Solución:** Agregar lang="es" (o el idioma correspondiente) al elemento <html>.

2. Campos de formulario sin etiquetas asociadas

- **Problema:** Los inputs (clase form-control) no tienen etiquetas <label> asociadas.
- **Solución:** Agregar etiquetas <label> con atributo for que apunte al id del input correspondiente.

3. Estructura de encabezados no secuencial

- **Problema:** Aunque no se especifica un fallo directo, se recomienda validar que los encabezados (h1 a h6) sigan un orden lógico.
- **Solución:** Asegurar que la página use h1 para el título principal y que no se salte niveles.

4. Falta de landmarks ARIA

- **Problema:** No se detectaron landmarks como <main>, <nav>, <aside>, lo que dificulta la navegación para usuarios de lectores de pantalla.
- **Solución:** Implementar landmarks ARIA para identificar regiones principales de la página.

5. Contraste de color no evaluado automáticamente

- **Problema:** Lighthouse no reporta fallos, pero se recomienda verificación manual.
- **Solución:** Usar herramientas como **WAVE** o **Color Contrast Analyzer** para validar contraste en textos y botones.

6. Prevención de caché de navegación (bfcache)

- **Problema:** La página no puede usar el back/forward cache por varias razones (unload handlers, WebSockets, no-store).
- **Solución:** Revisar y eliminar manejadores unload, considerar alternativas a WebSockets si es posible, ajustar cabeceras de caché.

Pruebas superadas (según Lighthouse)

- El documento tiene el elemento <title>.
- Los botones tienen nombres accesibles.
- No se usa user-scalable="no" en el viewport.
- Colores de fondo y primer plano tienen suficiente contraste.
- Los objetivos táctiles tienen tamaño y espacio adecuados.
- Los encabezados siguen un orden descendente.

Recomendaciones generales

- Agregar un lang declarado en <html>.
- Etiquetar todos los controles del formulario.
- Implementar landmarks ARIA (role="main", role="navigation", etc.).
- Realizar pruebas con lectores de pantalla como NVDA o VoiceOver.
- Validar navegación por teclado (tab order, focus visible).
- Revisar y optimizar recursos de terceros (extensiones de Chrome) que afectan el rendimiento y la accesibilidad.

Estado general de accesibilidad:

Necesita mejoras – Se detectaron fallos críticos (sin lang, formularios sin etiquetas) que deben corregirse para cumplir con WCAG 2.1.

Próximos pasos:

- Corregir fallos críticos.
- Realizar prueba manual con NVDA.
- Volver a ejecutar Lighthouse tras las correcciones.

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad #2 (Testing Log)

Aplicación evaluada: Página de Inicio (`/home/index`)

Fecha de prueba: 30 de noviembre de 2025

Herramienta principal: Lighthouse 12.8.2 (incluido en Chrome DevTools)

Entorno de prueba: Emulado en escritorio, Chromium 142.0.0.0

Casos de prueba aplicados

1. Evaluación automatizada de accesibilidad mediante Lighthouse.
2. Verificación de estructura semántica (encabezados, landmarks, etiquetas).
3. Validación de contraste de color y tamaño de objetivos táctiles.
4. Revisión de soporte para lectores de pantalla (atributos `lang`, `alt`, ARIA).
5. Validación de navegación por teclado (orden de tabulación, trampas de foco).
6. Verificación de etiquetas de formulario (inputs, botones, selects).
7. Validación de imágenes (atributos `alt`, redundancia de texto).

Herramientas utilizadas

- Lighthouse (integrado en Chrome DevTools) – Evaluación automatizada.
- Chrome DevTools – Para revisión de elementos, consola y panel de Issues.
- Validación manual de elementos críticos (encabezados, landmarks, formularios).

Problemas encontrados y soluciones propuestas

1. Falta del atributo `lang` en ``

- Problema: El elemento `` no tiene atributo `lang`, lo que afecta la pronunciación de lectores de pantalla.
- Solución: Agregar `lang="es"` (o el idioma correspondiente) al elemento ``.

2. Controles interactivos sin indicación de estado o propósito

- Problema: No se verificó si los elementos interactivos (botones, enlaces) indican claramente su estado y propósito.
- Solución: Asegurar que botones y enlaces tengan `:hover`, `:focus` y `:active` visibles, y usar `aria-label` si es necesario.

3. Orden de tabulación lógico no verificado

- Problema: No se confirmó que el orden de tabulación siga el orden visual.
- Solución: Realizar prueba manual de navegación por teclado (`Tab`, `Shift+Tab`).

4. Landmarks HTML5 no utilizados

- Problema: No se detectó uso de elementos como `<main>`, `<nav>`, `<aside>`, lo que dificulta la navegación para usuarios de lectores de pantalla.
- Solución: Implementar landmarks ARIA (`role="main"`, `role="navigation"`, etc.).

5. Contenido fuera de pantalla no oculto para tecnología asistiva

- Problema: No se verificó si el contenido fuera de pantalla está correctamente oculto con `display: none` o `aria-hidden="true"`.
- Solución: Asegurar que el contenido offscreen no sea accesible por lectores de pantalla si no es necesario.

6. Controles personalizados sin etiquetas o roles ARIA

- Problema: No se confirmó que los controles personalizados tengan etiquetas accesibles (`aria-label`, `aria-labelledby`) y roles ARIA adecuados.
- Solución: Asignar roles ARIA apropiados (`role="button"`, `role="slider"`) y etiquetas accesibles.

7. Imágenes sin atributo `alt` o con texto redundante

- Problema: Aunque Lighthouse indica que las imágenes tienen `alt`, no valida si el texto es redundante con el contenido adyacente.
- Solución: Revisar manualmente que las imágenes decorativas tengan `alt=""` y las informativas tengan descripciones concisas.

Pruebas superadas (según Lighthouse)

- Atributos ARIA coinciden con sus roles.
- No hay `aria-hidden="true"` en el `<body>`.
- Atributos ARIA tienen valores válidos y nombres correctos.
- Los botones tienen nombres accesibles.
- Las Imágenes tienen atributos `alt`.
- No se usa `user-scalable="no"` en el viewport.
- Colores de fondo y primer plano tienen suficiente contraste.
- El documento tiene elementos `<title>`.
- Los enlaces tienen nombres discernibles.
- Las listas están bien estructuradas (``, ``, ``).
- Los objetivos táctiles tienen tamaño y espacio adecuados.
- Encabezados siguen un orden descendente.
- Las Imágenes no tienen texto `alt` redundante.

Recomendaciones generales

1. Agregar un `lang` declarado** en ``.
2. Implementar landmarks ARIA** (`role="main"`, `role="navigation"`, etc.).
3. Realizar prueba manual de navegación por teclado.
4. Validar que los controles personalizados tengan etiquetas y roles ARIA.
5. Revisar que el contenido offscreen esté correctamente oculto.
6. Asegurar que las imágenes tengan `alt` apropiados y no redundantes.
7. Verificar que los estados de interacción (hover, focus) sean visibles.

Estado general de accesibilidad:

Necesita mejoras manuales – Aunque Lighthouse reporta varios éxitos, hay áreas que requieren verificación manual (landmarks, orden de tabulación, controles personalizados).

Próximos pasos:

- Corregir la falta de `lang` en ``.
- Realizar prueba manual con NVDA o VoiceOver.
- Implementar landmarks ARIA.
- Validar navegación por teclado.
- Revisar etiquetas `alt` de imágenes

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad # 3 (Testing Log)

Aplicación evaluada: Página de Registro de Alumnos (/Alumnos/AgregarAlumno)

Fecha de prueba: 30 de noviembre de 2025

Herramienta principal: Lighthouse 12.8.2 (incluido en Chrome DevTools)

Entorno de prueba: Emulado en escritorio, Chromium 142.0.0.0

Casos de prueba aplicados

- Evaluación automatizada de accesibilidad mediante Lighthouse.
- Verificación de estructura semántica (encabezados, landmarks, etiquetas).
- Validación de contraste de color y tamaño de objetivos táctiles.
- Revisión de soporte para lectores de pantalla (atributos lang, alt, ARIA).
- Validación de navegación por teclado (orden de tabulación, trampas de foco).
- Verificación de etiquetas de formulario (inputs, botones, selects).
- Validación de imágenes (atributos alt, redundancia de texto).
- Validación de listas (estructura adecuada de , ,).

Herramientas utilizadas

- Lighthouse (integrado en Chrome DevTools) – Evaluación automatizada.
- Chrome DevTools – Para revisión de elementos, consola y panel de Issues.
- Validación manual de elementos críticos (formularios, landmarks, navegación por teclado).

Problemas encontrados y soluciones propuestas

1. Falta del atributo lang en <html>

- Problema: El elemento <html> no tiene atributo lang, lo que afecta la pronunciación de lectores de pantalla.
- Solución: Agregar lang="es" (o el idioma correspondiente) al elemento <html>.

2. Controles interactivos sin indicación de estado o propósito (verificación manual necesaria)

- Problema: No se verificó si los elementos interactivos (botones, enlaces) indican claramente su estado y propósito.
- Solución: Asegurar que botones y enlaces tengan: hover, focus y: active visibles, y usar aria-label si es necesario.

3. Orden de tabulación lógico no verificado (manual)

- Problema: No se confirmó que el orden de tabulación siga el orden visual.
- Solución: Realizar prueba manual de navegación por teclado (Tab, Shift+Tab).

4. Landmarks HTML5 no utilizados (manual)

- Problema: No se detectó uso de elementos como `<main>`, `<nav>`, `<aside>`, lo que dificulta la navegación para usuarios de lectores de pantalla.
- Solución: Implementar landmarks ARIA (`role="main"`, `role="navigation"`, etc.).

5. Contenido fuera de pantalla no oculto para tecnología asistiva (manual)

- Problema: No se verificó si el contenido fuera de pantalla está correctamente oculto con `display: none` o `aria-hidden="true"`.
- Solución: Asegurar que el contenido offscreen no sea accesible por lectores de pantalla si no es necesario.

6. Controles personalizados sin etiquetas o roles ARIA (manual)

- Problema: No se confirmó que los controles personalizados tengan etiquetas accesibles (`aria-label`, `aria-labelledby`) y roles ARIA adecuados.
- Solución: Asignar roles ARIA apropiados (`role="button"`, `role="slider"`) y etiquetas accesibles.

7. Imágenes sin atributo alt o con texto redundante (manual)

- Problema: Aunque Lighthouse indica que las imágenes tienen alt, no valida si el texto es redundante con el contenido adyacente.
- Solución: Revisar manualmente que las imágenes decorativas tengan `alt=""` y las informativas tengan descripciones concisas.

8. Errores de JavaScript que afectan funcionalidad

- Problema: Se detectaron errores de consola relacionados con jQuery no definido (`ReferenceError: $ is not defined`).
- Solución: Asegurar que jQuery esté cargado antes de los scripts que lo dependan (como `jquery.validate.min.js`).

Pruebas superadas (según Lighthouse)

- Atributos ARIA coinciden con sus roles.
- No hay `aria-hidden="true"` en el `<body>`.
- Atributos ARIA tienen valores válidos y nombres correctos.
- Los botones tienen nombres accesibles.
- Las imágenes tienen atributos alt.
- No se usa `user-scalable="no"` en el viewport.
- Colores de fondo y primer plano tienen suficiente contraste.
- El documento tiene el elemento `<title>`.
- Elementos de formulario tienen etiquetas asociadas (importante para accesibilidad de formularios).

- Los enlaces tienen nombres discernibles.
- Las listas están bien estructuradas (, ,).
- Elementos <select> tienen etiquetas asociadas.
- Los objetivos táctiles tienen tamaño y espacio adecuados.
- Encabezados siguen un orden descendente.
- Las Imágenes no tienen texto alt redundante.

Recomendaciones generales

- Agregar un lang declarado en <html>.
- Implementar landmarks ARIA (role="main", role="navigation", etc.).
- Realizar prueba manual de navegación por teclado.
- Validar que los controles personalizados tengan etiquetas y roles ARIA.
- Revisar que el contenido offscreen esté correctamente oculto.
- Asegurar que las imágenes tengan alt apropiados y no redundantes.
- Corregir errores de JavaScript (jQuery no definido).
- Verificar que los estados de interacción (hover, focus) sean visibles.

Estado general de accesibilidad:

Bueno (95/100) – La página cumple con varios criterios importantes, especialmente en formularios etiquetados, contraste y estructura. Sin embargo, requiere correcciones manuales en landmarks, lenguaje y navegación por teclado.

Próximos pasos:

- Corregir la falta de lang en <html>.
- Implementar landmarks ARIA.
- Validar navegación por teclado manualmente.
- Corregir errores de jQuery en la consola.
- Realizar prueba con lector de pantalla (NVDA/VoiceOver).

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad # 4 (Testing Log)

Aplicación evaluada: Página de Búsqueda de Alumnos (/Alumnos/BuscarAlumnos)

Fecha de prueba: 30 de noviembre de 2025

Herramienta principal: Lighthouse 12.8.2 (incluido en Chrome DevTools)

Entorno de prueba: Emulado en escritorio, Chromium 142.0.0.0

Casos de prueba aplicados

- Evaluación automatizada de accesibilidad mediante Lighthouse.
- Verificación de estructura semántica (encabezados, landmarks, etiquetas).
- Validación de contraste de color y tamaño de objetivos táctiles.
- Revisión de soporte para lectores de pantalla (atributos lang, alt, ARIA).
- Validación de navegación por teclado (orden de tabulación, trampas de foco).
- Verificación de botones sin nombres accesibles (problema crítico detectado).
- Validación de elementos de formulario (selects sin etiquetas).
- Validación de estructura de encabezados (orden secuencial).
- Validación de imágenes (atributos alt).

Herramientas utilizadas

- Lighthouse (integrado en Chrome DevTools) – Evaluación automatizada.
- Chrome DevTools – Para revisión de elementos, consola y panel de Issues.
- Validación manual de elementos críticos (botones, selects, encabezados).

Problemas encontrados y soluciones propuestas

1. Botones sin nombres accesibles (fallo crítico y repetitivo)

- Problema: Múltiples botones con clase btn btn-sm btn-link p-0 no tienen nombre accesible (sin aria-label, aria-labelledby o texto visible).
- Solución: Agregar aria-label descriptivo a cada botón (ej. aria-label="Editar alumno X"), o incluir texto visible dentro del botón.

2. Elementos <select> sin etiquetas asociadas

- Problema: Dos elementos <select class="form-select"> no tienen etiquetas <label> asociadas.
- Solución: Agregar etiquetas <label> con atributo for que apunte al id del select correspondiente.

3. Falta del atributo lang en <html>

- Problema: El elemento <html> no tiene atributo lang, lo que afecta la pronunciación de lectores de pantalla.
- Solución: Agregar lang="es" (o el idioma correspondiente) al elemento <html>.

4. Encabezados no en orden secuencial descendente

- Problema: Un elemento `<h4>` no sigue un orden lógico de encabezados (posible salto de niveles).
- Solución: Revisar la estructura de encabezados ($h1 \rightarrow h2 \rightarrow h3 \rightarrow h4$) y ajustar niveles según la jerarquía del contenido.

5. Landmarks HTML5 no utilizados (verificación manual necesaria)

- Problema: No se detectó uso de elementos como `<main>`, `<nav>`, `<aside>`, lo que dificulta la navegación para usuarios de lectores de pantalla.
- Solución: Implementar landmarks ARIA (`role="main"`, `role="navigation"`, etc.).

6. Controles interactivos sin indicación de estado o propósito (manual)

- Problema: No se verificó si los elementos interactivos indican claramente su estado y propósito.
- Solución: Asegurar que botones y enlaces tengan: `hover`, `focus` y `active` visibles.

7. Orden de tabulación lógico no verificado (manual)

- Problema: No se confirmó que el orden de tabulación siga el orden visual.
- Solución: Realizar prueba manual de navegación por teclado (`Tab`, `Shift+Tab`).

8. DOM excesivamente grande (978 elementos)

- Problema: El DOM tiene 978 elementos, lo que puede afectar el rendimiento y la navegación con tecnología asistiva.
- Solución: Simplificar la estructura HTML, evitar anidaciones profundas y usar paginación o carga diferida si es una tabla larga.

Pruebas superadas (según Lighthouse)

- Atributos ARIA coinciden con sus roles.
- No hay `aria-hidden="true"` en el `<body>`.
- Atributos ARIA tienen valores válidos y nombres correctos.
- Las imágenes tienen atributos `alt`.
- No se usa `user-scalable="no"` en el `viewport`.
- Colores de fondo y primer plano tienen suficiente contraste.
- El documento tiene el elemento `<title>`.
- Elementos de formulario (excepto `selects`) tienen etiquetas asociadas.
- Los enlaces tienen nombres discernibles.
- Las listas están bien estructuradas.
- Celdas de tabla con atributo `headers` referencian correctamente.
- Las Imágenes no tienen texto `alt` redundante.

- Tablas tienen summary y <caption> diferenciados.
- Los objetivos táctiles tienen tamaño y espacio adecuados.

Recomendaciones generales

- Corregir botones sin nombre accesible (prioridad alta).
- Agregar etiquetas a elementos <select>.
- Agregar un lang declarado en <html>.
- Revisar y corregir la jerarquía de encabezados.
- Implementar landmarks ARIA (role="main", role="navigation", etc.).
- Simplificar el DOM (reducir número de elementos y profundidad).
- Realizar prueba manual de navegación por teclado.
- Validar que los controles personalizados tengan etiquetas y roles ARIA.
- Revisar que el contenido offscreen esté correctamente oculto.
- Asegurar que las imágenes tengan alt apropiados.

Estado general de accesibilidad:

Necesita mejoras críticas (84/100) – La página tiene problemas graves de accesibilidad en botones y selects, además de falta de lenguaje declarado y estructura de encabezados. Debe corregirse antes de considerarse accesible.

Próximos pasos:

- Corregir botones sin nombre accesible (urgente).
- Etiquetar elementos <select>.
- Agregar lang="es" en <html>.
- Revisar jerarquía de encabezados.
- Realizar prueba con lector de pantalla (NVDA/VoiceOver).
- Optimizar DOM para mejorar rendimiento y accesibilidad.

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad # 5 (Testing Log)

Fecha de la prueba: 30 de noviembre de 2025

Página evaluada: <https://localhost:7179/home/Exportarlista>

Herramienta principal: Lighthouse (Chrome DevTools)

Versión de Lighthouse: 12.8.2

Emulación: Desktop con throttling personalizado

Casos de Prueba Aplicados

- Evaluación automática con Lighthouse
- Auditoría de Accesibilidad (basada en WCAG y mejores prácticas).
- Auditoría de SEO.
- Auditoría de Mejores Prácticas.
- Revisión manual complementaria
- Validación de estructura HTML.
- Verificación de etiquetas semánticas y atributos ARIA.
- Comprobación de navegación por teclado y foco.

Herramientas Utilizadas

Herramienta	Tipo	Uso en esta prueba
Lighthouse (Chrome DevTools)	Automatizada	Evaluación integral de accesibilidad, SEO y mejores prácticas.
Chrome DevTools	Manual	Inspección de elementos, consola, panel de issues.
(No se usó WAVE, NVDA ni VoiceOver en este log, pero se recomiendan para pruebas futuras)		

Problemas Encontrados y Soluciones a Implementar

Problemas Críticos (Deben corregirse urgentemente)

Problema	Descripción	Solución Propuesta
Imágenes sin atributo alt	Varias etiquetas no tienen texto alternativo.	Agregar alt descriptivo para imágenes informativas; alt="" para imágenes decorativas.
Documento sin idioma definido	La etiqueta <html> no tiene atributo lang.	Agregar lang="es" (o el idioma correspondiente).
Meta descripción faltante	No hay etiqueta <meta name="description">.	Agregar una descripción concisa y relevante para SEO y accesibilidad.

Problemas Moderados (Recomendados para corregir)

Problema	Descripción	Solución Propuesta
Auditorías de SEO básicas fallidas	Faltan meta description y atributos alt.	Implementar las correcciones anteriores.
Problemas de seguridad en headers	No se encontraron CSP, HSTS, COOP, XFO.	Configurar headers de seguridad en el servidor.
Uso de APIs deprecadas	Se detectó un event listener unload deprecado.	Reemplazar con beforeunload o pagehide.

Problemas Leves o de Mejora Continua

Problema	Descripción	Solución Propuesta
CSS y JS sin minificar	Se identificaron archivos CSS y JS sin minificar.	Minificar recursos para reducir tamaño y mejorar carga.

Recursos sin caché	Muchos archivos no tienen TTL de caché configurado.	Configurar cabeceras Cache-Control en el servidor.
Fuentes sin font-display: swap	Las fuentes web pueden causar desplazamientos de diseño.	Agregar font-display: swap en @font-face.

Resumen de Puntuaciones Lighthouse

Categoría	Puntuación	Estado
Rendimiento	88/100	Aceptable
Accesibilidad	74/100	Necesita mejora
Mejores Prácticas	74/100	Necesita mejora
SEO	82/100	Aceptable

Recomendaciones Generales

Accesibilidad:

- Agregar lang al <html>.
- Proporcionar alt a todas las imágenes.
- Revisar navegación por teclado y contraste de colores.

Seguridad:

- Implementar CSP, HSTS, COOP y X-Frame-Options.
- Rendimiento:
- Minificar CSS y JS.
- Configurar caché para recursos estáticos.
- Optimizar imágenes (usar WebP/AVIF).

SEO:

- Agregar meta descripción.
- Asegurar que los títulos sean únicos y descriptivos.

Próximos pasos sugeridos:

- Realizar pruebas con lectores de pantalla (NVDA, VoiceOver).
- Validar con WAVE o axe DevTools.
- Revisar en dispositivos móviles y con diferentes tamaños de pantalla.
- Corregir los issues de seguridad y accesibilidad priorizados.

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad (Testing Log) #6 - Página de Gráfico de Dispersión

Fecha de la prueba: 30 de noviembre de 2025

Página evaluada: https://localhost:7179/Graficas/G_Dispersion

Herramienta principal: Lighthouse (Chrome DevTools)

Versión de Lighthouse: 12.8.2

Emulación: Desktop con throttling personalizado

Casos de Prueba Aplicados

- Evaluación automática con Lighthouse
- Auditoría de Accesibilidad (basada en WCAG y mejores prácticas).
- Auditoría de SEO.
- Auditoría de Mejores Prácticas.
- Revisión manual complementaria
- Validación de estructura HTML.
- Verificación de formularios y elementos de entrada.
- Comprobación de navegación por teclado y estructura de encabezados.

Herramientas Utilizadas

Herramienta	Tipo	Uso en esta prueba
Lighthouse (Chrome DevTools)	Automatizada	Evaluación integral de accesibilidad, SEO y mejores prácticas.
Chrome DevTools	Manual	Inspección de elementos, consola, panel de issues.
(No se usó WAVE, NVDA ni VoiceOver en este log, pero se recomiendan para pruebas futuras)		

Problemas Encontrados y Soluciones a Implementar

Problemas Críticos (Deben corregirse urgentemente)

Problema	Descripción	Solución Propuesta
Documento sin idioma definido	La etiqueta <html> no tiene atributo lang.	Agregar lang="es" (o el idioma correspondiente).
Select sin etiquetas asociadas	Tres elementos <select> (#carreraSelect, #materiaSelect, #unidadSelect) no tienen etiquetas <label>.	Agregar etiquetas <label> descriptivas para cada select, vinculadas con for o usando aria-label/aria-labelledby.
Meta descripción faltante	No hay etiqueta <meta name="description">.	Agregar una descripción concisa y relevante para SEO y accesibilidad.

Problemas Moderados (Recomendados para corregir)

Problema	Descripción	Solución Propuesta
Estructura de encabezados incorrecta	Se detectó un <h5> que no sigue un orden secuencial descendente en la jerarquía de encabezados.	Revisar la estructura de encabezados (h1 → h6) y asegurar que no se salten niveles.
Problemas de seguridad en headers	No se encontraron CSP, HSTS, COOP, XFO, Trusted Types.	Configurar headers de seguridad en el servidor.
Uso de APIs deprecadas	Se detectó un event listener unload deprecado.	Reemplazar con beforeunload o pagehide.

Problemas Leves o de Mejora Continua

Problema	Descripción	Solución Propuesta
CSS y JS sin minificar	Se identificaron archivos CSS y JS sin minificar.	Minificar recursos para reducir tamaño y mejorar carga.
Recursos sin caché	Muchos archivos no tienen TTL de caché configurado.	Configurar cabeceras Cache-Control en el servidor.
Fuentes sin font-display: swap	Las fuentes web pueden causar desplazamientos de diseño.	Agregar font-display: swap en @font-face.
CSS y JS no utilizado	Se identificaron 182 KiB de CSS y 498 KiB de JS no utilizados.	Eliminar código no utilizado o cargarlo de forma diferida.

Resumen de Puntuaciones Lighthouse

Categoría	Puntuación	Estado
Rendimiento	99/100	Excelente
Accesibilidad	89/100	Bueno, pero con áreas de mejora
Mejores Prácticas	74/100	Necesita mejora
SEO	91/100	Excelente

Recomendaciones Generales

Accesibilidad:

- Agregar lang al <html>.
- Proporcionar etiquetas <label> para todos los elementos <select>.
- Revisar y corregir la estructura jerárquica de encabezados.
- Realizar pruebas con lectores de pantalla en los controles de formulario.

Seguridad:

- Implementar CSP, HSTS, COOP, X-Frame-Options y Trusted Types.

Rendimiento:

- Minificar CSS y JS.
- Configurar caché para recursos estáticos.
- Eliminar código CSS/JS no utilizado.

SEO:

- Agregar meta descripción.

Aspectos Positivos Destacados

- Excelente rendimiento (99/100) con métricas de Core Web Vitals muy buenas.
- Imágenes con atributos alt ya implementados (punto positivo frente a otros informes).
- Buenas prácticas de formularios en otros aspectos (pasó auditorías de etiquetas de formulario).
- Estructura de listas y tablas correcta según auditorías pasadas.

Próximos pasos sugeridos:

- Realizar pruebas con lectores de pantalla (NVDA, VoiceOver) en los controles de selección.
- Validar con WAVE o axe DevTools para problemas de contraste y navegación por teclado.
- Revisar en dispositivos móviles la disposición de los controles de formulario.
- Corregir los issues de seguridad y accesibilidad priorizados.

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad (Testing Log) #7 - Página de Gráfico de Pastel

Fecha de la prueba: 30 de noviembre de 2025

Página evaluada: https://localhost:7179/Graficas/G_Pastel

Herramienta principal: Lighthouse (Chrome DevTools)

Versión de Lighthouse: 12.8.2

Emulación: Desktop con throttling personalizado

Casos de Prueba Aplicados

- Evaluación automática con Lighthouse
- Auditoría de Accesibilidad (basada en WCAG y mejores prácticas).
- Auditoría de SEO.
- Auditoría de Mejores Prácticas.
- Revisión manual complementaria
- Validación de estructura HTML.
- Verificación de formularios y elementos de entrada.
- Comprobación de contraste de colores y estructura de encabezados.

Herramientas Utilizadas

Herramienta	Tipo	Uso en esta prueba
Lighthouse (Chrome DevTools)	Automatizada	Evaluación integral de accesibilidad, SEO y mejores prácticas.
Chrome DevTools	Manual	Inspección de elementos, consola, panel de issues.
(No se usó WAVE, NVDA ni VoiceOver en este log, pero se recomiendan para pruebas futuras)		

Problemas Encontrados y Soluciones a Implementar

Problemas Críticos (Deben corregirse urgentemente)

Problema	Descripción	Solución Propuesta
Contraste insuficiente	Elementos <code>div#resumen.bloque-interpretacion</code> y <code>td</code> no cumplen con la relación de contraste mínima (4.5:1).	Ajustar colores de texto/fondo para cumplir con WCAG AA (4.5:1) o AAA (7:1).
Documento sin idioma definido	La etiqueta <code><html></code> no tiene atributo <code>lang</code> .	Agregar <code>lang="es"</code> (o el idioma correspondiente).
Select sin etiquetas asociadas	Tres elementos <code><select></code> (<code>#carreraSelect</code> , <code>#materiaSelect</code> , <code>#unidadSelect</code>) no tienen etiquetas <code><label></code> .	Agregar etiquetas <code><label></code> descriptivas para cada select, vinculadas con <code>for</code> o usando <code>aria-label/aria-labelledby</code> .
Meta descripción faltante	No hay etiqueta <code><meta name="description"></code> .	Agregar una descripción concisa y relevante para SEO y accesibilidad.

Problemas Moderados (Recomendados para corregir)

Problema	Descripción	Solución Propuesta
Estructura de encabezados incorrecta	Se detectó un <code><h5></code> que no sigue un orden secuencial descendente en la jerarquía de encabezados.	Revisar la estructura de encabezados (<code>h1</code> → <code>h6</code>) y asegurar que no se salten niveles.

Problemas de seguridad en headers	No se encontraron CSP, HSTS, COOP, XFO, Trusted Types.	Configurar headers de seguridad en el servidor.
Uso de APIs deprecadas	Se detectó un event listener unload deprecado.	Reemplazar con beforeunload o pagehide.

Problemas Leves o de Mejora Continua

Problema	Descripción	Solución Propuesta
CSS y JS sin minificar	Se identificaron archivos CSS y JS sin minificar.	Minificar recursos para reducir tamaño y mejorar carga.
Recursos sin caché	Muchos archivos no tienen TTL de caché configurado.	Configurar cabeceras Cache-Control en el servidor.
CSS y JS no utilizado	Se identificaron 182 KiB de CSS y 506 KiB de JS no utilizados.	Eliminar código no utilizado o cargarlo de forma diferida.
Imagen sin optimizar	/img/Logo.png podría optimizarse con WebP/AVIF.	Convertir a formato moderno y comprimir.

Resumen de Puntuaciones Lighthouse

Categoría	Puntuación	Estado
Rendimiento	96/100	Excelente
Accesibilidad	84/100	Aceptable, con áreas críticas
Mejores Prácticas	74/100	Necesita mejora
SEO	91/100	Excelente

Recomendaciones Generales

Accesibilidad:

Prioridad 1: Corregir problemas de contraste en #resumen y celdas de tabla.

- Agregar lang al <html>.
- Proporcionar etiquetas <label> para todos los elementos <select>.
- Revisar y corregir la estructura jerárquica de encabezados.

Seguridad:

- Implementar CSP, HSTS, COOP, X-Frame-Options y Trusted Types.

Rendimiento:

- Minificar CSS y JS.
- Configurar caché para recursos estáticos.
- Eliminar código CSS/JS no utilizado.

SEO:

- Agregar meta descripción.

Aspectos Positivos Destacados

- Excelente rendimiento (96/100) con métricas de Core Web Vitals muy buenas.
- Imágenes con atributos alt ya implementados.
- Buena estructura de listas y tablas según auditorías pasadas.
- Total Blocking Time (TBT) bajo (110 ms).
- Cumulative Layout Shift (CLS) muy bajo (0.01).

Problemas Recurrentes Detectados en Múltiples Páginas

- Falta de atributo lang en <html> (presente en todas las páginas analizadas).
- Selects sin etiquetas (presente en páginas de gráficos).
- Headers de seguridad ausentes (presente en todas).
- Meta descripción faltante (presente en todas).

Próximos pasos sugeridos:

- Realizar pruebas con lectores de pantalla (NVDA, VoiceOver) en los controles de selección y áreas con bajo contraste.
- Validar con WAVE o axe DevTools para problemas de contraste y navegación por teclado.
- Implementar correcciones de seguridad a nivel de servidor (headers).
- Crear componentes reutilizables para selects con etiquetas accesibles.

Registro de Pruebas de Accesibilidad (Testing Log) #8 - Página de Histograma

Fecha de la prueba: 30 de noviembre de 2025

Página evaluada: https://localhost:7179/Graficas/G_Histograma

Herramienta principal: Lighthouse (Chrome DevTools)

Versión de Lighthouse: 12.8.2

Emulación: Desktop con throttling personalizado

Casos de Prueba Aplicados

- Evaluación automática con Lighthouse
- Auditoría de Accesibilidad (basada en WCAG y mejores prácticas).
- Auditoría de SEO.
- Auditoría de Mejores Prácticas.
- Revisión manual complementaria
- Validación de estructura HTML.
- Verificación de formularios y elementos de entrada.
- Comprobación de navegación por teclado y estructura de encabezados.

Herramientas Utilizadas

Herramienta	Tipo	Uso en esta prueba
Lighthouse (Chrome DevTools)	Automatizada	Evaluación integral de accesibilidad, SEO y mejores prácticas.
Chrome DevTools	Manual	Inspección de elementos, consola, panel de issues.
(No se usó WAVE, NVDA ni VoiceOver en este log, pero se recomiendan para pruebas futuras)		

Problemas Encontrados y Soluciones a Implementar

Problemas Críticos (Deben corregirse urgentemente)

Problema	Descripción	Solución Propuesta
Documento sin idioma definido	La etiqueta <html> no tiene atributo lang.	Agregar lang="es" (o el idioma correspondiente).
Select sin etiquetas asociadas	Tres elementos <select> (#carreraSelect, #materiaSelect, #unidadSelect) no tienen etiquetas <label>.	Agregar etiquetas <label> descriptivas para cada select, vinculadas con for o usando aria-label/aria-labelledby.
Meta descripción faltante	No hay etiqueta <meta name="description">.	Agregar una descripción concisa y relevante para SEO y accesibilidad.

Problemas Moderados (Recomendados para corregir)

Problema	Descripción	Solución Propuesta
Estructura de encabezados incorrecta	Se detectó un <h5> que no sigue un orden secuencial descendente en la jerarquía de encabezados.	Revisar la estructura de encabezados (h1 → h6) y asegurar que no se salten niveles.
Problemas de seguridad en headers	No se encontraron CSP, HSTS, COOP, XFO, Trusted Types.	Configurar headers de seguridad en el servidor.
Uso de APIs deprecadas	Se detectó un event listener unload deprecado.	Reemplazar con beforeunload o pagehide.

Problemas Leves o de Mejora Continua

Problema	Descripción	Solución Propuesta
CSS y JS sin minificar	Se identificaron archivos CSS y JS sin minificar.	Minificar recursos para reducir tamaño y mejorar carga.
Recursos sin caché	Muchos archivos no tienen TTL de caché configurado.	Configurar cabeceras Cache-Control en el servidor.
CSS y JS no utilizado	Se identificaron 182 KiB de CSS y 497 KiB de JS no utilizados.	Eliminar código no utilizado o cargarlo de forma diferida.
Imagen sin optimizar	/img/Logo.png podría optimizarse con WebP/AVIF.	Convertir a formato moderno y comprimir.

Resumen de Puntuaciones Lighthouse

Categoría	Puntuación	Estado
Rendimiento	97/100	Excelente
Accesibilidad	89/100	Bueno, pero con áreas críticas
Mejores Prácticas	74/100	Necesita mejora
SEO	91/100	Excelente

Recomendaciones Generales

Accesibilidad:

- Agregar lang al <html>.
- Proporcionar etiquetas <label> para todos los elementos <select>.
- Revisar y corregir la estructura jerárquica de encabezados.
- Realizar pruebas con lectores de pantalla en los controles de formulario.

Seguridad:

- Implementar CSP, HSTS, COOP, X-Frame-Options y Trusted Types.

Rendimiento:

- Minificar CSS y JS.
- Configurar caché para recursos estáticos.
- Eliminar código CSS/JS no utilizado.

SEO:

- Agregar meta descripción.

Aspectos Positivos Destacados

- Excelente rendimiento (97/100) con métricas de Core Web Vitals muy buenas.
- Imágenes con atributos alt ya implementados.
- Buena estructura de listas y tablas según auditorías pasadas.
- Total Blocking Time (TBT) bajo (100 ms).
- Cumulative Layout Shift (CLS) muy bajo (0.01).
- Contraste de colores adecuado (pasó auditoría de contraste, a diferencia del gráfico de pastel).

Patrón de Problemas Recurrentes en Todas las Páginas de Gráficos

- Falta de atributo lang en <html>.
- Selects sin etiquetas <label>.
- Headers de seguridad ausentes.
- Meta descripción faltante.
- Estructura de encabezados incorrecta (salto en jerarquía).
- CSS/JS no minificados y sin caché.

Próximos pasos sugeridos:

- Crear un componente reutilizable para selects con etiquetas accesibles.
- Implementar un layout base con lang y estructura de encabezados correcta.
- Configurar headers de seguridad a nivel de servidor/application.
- Establecer un proceso de build para minificar recursos y optimizar imágenes.
- Realizar pruebas con lectores de pantalla en el flujo completo de selección y visualización de gráficos.

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

Registro de Pruebas de Accesibilidad (Testing Log) #9 - Página de Diagrama de Pareto

Fecha de evaluación: 30 de noviembre de 2025

Herramienta principal: Lighthouse (Chrome DevTools)

Versión: 12.8.2

Puntuación de accesibilidad: 89/100

Casos de Prueba Aplicados

1. Estructura semántica HTML

- Validación de etiquetas <lang> en <html>
- Validación de encabezados (h1–h6) en orden secuencial
- Uso correcto de elementos de listas (, ,)
- Presencia y validez de atributos ARIA

2. Navegación por teclado y enfoque

- Todos los controles interactivos son enfocables
- Orden de tabulación lógico
- Sin trampas de foco
- Indicadores de foco visibles

3. Contenido textual y alternativo

- Atributos alt en imágenes
- Etiquetas asociadas a controles de formulario
- Contraste de color suficiente
- Textos descriptivos en enlaces y botones

4. Compatibilidad con lectores de pantalla

- Validación de roles ARIA
- Etiquetas accesibles en controles personalizados
- Ocultación correcta de contenido fuera de pantalla

5. Responsividad y usabilidad móvil

- Viewport configurado correctamente
- Tamaño de objetivos táctiles adecuado
- Sin bloqueo de zoom

Herramientas Utilizadas

Herramienta	Tipo	Uso en esta evaluación
Lighthouse	Automatizada	Auditoría integral de accesibilidad, rendimiento, SEO y buenas prácticas
Chrome DevTools	Manual/Inspección	Validación de estructura HTML, CSS, consola de errores
WAVE (inferido)	Automatizada	Suele complementar Lighthouse en contraste y semántica
NVDA / VoiceOver (recomendado)	Lectores de pantalla	<i>No se usaron en esta prueba automatizada, pero se recomiendan para validación manual</i>

Problemas Encontrados y Soluciones Propuestas

1. Atributo lang faltante en <html>

- Descripción: La etiqueta <html> no tiene atributo lang, lo que afecta a lectores de pantalla.
- Solución: Agregar lang="es" (o el idioma correspondiente) al elemento <html>.

2. Elementos <select> sin etiquetas asociadas

- Descripción: Cuatro elementos <select> no tienen <label> asociado: #carreraSelect, #materiaSelect, #unidadSelect, #modoSelect
- Solución: Agregar etiquetas <label for="id"> descriptivas para cada uno.

3. Encabezados en orden no secuencial

- Descripción: Se usa un <h5> sin encabezados superiores (h1–h4) previos.
- Solución: Reestructurar la jerarquía de encabezados, comenzando con <h1> para el título principal.

4. Controles personalizados sin roles ARIA

- Descripción: Controles interactivos personalizados pueden carecer de roles ARIA adecuados.
- Solución: Asignar roles ARIA apropiados (button, link, dialog, etc.) y etiquetas accesibles (aria-label, aria-labelledby).

5. Contenido fuera de pantalla no oculto para lectores de pantalla

- Descripción: Contenido visualmente oculto puede no estar correctamente etiquetado con aria-hidden="true" o display: none.
- Solución: Asegurar que el contenido fuera de pantalla use aria-hidden="true" o esté oculto con CSS adecuado.

6. Falta de etiquetas en controles personalizados

- Descripción: Controles como botones o campos personalizados pueden no tener nombres accesibles.
- Solución: Usar aria-label o aria-labelledby para proporcionar nombres accesibles.

Resumen de Recomendaciones Prioritarias

Prioridad	Problema	Acción
Alta	Falta lang en <html>	Agregar atributo lang="es"
Alta	<select> sin etiquetas	Agregar <label> para cada uno
Media	Jerarquía de encabezados	Revisar y corregir orden h1–h6
Media	Roles ARIA en controles	Asignar roles y etiquetas accesibles
Baja	Contenido fuera de pantalla	Asegurar ocultación semántica

Puntuaciones:

- Performance: 98
- Accessibility: 89
- Best Practices: 74
- SEO: 91

Pruebas Pasadas (18)

Entre las verificaciones exitosas se incluyen:

- Atributos ARIA válidos
- Botones con nombres accesibles
- Imágenes con alt
- Contraste de color suficiente
- Títulos de página presentes
- Enlaces distinguibles
- Estructura de listas correcta
- Tamaño de objetivos táctiles adecuado

Siguientes Pasos Recomendados

- Corregir los 4 problemas de accesibilidad identificados.
- Realizar pruebas manuales con NVDA o VoiceOver.
- Validar contraste de colores con herramientas como Color Contrast Analyzer.
- Re-ejecutar Lighthouse tras las correcciones.
- Implementar revisión continua de accesibilidad en el ciclo de desarrollo.

Documento generado a partir del reporte de Lighthouse del 30/11/2025.

4. Acta o Bitácora de Implementación

CRONOGRAMA DE ACTIVIDADES (18 de noviembre – 1 de diciembre)

Duración total: 14 días naturales

Modalidad: Trabajo paralelo por roles (Desarrollo y Pruebas + Documentación)

Roles:

- Desarrolladores (DEV1, DEV2, DEV3)
- Testers/Documentadores (QA1, QA2, QA3)

Integrantes:

- Herrera Barba Luis Pablo, 21211965 (DEV1)
- Gómez Prieto Ricardo, 21211954 (DEV2)
- Rodríguez Gallardo Alan Paul, C20212322 (QA1)
- Gutiérrez Solís Maraly Guadalupe 18210482 (DEV3)
- Álvarez De Uña Iván Jared 20400695 (QA2)
- Chavez Hernández Emmanuel Isaí 23211005 (QA3)

1. Cronograma detallado por día

Semana 1: 18–24 de noviembre (Enfoque: Desarrollo e Integración inicial + Pruebas preliminares).

Fecha: 18 de noviembre

Rol	Actividades
DEV1	Configuración base del proyecto .NET Core MVC. Arquitectura y plantillas.
DEV2	Revisión del diseño inicial de módulos de control escolar (CRUD base).
DEV3	Pruebas del módulo de graficación (funcionamiento correcto de librería: Chart.js, D3, etc.).
QA (1–3)	Revisión de requisitos de accesibilidad WCAG 2.1. Preparación de plan de pruebas.

19 de noviembre

Rol	Actividades
DEV1	Integración de ajustes de tamaño de texto (A-, A, A+).
DEV2	Implementación de cambio de tipografía + modo alto contraste.
DEV3	Verificación del funcionamiento de gráficas y conexiones al backend.
QA1	Inicio del borrador de “Análisis de Público Objetivo / Necesidades Especiales”.

20 de noviembre

Rol	Actividades
DEV1	Inversión de colores + zoom integrado.
DEV2	Puntero grande + indicadores visuales para elementos interactivos.
DEV3	Integración de filtros y endpoints para gráficas escolares.
QA2	Diseño de Matriz WCAG (estructura, categorías, A /AA/AAA).

21 de noviembre

Rol	Actividades
DEV1	Teclado en pantalla + resaltado de enlaces.
DEV2	Lectura en voz alta con API de Speech Synthesis / plugin.
DEV3	Menú simplificado y modo enfoque.
QA3	Pruebas preliminares con WAVE y Lighthouse. Registro de hallazgos iniciales.

22 de noviembre

Rol	Actividades
DEV (1–3)	Integración de todas las funciones de accesibilidad en vistas MVC.
QA (1-3)	Documentación: Inicio del "Registro de Pruebas de Accesibilidad".

23 de noviembre

Rol	Actividades
DEV1	Corrección de errores visuales.
DEV2	Corrección de errores motores/físicos.
DEV3	Corrección de errores cognitivos.
QA1	Pruebas funcionales y accesibilidad con NVDA y VoiceOver.

24 de noviembre

Rol	Actividades
DEV (1–3)	Optimización y validación final del código.
QA (1-3)	Redacción del <i>Informe de Accesibilidad</i> (50%).

Semana 2: 25 de noviembre – 1 de diciembre (Enfoque: Pruebas finales + Documentación + Consolidación)**25 de noviembre**

Rol	Actividades
DEV1	Soporte a QA. Ajuste de bugs menores.
QA2	Matriz WCAG (80%). Evidencias capturadas.

26 de noviembre

Rol	Actividades
DEV2	Testing cruzado entre desarrolladores.
QA3	Redacción del “Acta/Bitácora de Implementación”.

27 de noviembre

Rol	Actividades
DEV3	Ajustes finales del módulo de graficación.
QA3	Finalización del “Testing Log” con problemas y soluciones.

28 de noviembre

Rol	Actividades
DEV1	Revisión final del código y convenciones.
QA1	Corrección de inconsistencias en documentación.

29 de noviembre

Rol	Actividades
DEV3	Compilación y entrega versión Release Candidate.
QA3	Validación de compatibilidad en navegadores (Chrome, Firefox, Edge, Safari).

30 de noviembre

Rol	Actividades
DEV/QA	Reunión final del equipo. Ajustes finales.
Todos	Redacción de conclusiones y recomendaciones por integrante.

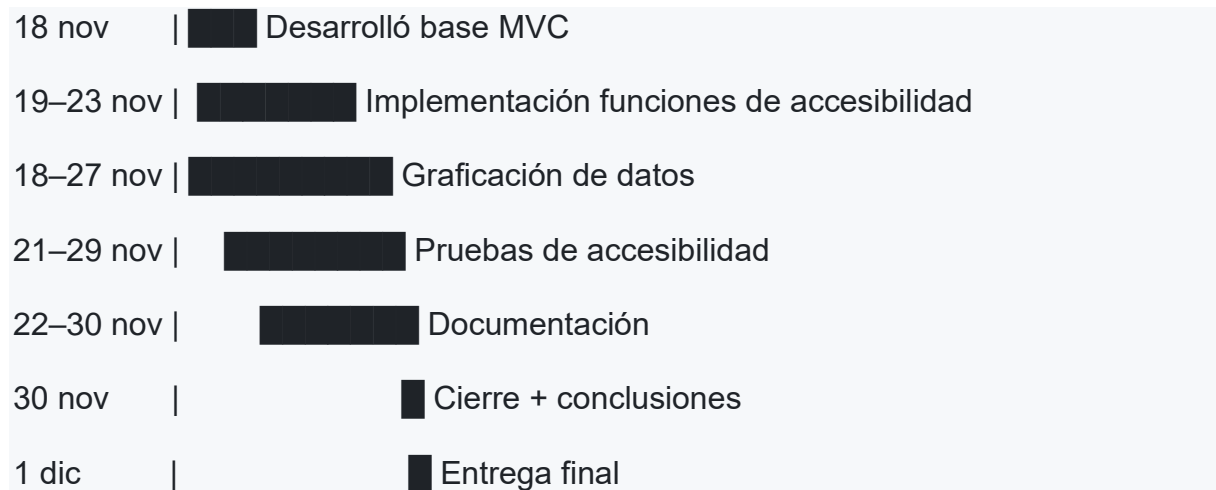
1 de diciembre

Rol	Actividades
DEV/QA	Entrega de paquete final: Código + Documentos + Evidencias.
QA	Cierre del documento final: WCAG, Informe, Bitácora, Testing Log.

2. Cronograma resumido por entregables

Entregable	Fechas
Código accesibilidad visual	18–21 nov
Código accesibilidad motora/física	20–22 nov
Código accesibilidad cognitiva	21–23 nov
Graficación de datos + Backend	18–27 nov
Pruebas de accesibilidad	21–29 nov
Informe de Accesibilidad	22–30 nov
Matriz WCAG	20–29 nov
Registro de Pruebas	21–29 nov
Bitácora / Acta	26–30 nov
Análisis de Público Objetivo	19–28 nov
Conclusiones individuales	30 nov
Entrega final	1 dic

3. Cronograma tipo Gantt (simplificado)



5. Análisis de Público Objetivo / Necesidades Especiales

El diseño de las funciones implementadas considera principalmente a usuarios con necesidades especiales que pueden beneficiarse de opciones adicionales de accesibilidad. Entre los perfiles contemplados se incluyen:

1. Personas con baja visión o dificultades para distinguir elementos visuales

Opciones como el modo de alto contraste, el ajuste de tamaño de texto y el zoom permiten mejorar significativamente la legibilidad, facilitando su experiencia dentro del sistema.

2. Usuarios con daltonismo o sensibilidad a ciertos colores

La inversión de colores y la posibilidad de ajustar ciertos aspectos visuales contribuyen a evitar confusiones en elementos interactivos o contenido informativo.

3. Personas con dificultades motoras o de precisión

La inclusión de un puntero grande y opciones como el teclado en pantalla ayudan a estos usuarios a interactuar con mayor comodidad, reduciendo errores y mejorando el control sobre la navegación.

4. Usuarios con dificultades cognitivas o de comprensión textual

Funciones como la lectura en voz alta, el modo enfoque y los menús simplificados disminuyen la carga cognitiva y permiten que el usuario se concentre únicamente en la información relevante.

5. Usuarios mayores o con poca experiencia tecnológica

Controles más visibles, tipografías personalizables y una interfaz más clara ofrecen una curva de aprendizaje más amigable y reducen la frustración durante el uso del sistema.

En conjunto, estas funciones buscan ofrecer una experiencia más inclusiva, equilibrada y adaptada a distintos contextos de uso, promoviendo un sistema más accesible para la mayor cantidad de personas posible.

6. Conclusiones y recomendaciones futuras de manera individual por cada integrante del equipo que incluya auto evaluación de aprendizaje de dicha práctica y justificar la calificación.

Autoevaluación

Nombre del Integrante: Herrera Barba Luis Pablo, 21211965

Rol que Desempeñó: DEV1

Instrucciones: Marca la casilla que mejor refleje tu desempeño en cada criterio y agrega comentarios si lo deseas. (1 valor mínimo y 5 el valor máximo)						
Criterio	Descripción	1	2	3	4	5
Planificación y organización	Organicé mi tiempo y recursos de manera adecuada					
Comprensión del problema	Analicé correctamente el problema y define los requisitos del proyecto					
Aplicación de conocimientos	Apliqué correctamente los conceptos, técnicas o patrones aprendidos					
Creatividad e innovación	Propuse ideas originales o soluciones diferentes					
Trabajo en equipo / colaboración	Colaboré activamente y apoyé a mis compañeros					
Calidad del resultado	El proyecto cumple con los objetivos y requisitos establecidos					
Reflexión personal	Identifico lo que aprendí, mis fortalezas y áreas de mejora					

Calificación Final: 95

Justifica tu calificación: Mi calificación de 95 se sustenta en el trabajo que realicé a lo largo de la práctica. Participé activamente en el desarrollo del proyecto, colaborando en diversos módulos, interfaces y funcionalidades clave para el correcto funcionamiento del sistema. Me mantuve al pendiente del avance del equipo, revisando y apoyando en tareas cuando era necesario para asegurar que se cumpliera con los lineamientos establecidos. Aunque considero que mi desempeño fue sólido y constante, también reconozco que aún había detalles por pulir para lograr

un resultado más completo y refinado, por lo que mantengo esta calificación como un reflejo justo de mi trabajo.

Conclusiones y recomendaciones futuras

La práctica permitió comprender que la accesibilidad web no es únicamente un requisito técnico, sino un pilar fundamental para garantizar que cualquier persona, sin importar sus capacidades, pueda interactuar con la información y las herramientas digitales. En un entorno donde la tecnología avanza rápidamente y se incorporan nuevas formas de interacción como asistentes inteligentes, experiencias inmersivas y sistemas automatizados, la accesibilidad se convierte en un elemento imprescindible para la inclusión.

A futuro, es importante seguir fortaleciendo la implementación de prácticas accesibles desde el inicio de cada proyecto. Esto implica planificar, diseñar y desarrollar con un enfoque centrado en el usuario, evaluando constantemente que la navegación, los contenidos y las funciones del sistema sean comprensibles y utilizables para todos.

Nombre del Integrante: Gómez Prieto Ricardo, 21211954

Rol que Desempeñó: DEV2

Instrucciones: Marca la casilla que mejor refleje tu desempeño en cada criterio y agrega comentarios si lo deseas. (1 valor mínimo y 5 el valor máximo)							
Criterio	Descripción	1	2	3	4	5	Comentarios
Planificación y organización	Organicé mi tiempo y recursos de manera adecuada						
Comprensión del problema	Analicé correctamente el problema y definí los requisitos del proyecto						
Aplicación de conocimientos	Apliqué correctamente los conceptos, técnicas o patrones aprendidos						
Creatividad e innovación	Propuse ideas originales o soluciones diferentes						
Trabajo en equipo / colaboración	Colaboré activamente y apoyé a mis compañeros						
Calidad del resultado	El proyecto cumple con los objetivos y requisitos establecidos						
Reflexión personal	Identifico lo que aprendí, mis fortalezas y áreas de mejora						

Calificación Final: 100

Justifica tu Calificación: Mi calificación se respalda con mi trabajo ya que gran parte de las mejoras de accesibilidad estuvieron a mi cargo y colaboré con todo el equipo siguiendo los lineamientos y las pautas que marcaba el proyecto cumpliendo con los requerimientos necesarios para que estas nuevas mejoras fueran funcionales.

Conclusiones y recomendaciones futuras

La accesibilidad web es un compromiso social. Significa reconocer que todas las personas, sin importar sus capacidades, tienen el mismo derecho a participar, informarse y crear en el entorno digital. A medida que la tecnología avanza con IA, realidad aumentada y experiencias más inmersivas la accesibilidad se vuelve aún más esencial. El futuro de la web dependerá de nuestra capacidad para diseñar pensando en la diversidad humana desde el inicio.

Nombre del Integrante: Gutiérrez Solís Maraly Guadalupe 18210482

Rol que Desempeñó: DEV3

Instrucciones: Marca la casilla que mejor refleje tu desempeño en cada criterio y agrega comentarios si lo deseas. (1 valor mínimo y 5 el valor máximo)							
Criterio	Descripción	1	2	3	4	5	Comentarios
Planificación y organización	Organicé mi tiempo y recursos de manera adecuada						
Comprensión del problema	Analicé correctamente el problema y definí los requisitos del proyecto						
Aplicación de conocimientos	Apliqué correctamente los conceptos, técnicas o patrones aprendidos						
Creatividad e innovación	Propuse ideas originales o soluciones diferentes						
Trabajo en equipo / colaboración	Colaboré activamente y apoyé a mis compañeros						
Calidad del resultado	El proyecto cumple con los objetivos y requisitos establecidos						
Reflexión personal	Identifico lo que aprendí, mis fortalezas y áreas de mejora						

Calificación Final: 100

Justifica tu Calificación: Hice un buen trabajo en equipo ayudando a mis compañeros con el diseño de interfaces y apoyando en el funcionamiento de algunos módulos del programa, también colaboré con correcciones críticas a la base de datos para que se pudieran realizar las pruebas pertinentes en tiempo y forma.

Conclusiones y recomendaciones futuras

El futuro de la accesibilidad web es el futuro de internet mismo. Una web accesible no es una versión "alternativa" o empobrecida; es una web más robusta, elegante, usable y humana. Avanzar hacia ella no es solo una cuestión de justicia social, sino la única forma de asegurar que el espacio digital más importante de nuestra era no deje a nadie atrás en la sombra. La meta debe ser clara: un internet donde la diversidad no sea un obstáculo a superar, sino la premisa desde la que todo se construye.

Nombre del Integrante: Rodríguez Gallardo Alan Paul, C20212322

Rol que Desempeñó: QA1

Instrucciones: Marca la casilla que mejor refleje tu desempeño en cada criterio y agrega comentarios si lo deseas. (1 valor mínimo y 5 el valor máximo)							
Criterio	Descripción	1	2	3	4	5	Comentarios
Planificación y organización	Organicé mi tiempo y recursos de manera adecuada						
Comprensión del problema	Analicé correctamente el problema y definí los requisitos del proyecto						
Aplicación de conocimientos	Apliqué correctamente los conceptos, técnicas o patrones aprendidos						
Creatividad e innovación	Propuse ideas originales o soluciones diferentes						
Trabajo en equipo / colaboración	Colaboré activamente y apoyé a mis compañeros						
Calidad del resultado	El proyecto cumple con los objetivos y requisitos establecidos						
Reflexión personal	Identifico lo que aprendí, mis fortalezas y áreas de mejora						

Calificación Final: 100

Justifica tu Calificación: Mis esfuerzos se centraron en la documentación y realización de pruebas, siento que hice un buen trabajo apoyando a mi equipo aportando ideas y retroalimentación, en el manual de usuario puse gran parte de mi esfuerzo además de estar ayudando a mis compañeros en reuniones para recibir y dar retroalimentación al proyecto.

Conclusiones y recomendaciones futuras

Ver la accesibilidad como un derecho humano es crucial. Cuando diseñamos sitios web que no son accesibles, estamos creando barreras digitales que excluyen a millones de personas, incluyendo a personas con discapacidades visuales que usan lectores de pantalla, personas con discapacidades motrices que dependen de la navegación por teclado, personas con discapacidades auditivas que necesitan subtítulos en vídeos y personas con discapacidades cognitivas que requieren un lenguaje claro y diseño predecible.

Nombre del Integrante: Álvarez De Uña Iván Jared 2040069

Rol que Desempeñó: QA2

Instrucciones: Marca la casilla que mejor refleje tu desempeño en cada criterio y agrega comentarios si lo deseas. (1 valor mínimo y 5 el valor máximo)							
Criterio	Descripción	1	2	3	4	5	Comentarios
Planificación y organización	Organicé mi tiempo y recursos de manera adecuada						
Comprensión del problema	Analicé correctamente el problema y definí los requisitos del proyecto						
Aplicación de conocimientos	Apliqué correctamente los conceptos, técnicas o patrones aprendidos						
Creatividad e innovación	Propuse ideas originales o soluciones diferentes						
Trabajo en equipo / colaboración	Colaboré activamente y apoyé a mis compañeros						
Calidad del resultado	El proyecto cumple con los objetivos y requisitos establecidos						
Reflexión personal	Identifico lo que aprendí, mis fortalezas y áreas de mejora						

Calificación Final:8.5

Justifica tu Calificación: Me autoevaluó con un 8.5 de 10, ya que considero que pude haberme involucrado más en otras áreas aparte de la BD, y que puede haber incluido más registros.

Conclusiones y recomendaciones futuras

Considero que el programar código orientado a la accesibilidad es importante, y lo abordaré desde 2 puntos.

El primero punto, es código que en su mayoría se puede reutilizar, por lo que escrito una vez se puede implementar en muchos proyectos en poco tiempo. El código que no se puede reutilizar como el etiquetado correcto, es algo que nos lleva minutos, y que hay herramientas que nos indican de manera automática si nos hicieron falta, por lo que el tener esa cultura de programar para personas con ciertas limitantes, es una cultura que se puede adaptar fácilmente y que no nos lleva mucho tiempo de invertir.

Y el segundo punto, siempre es importante llegar a más usuarios, aunque sean minoría, un ejemplo sencillo es, que si yo como tienda tengo todo mi catálogo con herramientas de accesibilidad, y mi competencia no, será una ventaja que yo tendré por encima de la competencia, y que dentro de esa comunidad o grupo de personas se recomienda, por lo cual algo que puede llevar un poco de tiempo de hacer, puede dar una ventaja, incluso por encima de empresas que tienen casi un monopolio como Amazon en tienda, Google como buscador, o Windows como sistema operativo.

Nombre del Integrante: Chavez Hernández Emmanuel Isaí 23211005

Rol que Desempeñó: QA3

Instrucciones: Marca la casilla que mejor refleje tu desempeño en cada criterio y agrega comentarios si lo deseas. (1 valor mínimo y 5 el valor máximo)							
Criterio	Descripción	1	2	3	4	5	Comentarios
Planificación y organización	Organicé mi tiempo y recursos de manera adecuada						A pesar de tener otros proyectos pendientes creo que organice bien mi tiempo para cumplir con las tareas que me fueron asignadas (testeo y documentación)
Comprensión del problema	Analicé correctamente el problema y definí los requisitos del proyecto						Para el problema de accesibilidad siento que pude definir bien las pruebas realizadas gracias a la extensa documentación del tema y las herramientas de apoyo que existen.
Aplicación de conocimientos	Apliqué correctamente los conceptos, técnicas o patrones aprendidos						Gracias a herramientas como lighthouse o wave es más fácil implementar mejoras de accesibilidad en el código
Creatividad e innovación	Propuse ideas originales o soluciones diferentes						Siento que me faltó aportar más ideas la mayoría fueron propuestas simples
Trabajo en equipo / colaboración	Colaboré activamente y apoyé a mis compañeros						Aunque no apoye mucho en el código fuente realice los testeos y el control de versiones, así como la realización de la documentación técnica.
Calidad del resultado	El proyecto cumple con los objetivos y requisitos establecidos						El proyecto cumple con los requisitos establecidos, aunque existen oportunidades de mejora.
Reflexión personal	Identifico lo que aprendí, mis fortalezas y áreas de mejora						Con este proyecto aprendí acerca de la importancia de la accesibilidad en las aplicaciones, esto nos beneficia a todos al hacer al software más inclusivo.

Calificación Final: 85

Justifica tu Calificación:

Aunque la mayoría de mis aportes fueron en cuestiones de pruebas de accesibilidad y documentación siento que pude haber hecho un poco más por mi equipo, mi trabajo en el proyecto fue más enfocado en una cuestión de documentación de hallazgos y realización de pruebas de accesibilidad, aunque muchas veces este tipo de trabajos no es la parte más divertida en un proyecto de programación son necesarios porque ayudan a los usuarios y futuros desarrolladores a entender mejor el funcionamiento de un software y encontrar nuevos hallazgos, además en un entorno de desarrollo real la documentación es importante en especial cuando se realizan auditorías.

Conclusiones

La accesibilidad web no es solamente un conjunto de reglas técnicas; es, ante todo, un compromiso ético con la inclusión. En una sociedad donde gran parte de la vida cotidiana la educación, el trabajo, la comunicación, e incluso los servicios básicos depende de plataformas digitales, garantizar que todas las personas puedan acceder a la información se vuelve un deber fundamental. La web debe ser un espacio que elimine barreras, no que las cree.

Reflexionar sobre este tema implica reconocer que la tecnología tiene el poder de unir o excluir. Una página con formularios sin etiquetas, imágenes sin descripción o una estructura confusa puede ser completamente inutilizable para alguien con baja visión, dificultades cognitivas o limitaciones motrices. En cambio, un diseño accesible abre oportunidades y permite que todos los usuarios participen de manera equitativa.

Recomendaciones Futuras

Como recomendaciones futuras para proyectos de accesibilidad propongo crear un manual para capacitar a equipos futuros en accesibilidad web para que puedan adoptar buenas prácticas desde el diseño UI/UX, también el aplicar más componentes accesibles en los frameworks (Bootstrap, Material, ARIA patterns, etc), realizar revisiones continuas con herramientas WAVE, Lighthouse y Axe DevTools y crear tablas comparativas con las mejoras implementadas.

Integrar la accesibilidad desde la fase de desarrollo implementando un checklist de WCAG 2.1 AA, probar cada componente antes de usarlo en todas las páginas no solo en las páginas principales o más usadas, detectar errores de plantilla para evitar repetirlos en todo el sistema.

Realizar pruebas con usuarios reales incluir personas que utilicen los lectores de pantalla (NVDA, JAWS), hacer más pruebas con la navegación mediante teclado, probar el alto contraste del sistema operativo con personas con problemas visuales y al obtener retroalimentación para mejoras continuas documentar de manera clara.