

FACULTAD DE INFORMÁTICA

TESINA DE LICENCIATURA

TÍTULO: Accesa Web: Complemento para accesibilizar y adaptar páginas web AUTORES: Castillo Miguel Alejandro, Martínez María Emilia DIRECTOR: Mg. Harari Ivana, Lic. Javier Diaz CODIRECTOR: -ASESOR PROFESIONAL: -CARRERA: Licenciatura en Informática



Resumen

En Argentina, existe la Ley 26.653 que solicita a la comunidad IT a respetar las normas de accesibilidad (WCAG) para brindar inclusión y las mismas oportunidades a personas en diversa situación de discapacidad. Dado el contexto actual muchas gestiones, trámites, entre otros que se realizan de manera online poseen una escasa accesibilidad lo que dificulta el acceso a la información y servicios.

Por lo expuesto anteriormente, esta tesis propone una herramienta, que sea add-on del navegador Google Chrome, para accesibilizar los sitios web y que brinde asistencia para las personas con diferentes discapacidades. Se propone incluir una serie de funcionalidades de apoyo y de configuración para adaptar la navegación, percepción e interacción de las páginas web de acuerdo a las necesidades de la persona usuaria.Esto permitirá que una audiencia más amplia pueda disfrutar de la web sin enfrentarse a condicionamientos ni restricciones arbitrarias.

Palabras Clave

Normas WCAG Pautas WAI-ARIA Accesibilidad web Ley 26.653 Ley 26.378 Extension Google Chrome

Trabajos Realizados

Realizamos diferentes etapas de investigación, la primera constó de indagar el marco legal de la aprobación de la ley de accesibilidad 26.653 y su implementación. Luego se investigó la situación actual de la web desde la perspectiva de diferentes perfiles de usuarios. Se desarrolló una extensión de Google Chrome que permite al usuario accesibilizar y adaptar los sitios web dependiendo de sus necesidades. Como tarea final hicimos distintas pruebas con los usuarios para probar la herramienta.

Conclusiones

Como conclusión AccesaWeb es una herramienta que considera, asiste e incluye a la persona usuaria en su contexto de discapacidad aceptando sus distintas formas de interacción y navegación. Constituye un aporte al cumplimiento de la ley de accesibilidad 26.653 garantizando los derechos de acceso a la información y contenido de las páginas web.

Trabajos Futuros

- Agregar más idiomas a la herramienta.
- Resolver problemas de accesibilidad en iframes.
- Adicionar más funciones de accesibilidad HTML.
- Mejorar el diseño de los comandos.
- Agregar subtítulos a los videos.
- Reconocimiento de imágenes.
- Migración del aplicativo de la herramienta a manifest V3.

Fecha de la presentación: 29/11/2024

*El presente trabajo adhiere al uso de lenguaje inclusivo, respetando los derechos de las personas a ser nombradas según su elección.

Agradecimientos

Queremos expresar nuestro más sincero agradecimiento a nuestros directores de tesina Ivana y Javier por su apoyo, orientación y comentarios que fueron fundamentales para la finalización de esta tesis.

A las personas que colaboraron en las encuestas y las que colaboraron en las pruebas de usabilidad de la herramienta por su tiempo, dedicación y por aportar su punto de vista y opinión como usuarios de la web.

A Javier Perez y Victoria Mussa por el aporte y explicación de sus trabajos que nos sirvieron de inspiración.

Por último un agradecimiento a nuestras madres por el apoyo y acompañamiento recibido durante la carrera y el desarrollo de este trabajo.

¡Muchas gracias a todos! Miguel y Emilia.

Índice

Agradecimientos	2
Capítulo 1	5
Introducción	5
Motivación	5
Objetivo	6
Estructura de la tesina	6
Capítulo 2	7
Normativa y legislación	8
Pautas y estándares de accesibilidad web	8
Leyes vigentes en la República Argentina	16
Leyes internacionales	18
Conclusión del capítulo	21
Capítulo 3	22
Análisis de accesibilidad en la web	22
Herramientas de validación	22
Encuestas realizadas a usuarios con diferentes discapacidades	31
Conclusión del capítulo	39
Capítulo 4	40
Aplicaciones de software sobre accesibilidad	40
Complementos	40
Aplicaciones con funciones de accesibilidad	46
Funciones de configuración en sistemas operativos	49
Páginas Web con opciones de accesibilidad	51
Tesinas de grado sobre accesibilidad	54
Conclusión del capítulo	56
Capítulo 5	58
Desarrollo de la extensión AccesaWeb	58
¿Qué es AccesaWeb?	58
Implementación de AccesaWeb	59
Arquitectura	60
Tecnologías utilizadas	61
Diseño de la herramienta	73
Funcionalidades	75
Funcionalidades recuperadas y mejoradas de tesis analizadas	81
Desarrollo de la herramienta	82
Guía de instalación	102

Alcance de la accesibilidad	103
Conclusión del capítulo	103
Capítulo 6	104
Pruebas de usabilidad	104
Objetivos de la prueba	104
Selección de participantes	105
Análisis de los resultados	105
Capítulo 7	109
Conclusiones y trabajos futuros	109
Bibliografía	110
Anexos	115

Capítulo 1 Introducción

Motivación

Argentina y el resto del mundo atravesó una dura pandemia desde principios del año 2020. Este complicado contexto de salud dio lugar a un confinamiento social preventivo que se implementó de manera obligatoria y rigurosa en la mayoría de los países, y Argentina no fue la excepción. Indudablemente muchos negocios, comercios, entidades gubernamentales, académicas y estudiantiles, edificios públicos, organismos, entre otros, tuvieron que cerrar sus puertas. Esta situación trajo consigo el cese por tiempo indeterminado de la atención al público y los trámites presenciales, obligó a que las gestiones, actividades laborales, comerciales y educativas, se realicen de manera online y remota, siendo ésta, la única modalidad disponible. Por cuestiones de costos, evitar traslados y minimizar la atención presencial, esta modalidad virtual permaneció más allá de las restricciones y el confinamiento, haciendo que se evidencie aún más las dificultades que tienen las personas con discapacidad para acceder a la web. De esta manera la accesibilidad digital se volvió una cualidad indispensable para lograr una web más inclusiva.

Existen muchos sitios webs que aún no implementan las normas estándares e internacionales de accesibilidad web del consorcio World Wide Web (W3C) [1], guía de accesibilidad para contenidos web (WCAG por sus siglas en inglés) [2], para brindar inclusión y las mismas oportunidades para personas con discapacidad. Los motivos pueden ser varios, ya sea por falta de formación, de tiempo, de interés, falta de conciencia o responsabilidad social, malas estimaciones sobre los costos, escasos recursos tanto humanos como económicos, entre otros. Si bien en Argentina se encuentra la Ley 26653 [3] aprobada por unanimidad en noviembre de 2010, que solicita a la comunidad IT (tecnología de la información) a respetar dichas normas, éstas rara vez se cumplen en su totalidad. Esta situación trae aparejado marginalidad y falta de acceso a la información por parte de poblaciones vulnerables que tienen el mismo derecho de acceso a la información y de gozar de las oportunidades que brinda la web, que los demás usuarios. Además se encuentra la Convención sobre los Derechos de las Personas con Discapacidad y su protocolo facultativo que en su artículo 9 sobre accesibilidad dice [4]:

"A fin de que las personas con discapacidad puedan vivir en forma independiente y participar plenamente en todos los aspectos de la vida, los Estados Partes adoptarán medidas pertinentes para asegurar el acceso de las personas con discapacidad, en igualdad de condiciones con las demás, al entorno físico, el transporte, la información y las comunicaciones, incluidos los sistemas y las tecnologías de la información y las comunicaciones, y a otros servicios e instalaciones abiertos al público o de uso público, tanto en zonas urbanas como rurales."

Este documento promueve el acceso de las personas con discapacidad a la información en sistemas y tecnologías de información, lo que incluye la web.

Con respecto a las complejidades de acceso, existe un alto grado de contaminación visual en las páginas web, es decir, un bombardeo constante de avisos, spyware, spam y publicidad, lo que dificulta la lectura y el procesamiento de la información para personas que padecen problemas tanto cognitivos como con afecciones visuales. También afecta al lector de pantalla, producto de apoyo utilizado por las personas ciegas, por aquellas con escaso resto visual o que requieren un refuerzo auditivo del contenido, al cual se le dificulta procesar esta información prioritaria de los sitios web. Además, muchos presentan recursos multimediales como fotos, GIF o videos sin descripción alternativa que no llegan a ser percibidos ni comprendidos por parte del usuario con discapacidad ni siquiera utilizando herramientas automáticas de asistencia.

Por lo expuesto anteriormente, esta tesis propone una herramienta, que sea add-on del navegador Chrome, para accesibilizar los sitios web y que brinde asistencia para las personas con discapacidad y/o limitaciones de interacción. Se propone incluir una serie de funcionalidades de apoyo y de configuración para adaptar la navegación, percepción e interacción de las páginas web de acuerdo a las necesidades de la persona usuaria. Esto permitirá que una audiencia más amplia pueda disfrutar de la web sin enfrentarse a condicionamientos ni restricciones arbitrarias.

Objetivo

El objetivo de esta tesis es desarrollar una herramienta tecnológica que permita accesibilizar automáticamente sitios web para personas con discapacidad, fomentando el cumplimiento de la Ley nacional n° 26.653 sobre Accesibilidad en los contenidos de las páginas web y a su vez respetando la Convención sobre los Derechos de las Personas con Discapacidad y su protocolo facultativo. Esta herramienta a implementar como complemento del navegador, además de accesibilizar cualquier sitio web visitado por la persona usuaria, dispondrá de herramientas de apoyo y de configuración de acuerdo a sus necesidades específicas, ofreciéndole un servicio que le permita acceder al contenido de la información con la menor cantidad de restricciones posibles.

Estructura de la tesina

Para cumplir con el objetivo propuesto, se desarrolló la herramienta tecnológica AccesaWeb cuyo proceso de construcción queda constatado en este documento que contiene 8 capítulos.

- Capítulo 1- Introducción: el primer capítulo presenta la motivación que nos llevó a realizar este proyecto, el objetivo de la tesis y la estructura de la misma.
- Capítulo 2- Normativa y Legislación: este capítulo describe la legislación provincial, nacional e internacional para dar a conocer el marco legal de las leyes que amparan la accesibilidad web. Además se describen las normas o estándares de la web basándonos en la W3C.
- Capítulo 3- Análisis de accesibilidad en la web: este capítulo analiza y compara los resultados de las encuestas realizadas a usuarios con diferentes discapacidades.
- Capítulo 4- Herramientas de accesibilidad: en este capítulo se recopilan y comparan las diferentes herramientas existentes que proveen accesibilidad, tales como páginas web, complementos del navegador y aplicaciones de escritorio y móviles.
- Capítulo 5- Desarrollo de AccesaWeb: en este capítulo se detallan las decisiones técnicas y conceptuales de la herramienta, las tecnologías utilizadas, la arquitectura sobre la cual está montada. Se explican los diversos módulos que componen la extensión, su funcionalidades, las dificultades encontradas en el proceso de desarrollo y el alcance de accesibilidad que brinda a los sitios web.
- Capítulo 6- Pruebas de usabilidad: en este capítulo se describe el objetivo de las pruebas de usabilidad, como fue el proceso de selección de candidatos para las pruebas, se informa y analiza los resultados de las mismas con los diferentes usuarios de la extensión.
- Capítulo 7- Conclusiones y trabajos futuros: este capítulo se presentan las conclusiones a las que se llegaron durante el desarrollo de este trabajo y se resume a modo de conclusión las contribuciones de esta tesina, así como las diferentes propuestas de trabajos futuros.

Capítulo 2 Normativa y legislación

En el capítulo anterior ofrecimos una introducción para dar a conocer el motivo de nuestra elección de tesina, el objetivo y la estructura de la misma. Expresamos la problemática acerca de la falta de accesibilidad e inclusión en sitios webs, el cual fue evidenciado aún más durante el contexto de la pandemia.

En este capítulo daremos a conocer el marco legal relacionado a la accesibilidad web, tanto nacional como internacional. Estas leyes protegen los derechos de las personas con discapacidad y son necesarias porque muchos aspectos de la vida actual se realizan en un entorno digital o virtual, y por tanto, ayudan a salvaguardar derechos como el derecho a la información, el derecho a la educación y el derecho a la sanidad, por citar algunos derechos humanos básicos.

Como punto de partida en esta sección del proyecto se describirán las pautas y estándares vigentes publicadas por la World Wide Web Consortium que especifican cómo hacer el contenido web accesible. También se incluye el marco histórico de dichas pautas, los principios sobre los cuales están basados, el alcance y definiciones de las mismas.

Para asegurar que estas pautas sean aplicadas se dictan leyes que las contemplan, dentro de este capítulo damos a conocer las leyes y/o convenciones a nivel nacional, provincial, adicionalmente seleccionamos algunas leyes que rigen en otros países.

Pautas y estándares de accesibilidad web

Las Web Content Accessibility Guidelines, WCAG (Pautas de accesibilidad para el contenido web) son una parte de las directrices de accesibilidad web que especifican cómo hacer el contenido web más accesible, especialmente a personas con discapacidades, pero también a otros agentes como aquellos dispositivos con altas limitaciones, como lo son algunos teléfonos celulares. Son publicadas por la Web Accessibility Initiative (Iniciativa de Accesibilidad Web, WAI) [5] parte del World Wide Web Consortium que es un consorcio internacional e independiente creado en 1999, este reúne a organizaciones gubernamentales, no gubernamentales y empresas, cuya finalidad es la evolución e interoperatividad de la web para fomentar la accesibilidad. El objetivo es promover el acceso universal, es decir la posibilidad para todos los individuos de acceder a los recursos de la web, cualquiera sea el material o software, la infraestructura de red, lengua materna, cultura, localización geográfica, o aptitudes físicas o mentales.

El objetivo de las directrices de accesibilidad para el contenido web consiste en facilitar el acceso de las personas con discapacidad, desarrollando pautas de

accesibilidad, mejorando las herramientas para la evaluación y reparación de accesibilidad web, llevando a cabo una labor educativa y de concientización en relación a la importancia del diseño accesible de páginas web y abriendo nuevos campos de accesibilidad a través de la investigación en esta área. Para ello proporcionan un estándar compartido único para la accesibilidad del contenido web que satisface las necesidades de individuos, organizaciones y gobiernos a nivel internacional.

Seguir las pautas hará que el contenido sea más accesible para una gama más amplia de personas con discapacidades, incluidas adaptaciones para ceguera y baja visión, sordera y pérdida auditiva, movimiento limitado, discapacidades del habla, fotosensibilidad y combinaciones de estos, algunas adaptaciones para discapacidades de aprendizaje y limitaciones cognitivas. También para personas mayores con habilidades cambiantes debido al envejecimiento. Mejoran la usabilidad y disponibilidad del contenido web para los usuarios en general, sea cual sea el agente de usuario que estén utilizando (navegador de escritorio, navegador de voz, teléfono móvil, computadora de abordo, etc.) o las limitaciones bajo las que pueden estar operando (entornos ruidosos, habitaciones con poca o excesiva iluminación, ambiente, etc). Éstas no desalientan a los desarrolladores a usar imágenes, videos u otro recurso audiovisual; sino que explican cómo hacer que el contenido multimedia sea más accesible para una amplia audiencia.

Aunque estas pautas cubren una amplia gama de temas, no pueden abordar las necesidades de todas las personas con los tipos, grados y combinaciones de discapacidad.

Marco histórico de las WCAG

En 1999 el W3C (World Wide Web Consortium) publicó las WCAG 1.0 (Pautas de Accesibilidad para el Contenido Web), que establecen 14 pautas (principios generales del diseño accesible) para el desarrollo de webs con contenidos accesibles. Abordan desde las herramientas de autor, hasta los contenidos, pasando por las aplicaciones con las que los usuarios acceden. El cumplimiento de estas 14 pautas determina el grado de adecuación de la página web, distinguiéndose tres niveles, de menor a mayor accesibilidad: "A", "AA" y "AAA", siendo esta última la que indica la mayor accesibilidad. El cumplimiento de uno de estos tres niveles habilita el uso del distintivo homologado correspondiente, que notifica a los visitantes el grado de adecuación a los criterios de accesibilidad y se ubica al pie de página.

En diciembre de 2008, se publica la versión 2.0.de las pautas WCAG, estas alcanzan el estatus de recomendación, siendo una continuación y una evolución de las WCAG 1.0. Cuentan con un nuevo tipo de organización, quedando ordenadas a partir de 4 principios fundamentales para la accesibilidad del contenido: perceptible, operable, comprensible y robusto.

- 1. **Perceptible**: la información y los componentes de la interfaz de usuario deben ser presentados a las personas de modo que puedan percibirlos. Por ejemplo, debe ofrecerse una alternativa textual a todos los contenidos no-textuales (imágenes, gráficos, animaciones, etc.), y los elementos gráficos deben poder distinguirse claramente.
- 2. Operable: los componentes de la interfaz y la navegación deben ser fácilmente utilizables por todos los usuarios. Por ejemplo, ofrecer otros métodos de entrada alternativos al mouse, como los atajos de teclado.
- **3. Comprensible**: los contenidos y la información deben ser inteligibles. Por ejemplo, el sitio web tiene que contar con una disposición de la información que sea predecible, con un idioma establecido, y un lenguaje simple y claro.
- 4. Robusto: el contenido debe ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo las ayudas técnicas. Por ejemplo, los lectores de pantalla.

Desde octubre de 2012, las WCAG 2.0 son un estándar ISO (International Organization for Standardization) [6] a través de la Norma ISO/IEC 40500 que dice [7]:

"[Pautas de accesibilidad al contenido web (WCAG) 2.0] cubre una amplia gama de recomendaciones para hacer que el contenido web sea más accesible. Seguir estas pautas hará que el contenido sea accesible para una gama más amplia de personas con discapacidades, que incluyen ceguera y baja visión, sordera y pérdida auditiva, problemas de aprendizaje, limitaciones cognitivas, movimiento limitado, discapacidades del habla, fotosensibilidad y combinaciones de estos. Seguir estas pautas también hará que su contenido web sea más útil para los usuarios en general[...]."

El 5 de junio de 2018 se publicó la recomendación de las Pautas de Accesibilidad para el Contenido Web, las WCAG 2.1 que proponen mejorar las pautas de accesibilidad para tres grupos específicos de usuarios:

- 1. Las personas con discapacidad cognitiva o del aprendizaje.
- 2. Las personas con baja visión.
- 3. Las personas con discapacidad que acceden desde dispositivos móviles.

Las WCAG 2.1 es una extensión de la WCAG 2.0, adiciona 17 criterios entre los 3 niveles: 5 de nivel A, 7 de nivel AA y 5 de nivel AAA.

Nivel A:

Atajos de teclado, gestos del puntero, cancelación del puntero, etiqueta en el nombre, activación mediante movimiento.

Nivel AA:

Orientación, identificación del propósito de entrada, reajuste del texto, contraste de contenido no textual, espaciado de texto, contenido con hover o focus, mensajes de estado.

Nivel AAA:

Identificación del propósito, tiempos de inactividad, animación iniciada desde interacciones, tamaño del objetivo, mecanismos de entrada concurrentes.

Adicionalmente la versión 2.1 de la normativa introduce varios cambios significativos respecto a la versión 2.0:

- Se añade una nueva pauta 2.5 "Modalidades de entrada" al principio de Operabilidad, pasando de 12 a 13 pautas.
- Se incluyen criterios específicos para mejorar la accesibilidad en dispositivos móviles, como la orientación de pantalla y el reajuste del texto.
- Se añaden nuevos términos al glosario para apoyar los nuevos criterios de conformidad.

La última versión de las pautas de accesibilidad del contenido en la web es la 2.2 publicada en octubre de 2023. Los nuevos criterios introducidos en esta versión se distribuyen entre los niveles de conformidad A y AA.

Nivel A:

Enfoque no oscurecido (Mínimo), movimientos de arrastre, entrada redundante.
 Nivel AA:

Enfoque no oscurecido (Mejorado), apariencia de enfoque, tamaño del objetivo (Mínimo), ayuda consistente, autenticación accesible (Mínimo y Mejorada).

Estructura de la WCAG

Las WCAG se componen de una colección de documentos interrelacionados. Por una parte, el propio "Estándar", que es normativo y no sufre cambios; y por otra los documentos de "Técnicas", "Cómo cumplir las WCAG" y "Comprender las WCAG", que son informativos y se actualizan cada cierto tiempo.

Como podemos observar en la imagen 2.1. Las WCAG están organizadas en cuatro capas, cuya complejidad crece exponencialmente.



Imagen 2.1. Organización de las WCAG. Imagen extraída del libro Accesibilidad Web. WCAG 2.1 de forma sencilla [8].

4 Principios

Son la base de la accesibilidad web, todo sitio web debe ser perceptible, operable, comprensible y robusto, estos principios no son evaluables.

13 pautas

A su vez, los principios se dividen en pautas con el objetivo de agrupar de un modo lógico los criterios de conformidad. Estas pautas tampoco son evaluables.

- El sitio web debe ser "Perceptible": los sitios web pueden ser visitados por personas con diferentes tipos de preferencias y posibilidades, pero también por robots (buscadores, traductores, etc). La información e interfaz deben tener en cuenta esta necesidad y por ello deben ofrecer alternativas a los usuarios que no pueden utilizar alguno de sus sentidos. Las pautas de este principio son:
 - Pauta 1.1, alternativas textuales a contenidos no textuales.
 - Pauta 1.2, alternativas a medios tempodependientes, es decir, vídeo y audio.
 - Pauta 1.3, el contenido es adaptable a diferentes formas de presentación.
 - Pauta 1.4, distinguible: el contenido es fácil de ver y escuchar.
- El sitio web debe ser "**Operable**": los diseñadores y programadores deben proporcionar elementos de interacción y de navegación que puedan ser manejados por personas con diferentes posibilidades. Las pautas de este principio son:
 - Pauta 2.1, funcionalidad accesible por teclado.
 - Pauta 2.2, los usuarios tienen tiempo suficiente para leer y usar el contenido.

- Pauta 2.3, El contenido no causa convulsiones.
- Pauta 2.4, navegable: los usuarios pueden navegar, encontrar contenido y saber dónde están en todo momento.
- Pauta 2.5, facilita formas de introducir información.
- El sitio web debe ser "Comprensible": si el usuario no entiende de que se está hablando o se siente perdido es un problema. Se debe diseñar el sitio web (incluyendo la información e interfaz de usuario) para que sea fácil de usar. Las pautas de este principio son:
 - Pauta 3.1, el contenido es fácil de leer y de comprender.
 - Pauta 3.2, el contenido aparece y se maneja de una forma predecible.
 - Pauta 3.3, ayuda en la introducción de datos para evitar y corregir errores.
- El sitio web debe ser "Robusto": este es el principio más dependiente de la tecnología. Se basa en la capacidad del sitio web para ser transmitido e interpretado por los diferentes agentes de usuario. La única pauta de este principio es:
 - Pauta 4.1, el contenido es compatible con las herramientas de usuario actuales y futuras.

Los agentes de usuario son programas diseñados para facilitar la visualización y la interacción con contenidos en internet. Estos incluyen:

- Los navegadores web: son esenciales para acceder a páginas y servicios en línea. Ejemplos de navegadores populares son Google Chrome, Firefox, Safari y Edge, cada uno con sus propias características y funciones que mejoran la experiencia del usuario.
- Reproductores multimedia: que permiten la reproducción de archivos de audio y vídeo. Algunos ejemplos destacados son QuickTime, Real Player y Windows Media Player. Estos reproductores son cruciales para disfrutar de contenido multimedia en línea, como videos, música y transmisiones en vivo.
- Los plug-ins: son complementos que amplían las capacidades de los navegadores y reproductores. Algunos plug-ins permiten la visualización de contenido interactivo, animaciones y aplicaciones específicas en el navegador. Estos elementos son importantes para ofrecer experiencias más ricas y dinámicas en la web.

Además de estos programas, también se consideran agentes de usuario otras aplicaciones que ayudan en la interacción con internet, como herramientas de gestión de descargas y extensiones del navegador que mejoran la funcionalidad.

En conjunto, todos estos programas contribuyen a la forma en que los usuarios acceden, experimentan e interactúan con la vasta cantidad de información y servicios disponibles en línea. Los productos de apoyo, también conocidos como ayudas técnicas, son programas o dispositivos diseñados para proporcionar funcionalidades adicionales que satisfacen las necesidades específicas de los usuarios con discapacidad. Estos productos complementan las aplicaciones de usuario principales, facilitando una experiencia más accesible y cómoda en el uso de tecnología. Entre los ejemplos más comunes de productos de apoyo se encuentran

- Los magnificadores de pantalla: que amplían el contenido visual para personas con discapacidades visuales. Estos dispositivos permiten a los usuarios ver textos e imágenes con mayor claridad, mejorando su capacidad para interactuar con el contenido digital.
- Los teclados en pantalla: ofrecen una alternativa para aquellos que tienen dificultades para usar un teclado físico. Estos teclados virtuales permiten a los usuarios ingresar texto y comandos utilizando un mouse o dispositivos de entrada alternativos. Además, los punteros grandes son herramientas que facilitan la navegación en pantalla para personas con problemas de motricidad fina o visibilidad. Al tener un tamaño mayor, estos punteros permiten una selección más fácil y precisa de elementos en la interfaz.
- Otras ayudas técnicas: como software de lectura de pantalla que convierte el texto en voz, permitiendo a las personas con discapacidad visual acceder a la información escrita. Asimismo, hay dispositivos de entrada alternativos, como trackballs o dispositivos de seguimiento ocular, que ofrecen diferentes formas de interactuar con la computadora.

En conjunto, estos productos de apoyo no solo mejoran la accesibilidad, sino que también fomentan la independencia y la inclusión de los usuarios con discapacidad en entornos digitales. Al proporcionar herramientas adecuadas, se contribuye a un acceso equitativo a la información y a una mejor calidad de vida.

Nuestro sitio web debe interactuar con toda esta tecnología, así como con sus versiones futuras, de manera que el contenido permanezca accesible, aunque las tecnologías y los agentes de usuario evolucionen.

78 criterios de conformidad

Los criterios de conformidad sí son evaluables y se escalan en 3 niveles:

- A (o Simple-A), máxima prioridad: es el nivel más bajo de conformidad. El cumplimiento de las pautas en este nivel es esencial para que el contenido web sea accesible para las personas con discapacidad.
- AA (o Doble-A), mediana prioridad: este es el nivel de conformidad recomendado para la mayoría de los sitios web, ya que cubre los problemas más comunes de accesibilidad que afectan a un gran número de usuarios.

Cumplir con los requisitos del nivel AA asegura que el sitio sea accesible para la mayoría de los usuarios con discapacidades.

 AAA (o Triple-A), mínima prioridad: este es el nivel más alto de conformidad y representa un nivel ideal de accesibilidad. Cumplir con las pautas de este nivel garantiza que el contenido sea accesible para la mayor cantidad de personas posible, incluyendo aquellos con discapacidades muy específicas o complejas. Este nivel es el ideal, pero no siempre es necesario y puede ser difícil de implementar en todos los casos.

Para decidir el nivel que debía tener cada criterio, el W3C estableció las siguientes preguntas:

"¿Este criterio de conformidad es esencial? Es decir, ¿si no se cumple este criterio, los agentes de usuario (incluyendo los productos de apoyo) pueden o no reproducir la página web?

¿Este criterio de conformidad es aplicable para todos los sitios web, independientemente de su tema, del tipo de contenido, de la tecnología usada, etc.?

¿Los creadores de contenidos pueden aprender en menos de una semana a cumplir este criterio de conformidad?

¿Cómo limitaría este criterio de conformidad la apariencia, la funcionalidad, la forma de presentación, la libertad de expresión, el diseño y la estética de la página?

¿Hay una alternativa u otro camino para acceder al contenido si no se cumple el criterio de conformidad?"

Todos los criterios de conformidad son importantes: los más bajos aseguran que se pueda acceder al contenido de las páginas web y los más altos ayudan a mejorar su usabilidad y a llegar a un mayor número de personas. Cuando se evalúan los sitios web, se repasan los criterios y se comprueba si se cumplen o no.

- Si el sitio cumple con todos los criterios de nivel A, pero no cumple con los de nivel AA o AAA, la conformidad será solo A.
- Si el sitio cumple con los criterios de nivel A y también con los de nivel AA, pero no con los de nivel AAA, la conformidad será AA.
- Si el sitio cumple con los criterios de nivel A, AA y AAA, la conformidad será AAA.

Más de 580 técnicas y errores

Cada criterio de conformidad propone una serie de técnicas a seguir y documenta una serie de errores a evitar para alcanzar la conformidad. Por su parte cada técnica y cada error tienen su procedimiento de prueba.

Estas técnicas están diseñadas para ayudar a garantizar que el contenido web sea accesible a las personas con diferentes tipos de discapacidades, ya sean visuales, auditivas, motrices, cognitivas, entre otras. Estas técnicas pueden clasificarse de la siguiente manera:

- Técnicas aceptadas: son las mejores prácticas que cumplen con los criterios de accesibilidad. Son ampliamente utilizadas y recomendadas para asegurar que el sitio sea accesible.
- **Técnicas extensibles**: son aquellas que no son estrictamente necesarias pero pueden ser útiles dependiendo del contexto. Si bien no son un requisito obligatorio para cumplir con las pautas, pueden mejorar la accesibilidad.
- **Técnicas fallidas**: son aquellas que se han probado, pero no cumplen con los criterios de accesibilidad. Estas técnicas pueden haberse implementado incorrectamente en algunos sitios y, por tanto, deben evitarse.

Los errores comunes son prácticas que se deben evitar porque van en contra de las pautas de accesibilidad y, en muchos casos, pueden crear barreras significativas para los usuarios con discapacidades. Los errores se pueden clasificar de la siguiente manera:

- Errores de contenido: son problemas directamente relacionados con el contenido del sitio web, como la falta de texto alternativo en imágenes o el uso de colores con un contraste insuficiente.
- Errores de estructura: incluyen problemas con la organización del contenido, como la falta de encabezados adecuados (<h1>, <h2>, etc.), lo que dificulta la navegación de los usuarios con discapacidades cognitivas o motoras.
- Errores de interacción: se refieren a los problemas con la accesibilidad de los formularios, botones y otras formas de interacción en la web, como no proporcionar suficientes indicaciones de enfoque para los usuarios que navegan con teclado.

Leyes vigentes en la República Argentina

La Ley n° 26.378, de adhesión a la Convención sobre los Derechos de las Personas con Discapacidad y su Protocolo Facultativo fue sancionada el 11 de Mayo de 2008 por el Senado y Cámara de Diputados de la Nación Argentina reunidos en el honorable congreso de la Nación y promulgada el 6 de Junio de 2008. Dicha Convención sobre los Derechos de las Personas con Discapacidad es un documento aprobado por la Asamblea General de las Naciones Unidas el 13 de diciembre de 2006, en aras de promover y proteger los derechos inherentes de las personas con discapacidad.

El propósito de esta ley es promover, proteger y asegurar el goce pleno y en condiciones de igualdad de todos los derechos humanos y libertades fundamentales por todas las personas con discapacidad y promover el respeto de su dignidad inherente. Las personas con discapacidad incluyen a aquellas que tengan deficiencias físicas, mentales, intelectuales o sensoriales a largo plazo que al interactuar con diversas barreras, pueden impedir su participación plena y efectiva en la sociedad en igualdad de condiciones con las demás.

La Ley n° 26.653, Accesibilidad de la Información en las Páginas Web fue sancionada el 3 de noviembre de 2010 por el Senado y Cámara de Diputados de la Nación Argentina reunidos en el honorable congreso de la Nación, promulgada de hecho el 26 de noviembre de 2010.

Según el artículo 1 de la ley establece que:

"El Estado nacional, los entes públicos no estatales, las empresas del Estado y las empresas privadas concesionarias de servicios públicos, deberán respetar en los diseños de sus páginas web las normas y requisitos sobre accesibilidad de la información que faciliten el acceso a sus contenidos, a todas las personas con discapacidad."

La ley según el artículo 3 define la accesibilidad web como:

"La posibilidad de que la información de la página web, pueda ser comprendida y consultada por personas con discapacidad y por usuarios que posean diversas configuraciones en su equipamiento o en sus programas."

En el artículo 5 de la ley se delega la elección de las normas y requisitos de accesibilidad a la Oficina Nacional de Tecnologías de la Información (ONTI) [9] de la siguiente forma:

"Las normas y requisitos de accesibilidad serán las determinadas por la Oficina Nacional de Tecnologías de la Información (ONTI), debiendo actualizarse regularmente dentro del marco de las obligaciones que surgen de la Convención sobre los Derechos de las Personas con Discapacidad (Ley 26.378)."

A partir de estas dos leyes, en el año 2011 se resuelve mediante la Resolución n° 69/2011 [10] que se aprueba la "Guía de Accesibilidad para Sitios Web del Sector Público Nacional", en la cual se detalla el cumplimiento de la WCAG 1.0. Como la tecnología avanza y las pautas de accesibilidad también, en el año 2014 sale la Disposición n° 2/2014 [11] qué dice en sus artículos:

"ARTICULO 1° — Apruébase la "Norma de Accesibilidad Web 2.0." que como Anexo I forma parte integrante de la presente Disposición.

ARTICULO 2° — Apruébanse los "Niveles Mínimos de Conformidad" que como Anexo II forma parte integrante de la presente Disposición.

ARTÍCULO 3° — Establécese que el nivel mínimo de conformidad a ser cumplimentado por los organismos, durante el primer período evaluatorio deberá ser de CINCUENTA (50) puntos y en el segundo período el umbral de aprobación se establecerá en OCHENTA (80), acorde lo establecido en el Anexo II que aprueba la presente Disposición."

Aprobando el cumplimiento de las WCAG 2.0 al menos en sus niveles mínimos de conformidad pero esta disposición se deja sin efectos al salir la Disposición n° 6/2019 [12] en el año 2019. Esta nueva disposición es más clara sobre los criterios que se deben cumplir y el tiempo que las organizaciones tienen para cumplir con los criterios de conformidad.

A partir de la ley nacional sobre accesibilidad, las provincias se fueron adhiriendo. El 17 de enero de 2019 se publica la Ley n° 15115 [13] sancionada por el Senado y Cámara de Diputados de la Provincia de Buenos Aires, la cual establece que el Estado provincial, sus organismos descentralizados, las empresas del Estado, las empresas privadas concesionarias de servicios públicos y las empresas prestadoras de bienes y servicios deben cumplir con normas y requisitos de accesibilidad en sus páginas web. Esta legislación garantiza que todas las plataformas digitales, tanto públicas como privadas (en cuanto a servicios y bienes), sean accesibles para personas con discapacidad.

El 1 de abril de 2014 el Senado y Cámara de Diputados de la Provincia de Mendoza sanciona la Ley n°8.654 Accesibilidad web a personas con discapacidad [14], donde se adhiere a la Ley nacional n° 26.653 sobre accesibilidad web.

El 18 de junio de 2019, la Provincia de Río Negro adhiere a la Ley nacional nº 26.378 bajo la Ley nº 4532 [15] que aprueba la Convención sobre los Derechos de las Personas con Discapacidad y su Protocolo Facultativo, incorporando su texto como artículo 79 a la Ley nº 2055 -Régimen de Promoción Integral de las Personas con Discapacidad.

Leyes internacionales

Se ha estudiado legislación sobre accesibilidad web en otros países como Estados Unidos, Reino Unido, España y la Unión Europea.

Estados Unidos fue el primer país en desarrollar una ley que trate sobre accesibilidad web en 1998, la Section 508 Amendment to the Rehabilitation Act of 1973 [16]. En realidad se trata de una enmienda a la Ley Workforce Rehabilitation Act de 1973. La Section 508 exige que toda la tecnología electrónica y de la información que sea desarrollada o adquirida por las Agencias Federales sea accesible a las personas con discapacidad. Las pautas de esta ley tienen muchas similitudes con las WCAG 1.0.

En Reino Unido la Ley Disability Discrimination Act (DDA) 1995 (Part III Access to Goods and Services) junto con otras leyes sobre discriminación promulgadas durante más de 40 años fueron sustituídas por la Ley Equality Act 2010 [17] en Inglaterra, Gales y Escocia, pero DDA sigue en vigor en Irlanda del Norte. Por ahora no existen leyes concretas que hagan referencia a la accesibilidad de los sitios web, pero Royal National Institute of Blind People recomienda en UK Law for websites, el cumplimiento como mínimo del nivel mínimo de WCAG 1.0 para todos los sitios web y el nivel AA para los sitios web del gobierno.

El 8 de diciembre de 1999, la Comisión Europea puso en marcha la iniciativa eEurope - Una Sociedad de la Información para todos. Uno de sus objetivos clave era garantizar que la sociedad de la información no se traduzca en exclusión social.

18

En junio de 2000 el Consejo Europeo aprobó el plan de acción eEurope 2002: una sociedad de la información para todos [18].

En septiembre de 2001, la Comisión Europea aprobó la Comunicación titulada eEurope 2002: accesibilidad de los sitios web públicos y de su contenido, que pretende que los sitios web sean más accesibles para las personas mayores y las personas con discapacidad. Dentro de esta comunicación se anima a las demás instituciones europeas y a los Estados miembros a adoptar las medidas necesarias para que los sitios web sean accesibles. Se recomienda la adopción de las Pautas de Accesibilidad al Contenido en la Web 1.0 del WAI.

En abril de 2002 el Informe sobre la Comunicación de la Comisión y la Resolución del Parlamento Europeo apuesta por el desarrollo de una internet accesible para todos los ciudadanos. La Comisión establece que las administraciones públicas deberán procurar en todo momento mejorar la accesibilidad de sus páginas web y buscar nuevas formas de ofrecer los contenidos y los servicios, a medida que evolucionen las nuevas tecnologías y aparezcan nuevas versiones de las pautas de accesibilidad. Además, la Comisión recomienda que se inste a las organizaciones que reciban fondos públicos de las instituciones europeas o los Estados miembros a hacer que sus sitios web sean accesibles y pide a los estados miembros que fomenten el cumplimiento de las pautas WAI, no sólo en los sitios web públicos de carácter nacional, sino también en los locales y regionales. El citado documento subraya que para que los sitios web sean accesibles es fundamental que satisfagan el nivel AA y que se aplique en su totalidad la prioridad 2 de las Pautas WAI. Además, se propone que se consiga la accesibilidad de los sitios web del sector privado para 2003 y que se empiece por los sitios web que reciben financiación pública. En septiembre de 2005, la Comisión Europea adoptó la comunicación "La accesibilidad electrónica", que tenía como objetivo principal promover un enfoque coherente de las iniciativas relacionadas con la accesibilidad electrónica en los Estados miembros de manera voluntaria, así como fomentar la autorregulación del sector industrial.

En 2006, la Declaración Ministerial de Riga sobre una sociedad de la información incluyente vuelve a incorporar el compromiso de que la totalidad de los sitios web públicos sean accesibles y fija el año 2010 como límite para lograrlo. Para asegurar la accesibilidad se establece que se tienen que cumplir los estándares y las pautas del W3C.

Desde el año 2002, en España se han desarrollado varias leyes que definen los niveles de accesibilidad y fechas de cumplimiento. La Ley n° 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico [19] es la primera ley en España que fija la obligación de que las páginas web de las Administraciones Públicas sean accesibles.El texto original establecía, en su disposición adicional quinta, que las páginas web de la administración pública debían ser accesibles para las personas con

discapacidad y de edad avanzada antes del 31 de diciembre de 2005. No obstante, no especificaba el nivel requerido o en base a qué normativa.

Esta ley ha sido ampliada por el Real Decreto n° 1494/2007, de 12 de noviembre [20] por el que se aprueba el reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios de comunicación social. En él se determinaba que el nivel de adecuación que debían cumplir las páginas web de la Administración pública, o con financiación pública, era el nivel AA de la norma UNE 139.803. Por aquel entonces, la UNE 139803:2004 era equivalente a las WCAG 1.0.

Otra ley importante es la Ley n° 56 de 28 de diciembre del 2007, de Medidas de Impulso de la Sociedad de la Información [21], la misma que hace referencia explícitamente en el cuarto párrafo del apartado Uno en el que obliga a las administraciones públicas a informar sobre el nivel de accesibilidad de sus páginas web, de forma que: "Las páginas de internet de las Administraciones Públicas deberán ofrecer al usuario información sobre su nivel de accesibilidad y facilitar un sistema de contacto para que puedan transmitir las dificultades de acceso al contenido de las páginas de internet o formular cualquier queja, consulta o sugerencia de mejora". Además en el artículo Segundo, esta ley obliga a las empresas que cumplan una serie de condiciones para satisfacer un nivel de accesibilidad en sus páginas web equivalente al nivel AA. Por último, esta ley señala que las Administraciones Públicas promoverán la aplicación de los estándares de accesibilidad en todos los elementos y procesos basados en las nuevas tecnologías de la Sociedad de la Información.

El 3 de septiembre del 2012 fue publicada en el BOE n° 237 de 2/10/2012, la Norma UNE 139803:2012 Requisitos de accesibilidad para contenidos en la web [22] que sustituye a Norma UNE 139803:2004. Esta nueva norma es equivalente a las Pautas de accesibilidad al contenido de la web de la WCAG 2.0 de la W3C ya que las referencia directamente.

El "Real Decreto 1112/2018 sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público" [23] que entró en vigor el 20 de septiembre de 2018, transpone la Directiva europea 2016/2012, y obliga a que todos los sitios web y apps nativas de la Administración Pública, o que reciban financiación pública, sean accesibles.

El Real Decreto deroga además los artículos 5, 6 y 7 del Real Decreto 1494/2007, que determinaban la normativa y el nivel de accesibilidad que tenían que cumplir los sitios web, así como el contenido mínimo que debían incluir en la declaración de conformidad. Ahora deben cumplir con la versión más reciente de la EN 301 549, actualmente equiparada al nivel AA de las WCAG 2.1, pero que incluye más requisitos que éstas.

20

Conclusión del capítulo

Como hemos observado, cada región o país analizado en este capítulo legisla de manera diferente respecto a la accesibilidad web. Algunas de estas leyes siguen los lineamientos de la WCAG en sus distintas versiones, desde la 1.0 hasta la versión más actual, otras se rigen por pautas similares a la WCAG; esto genera una falta de estandarización en materia de legislación, pero podemos concluir que las pautas de la WCAG son las más utilizadas y pueden tomarse como referencia para la implementación de páginas web accesibles. En este trabajo, seguiremos este estándar para accesibilizar las páginas web visitadas por la herramienta.

Capítulo 3

Análisis de accesibilidad en la web

En el capítulo anterior, se analizaron la legislación y las normas sobre la accesibilidad web en distintos lugares del mundo, concluyendo que para este trabajo seguiremos los estándares de la WCAG. En este tercer capítulo, analizaremos herramientas que validan páginas web basándose en las normativas WCAG, que son consideradas como un estándar internacional.

Luego, elaboramos un cuestionario con el objetivo de analizar los problemas más comunes de accesibilidad en la web, identificar las funcionalidades deseables y de uso frecuente, conocer las páginas web que los usuarios visitan habitualmente en su día a día, determinar el porcentaje de usuarios que emplean tecnología asistiva, y obtener una perspectiva del usuario.

Para llevar a cabo este cuestionario, redactamos un informe que incluye la investigación realizada para el diseño del cuestionario, la elección de las preguntas y, finalmente, la presentación y análisis de los resultados obtenidos.

Herramientas de validación

Investigamos varios validadores de accesibilidad web con la finalidad de estudiar sus funcionalidades para incorporar algunos de sus servicios más interesantes a la extensión AccesaWeb. En esta sección utilizamos la misma página para poder comparar las diferentes herramientas analizadas. A continuación listamos los más relevantes:

Accessibility Insights for Web

Es una extensión para Chrome y el nuevo Microsoft Edge que ayuda a los desarrolladores a encontrar y solucionar problemas de accesibilidad en aplicaciones y sitios web.

La herramienta admite dos escenarios principales:

- **FastPass** es un proceso ligero de dos pasos que ayuda a los desarrolladores a identificar problemas de accesibilidad comunes y de alto impacto en menos de cinco minutos.
 - **Verificaciones automatizadas**: la herramienta verifica automáticamente el cumplimiento de aproximadamente 50 requisitos de accesibilidad.
 - Tabulaciones: la herramienta proporciona instrucciones claras, automatización parcial y una ayuda visual que facilita la identificación de problemas críticos de accesibilidad relacionados con el acceso al teclado, como tabulaciones faltantes, trampas de teclado y orden de tabulación incorrecto.

- La **evaluación** permite que cualquier persona con conocimientos de HTML verifique que una aplicación web o un sitio web cumple con las Pautas de accesibilidad al contenido web (WCAG) 2.1 Nivel AA .
 - **Verificaciones automatizadas**: la herramienta verifica automáticamente el cumplimiento de aproximadamente 50 requisitos de accesibilidad.
 - Pruebas manuales: la herramienta proporciona instrucciones paso a paso, ejemplos y orientación sobre cómo solucionar aproximadamente 20 pruebas; muchas pruebas son "asistidas", lo que significa que la herramienta identifica las instancias de prueba o proporciona una ayuda visual.

Con esta herramienta, validamos mediante el modo Fast Pass el login del sitio de la AFIP, detectando 8 errores de accesibilidad. La herramienta proporciona un feedback visual como se muestra en la imagen 3.1, resaltando el elemento origen con un recuadro rojo.



Imagen 3.1. Captura de pantalla del resultado al validar la pantalla de login de la AFIP con la extensión de Accesibility Insights for web para Chrome [24].

Ofrece enlaces a la información sobre las pautas de la WCAG que no se cumplen, muestra la identificación del elemento, su código, y proporciona una o más sugerencias de corrección. Esto se puede observar en la imagen 3.2.



Imagen 3.2. Captura de pantalla del resultado al validar la pantalla de login del AFIP con la extensión de Accesibility Insights for web para Chrome.

TAW

Es una herramienta automática on-line para analizar la accesibilidad de sitios web. Creada teniendo como referencia técnica las pautas de accesibilidad al contenido web (WCAG 2.0) del W3C, cuenta con más de 15 años, siendo la herramienta de referencia en habla hispana.

El objetivo de TAW es comprobar el nivel de accesibilidad alcanzado en el diseño y desarrollo de páginas web con el fin de permitir el acceso a todas las personas independientemente de sus características diferenciadoras. Está destinada tanto a usuarios sin experiencia que quieren conocer el grado de accesibilidad de su sitio web como para profesionales de campo como webmasters, desarrolladores, diseñadores de páginas web, etc. [25].

Con esta herramienta validamos el login del sitio de la AFIP según el nivel de accesibilidad AA de la WCAG 2.1, detectando 14 problemas repartidos entre 4 principios de la WCAG, los cuales se pueden observar en la imagen 3.3. En este caso, la herramienta proporciona un informe detallado de los criterios que no se cumplieron, pero no incluye un enlace directo al elemento donde ocurren los errores, lo que complica la identificación del origen de dichos problemas, dicho informe se puede observar en la imagen 3.4.

~	Email enviado correctamente. Revise su buzón de correo. S	i no recibe el mensaje, compruebe que no haya sido marcad	do como spam. ×
Inicio	Taw		[ES] [EN] [PT]
Servicios			
	Resumen		
Herramientas	14 Problemas en 7 criterios de éxito Son necesarias correcciones Perceptible 3 Operable 7 Comprensible 1 Robusto 3	25 Advertencias en 9 criterios de éxito Es necesario revisar manualmente Perceptible 11 Operable 8 Comprensible 6 Robusto o	15 No verificados en 15 criterios de éxito Comprobación completamente manual ? Perceptible 4 ? Operable 6 ? Comprensible 5 ? Robusto o
	Recurso: https://auth.afip.gob.ar/contribuyente_/logi Acceda al informe detallado para obtener m	in.xhtml 🏾 Fecha: 25/06/2022 19:04 Pautas WCAG 2.0 ás información sobre las incidencias de) Nivel del análisis: AA Tecnologías: HTML, CSS tectadas.
	@ email		
	Recibir informe		Aviso legal Condiciones de Uso CTIC ()

Imagen 3.3. Captura de pantalla del resultado al validar la pantalla de login del AFIP.

1 O Î	🖻 C 🔇 D	• • •			1 d	e 1,314 🔍 <
Inform	ación del análisis					
Recurso CSS	: https://auth.afip.gob.ar/contribuyente_/	login.xhtml Fecha: 25/06/2	022 19:04 Pautas WCAG 2.0	Nivel del análisis: /	AA Tecnolo	gias: HTML,
Percep	tible					
La informac	ón y los componentes de la interfaz de u	usuario deben ser presentado:	s a los usuarios de modo que pueda	an percibirlos.		
Tipología	Comprobación		Técnicas	Resultado I	ncidencias	Números de Lineas
1.1.1-Conte	nido no textual					
Formulario	s Controles de formulario sin etiquetar	8	H44 H65	×	1	<u>100</u>
Imágenes	Imágenes que pueden requerir descripcio	ón larga 👔	<u>H45</u>	0	2	<u>45 130</u>
1.3.1-Inform	nación y relaciones					
Formulario	s Controles de formulario sin etiquetar	8	H44 H65	×	1	<u>100</u>
Presentacio	ón Utilización de etiquetas de presentación	0	<u>G140</u>	0	9	<u>104</u> <u>112</u> <u>131</u> <u>141</u> <u>146</u> <u>151</u> <u>156</u> <u>161</u> <u>167</u>
Estructura semántica	y Inexistencia de elemento h1 👔		<u>H42</u>	×	1	
1.3.3-Carac	terísticas sensoriales					

Imagen 3.4. Captura de pantalla del informe recibido vía mail al validar la pantalla de login del AFIP .

WAVE

Es un conjunto de herramientas de evaluación que ayuda a los autores a hacer que su contenido web sea más accesible para personas con discapacidades. WAVE puede identificar muchos errores de accesibilidad y de las Directrices de Accesibilidad al Contenido Web (WCAG), pero también facilita la evaluación humana del contenido web. Nuestra filosofía es centrarnos en los problemas que sabemos que afectan a los usuarios finales, facilitar la evaluación humana y educar sobre la accesibilidad web [<u>26</u>].

Con esta herramienta, validamos el login del sitio de la AFIP según las pautas de la WCAG 2.0, detectando 4 errores de contraste, los cuales se pueden visualizar en la imagen 3.5, representados con iconos rojos. La herramienta agrupa los resultados de la validación en 6 categorías: errores, errores de contraste, advertencias, características, estructurales y ARIA. Cada elemento evaluado tiene asociado un ícono diferente según la clasificación asignada.



Imagen 3.5. Captura de pantalla del resultado al validar la pantalla de login del AFIP desde la página de WAVE.

Al acceder al ícono, se despliega una breve descripción con dos opciones: 'Referencia', en esta opción podemos ver una descripción detallada de la observación, una sugerencia para resolverla y un enlace a la pauta relacionada; y 'Código', donde podemos visualizar el código del elemento en cuestión. Los botones de referencia y código del icono de error de contraste se visualizan en la imagen 3.6.



Imagen 3.6. Captura de pantalla del resultado al validar la pantalla de login del AFIP con la extensión de WAVE para Chrome.

AChecker

Es una herramienta online para monitorear la accesibilidad web y validar el cumplimiento de AODA (Ley de Accesibilidad para Discapacidades de Ontario). Da la opción de introducir la URL del sitio web a analizar o bien subir archivos HTML para su validación [27]. La siguiente imagen visualiza las opciones mencionadas para subir el origen a ser evaluado y opciones configurables sobre el estándar de accesibilidad, formato de informe y opciones varias respecto a HTML, CSS y código fuente.

-Comprobar accesibilidad n	nediante:	
URL Subir Margen		
DIRECCIÓN:	https://www.afip.gob.ar/landing/default.asp	
	Compruébalo	
- <u>Opciones</u>		
 Habilitar el validador HTML 	 Habilitar el validador CSS 	Mostrar fuente
Pautas para comproba	r	
 BITV 1.0 (Nivel 2) 	O Sección 508	O Ley Stanca
 WCAG 1.0 (Nivel A) 	O WCAG 1.0 (Nivel AA)	\odot WCAG 1.0 (Nivel AAA)
 WCAG 2.0 (Nivel A) 	WCAG 2.0 (Nivel AA)	\odot WCAG 2.0 (Nivel AAA)
Formato del informe		
● Ver por pauta	○Ver por número de línea	

Imagen 3.7. Captura de pantalla del menú de opciones de AChecker.

La herramienta permite seleccionar el estándar de accesibilidad que se desea validar y el formato del informe, que puede estructurarse según las pautas incumplidas o por el número de línea de código. A modo de prueba, validamos el login del sitio de la AFIP según el nivel de accesibilidad AA de la WCAG 2.0, detectando 14 problemas conocidos, 1 posible problema y 83 problemas potenciales. El informe se organiza en 5 pestañas que clasifican los diferentes errores según su grado de ocurrencia; para cada error, se describe la pauta, se ofrece una sugerencia para repararlo y se indica la línea de código donde ocurre. Dicho informe se puede visualizar en la siguiente imagen, a modo de prueba se validó el sitio de AFIP, en esta caso sería la pantalla de login.

🤊 🖸 🔳 a	checker.achecks.ca/checker/index.php	🖻 ✿
Check A	CCESSIBILITY BY: Upload Markup :: https://auth.afip.gob.ar/contribuyente_/login.xhtml	
Check It		
• <u>Optior</u>	<u>15</u>	
Accessin	ility Review	
Accessib Accessib Kno	ility Review ility Review (Guidelines: <u>WCAG 2.0 (Level AA))</u> wn Problems(14) Likely Problems (1) Potential Problems (83) HTML Validation CSS Va Alternatives: Provide text alternatives for any non-text	alidation
Accessib Kno 1.1 Text content	ility Review ility Review (Guidelines: <u>WCAG 2.0 (Level AA))</u> wn Problems(14) Likely Problems (1) Potential Problems (83) HTML Validation CSS Va Alternatives: Provide text alternatives for any non-text	alidation
Accessib Kno 1.1 Text content Succes	ility Review ility Review (Guidelines: <u>WCAG 2.0 (Level AA)</u>) wn Problems(14) Likely Problems (1) Potential Problems (83) HTML Validation CSS Va Alternatives: Provide text alternatives for any non-text ss Criteria 1.1.1 Non-text Content (A)	alidation
Accessib Accessib Kno 1.1 Text content Succes Che	ility Review Guidelines: <u>WCAG 2.0 (Level AA)</u>) Wn Problems(14) Likely Problems (1) Potential Problems (83) HTML Validation CSS Va Alternatives: Provide text alternatives for any non-text SS Criteria 1.1.1 Non-text Content (A) SC 238: <u>input element has alt attribute.</u>	alidation
Accessib Kno 1.1 Text content Succes Che	ility Review Guidelines: WCAG 2.0 (Level AA)) Wn Problems(14) Likely Problems (1) Potential Problems (83) HTML Validation CSS Va Alternatives: Provide text alternatives for any non-text SS Criteria 1.1.1 Non-text Content (A) SC 238: <u>input element has alt attribute</u> . Repair: Remove the alt attribute from all input elements except those with a type attribute value of	alidation
Accessib Kno 1.1 Text content Succes Che	ility Review Guidelines: WCAG 2.0 (Level AA)) wn Problems(14) Likely Problems (1) Potential Problems (83) HTML Validation CSS Va Alternatives: Provide text alternatives for any non-text ss Criteria 1.1.1 Non-text Content (A) sck 238: <u>input element has alt attribute</u> . Repair: Remove the alt attribute from all input elements except those with a type attribute value of <i>Line 104, Column 19</i> :	alidation

Imagen 3.8. Captura de pantalla del resultado al validar la pantalla de login del AFIP con AChecker.

WCAG color contrast checker

Esta herramienta comprueba el contraste de color entre el primer plano y el fondo de los elementos que se encuentran en la página según las WCAG 2, evalúa a partir de las propiedades CSS color y background-color. También, puede simular el daltonismo y evaluar el contraste para dichas simulaciones [28].

Se evaluó el login del sitio web de la AFIP utilizando este validador de contraste configurado en el nivel AA de la WCAG 2.2, detectando 6 errores. El informe incluye un listado con cada error, junto con su valor de contraste, y al seleccionarlo, se resalta el elemento donde ocurre el problema mediante un recuadro rojo. Los errores de contraste detectados y el informe mencionado se pueden visualizar en la siguiente imagen.



Imagen 3.9. Captura de pantalla del resultado al validar la pantalla de login del AFIP con WCAG color contrast checker.

EqualWeb Accessibility Checker

Es una herramienta de auditoría automatizada gratuita para la validación de WCAG 2.1. Este validador analiza el sitio web y proporciona un informe visual completo sobre los problemas de accesibilidad del sitio, incluidas instrucciones y orientación sobre cómo solucionarlos.

Para usar el validador de accesibilidad de EqualWeb, simplemente se hace clic en el ícono de EqualWeb en el navegador Chrome, el validador marcará los problemas de accesibilidad estática y dinámica en las páginas HTML filtradas por los niveles de conformidad WCAG 2.1 A, AA y AAA.

EqualWeb es la única empresa reconocida por el World Wide Web Consortium (W3C) por cumplir plenamente con sus pautas de accesibilidad de contenido web (WCAG 2.1) y las pautas de la Section 508 (ley estadounidense sobre accesibilidad web) en mercados importantes como Italia, Israel, Irlanda, Japón, Francia, Alemania ,Canadá y Estados Unidos [29].

Evaluamos la pantalla de login del sitio web de la AFIP utilizando el escáner de esta herramienta, que identificó errores generales de accesibilidad según las pautas de la WCAG 2.2. Los problemas incluyen contraste, advertencias, y otros. Los resultados de evaluación juntos los distintos problemas mencionados se pueden observar a modo de feedback en la imagen 3.10. Para cada error, se muestra la pauta incumplida, su título y el nivel de cumplimiento de accesibilidad. Además, se detalla el error, incluyendo el elemento HTML afectado y una explicación. La herramienta ofrece una guía para resolver el problema, junto con un botón para solucionarlo.



Imagen 3.10. Captura de pantalla del resultado al validar la pantalla de login de AFIP con Equalweb Accessibility Checker.

Siteimprove Accessibility Checker

Es una herramienta fiable que permite validar de acuerdo a los niveles de conformidad WCAG 2.0 A, AA y AAA. Entre las opciones está también mostrar o no: errores, advertencias y aspectos que deben revisarse manualmente; o aquellos relacionados con determinados perfiles que tienen la responsabilidad de los cambios: editor, webmaster o desarrollador. En el detalle de error indica el criterio, describe el error, lista las técnicas para corregirlo e incluye las instancias del error que se resaltan dentro la página web. Sólo está disponible en inglés [<u>30</u>].

A modo de prueba, utilizamos el complemento en la página de login del sitio web de la AFIP, donde se detectaron 12 incidentes. En el informe generado por la herramienta, se pueden visualizar errores de nivel de accesibilidad A, AA y AAA, el informe junto a su formato se puede visualizar en la imagen 3.11. Al seleccionar uno de estos errores, se despliega una descripción del problema, la cantidad de veces que aparece en la página y un listado de los elementos afectados. Al seleccionar uno de estos elementos, se marca en la pantalla con un recuadro rosa. Además, la herramienta ofrece un botón que genera ejemplos HTML para resolver el problema.



Imagen 3.11. Captura de pantalla del resultado al validar la pantalla de login de AFIP con Siteimprove Accessibility Checker.

Además de esta comparación realizamos un análisis en diferentes páginas con el objetivo de encontrar los errores más comunes en la web. Elegimos las siguientes páginas del país: Mercadolibre, Infobae, Clarín, Banco Nación, Olé, TN y Anses.

La mayoría de los errores que las herramientas detectaron en sus respectivos análisis corresponden fundamentalmente a problemas de contraste entre el texto y sus fondos, además de atributos *aria* faltantes o mal relacionados entre los elementos asociados, tamaño de texto mínimo, interlineado y espaciado entre letras que no cumplen con las medidas mínimas establecidas por las pautas. También encontramos atributos *alt* vacíos en enlaces, imágenes e *iframes,* encabezados vacíos y roles no definidos en algunos elementos como *contenedores* divs y listas.

Encuestas realizadas a usuarios con diferentes discapacidades

Reunimos información mediante una encuesta realizada a personas con o sin discapacidad, con el objetivo de analizar los hábitos y costumbres de los usuarios de la web, qué navegador utiliza la mayoría, si necesitan herramientas de apoyo o de qué manera acceden a la información y qué dificultades tienen al acceder a la misma.

A partir de los datos reunidos, se pudo analizar las necesidades de los usuarios para saber en qué medida nuestra extensión podría ayudar o qué funcionalidades serían más útiles.

La encuesta se dividió en 3 secciones, la primera para los datos personales, otra solo para las personas que se identifiquen con alguna discapacidad y la última sobre el uso de internet. Fue dirigida a 16 personas, incluyendo tanto a individuos con discapacidad como sin discapacidad, abarcando un rango de edad entre los 19 y los 69 años.

En la sección "datos personales" solicitamos el nombre, la edad y si la persona se identifica con alguna discapacidad, en caso de seleccionar "si", se habilita la sección sobre discapacidad, en el caso contrario, se dirige al encuestado a la última sección sobre uso de internet.

Para la redacción de la sección de "**discapacidad**" del cuestionario, investigamos las listas de preguntas sobre funcionamiento del grupo de Washington, lista breve, mejorada y ampliada. Aunque estas preguntas están más orientadas al uso médico, nos basamos en la lista breve extrayendo algunas partes más ligadas a nuestro objetivo que es la accesibilidad web.

El Grupo de Washington sobre Estadísticas de la Discapacidad [31] se formó como grupo de estudio de la Comisión de Estadística de las Naciones Unidas, su propósito principal es promover y coordinar la cooperación internacional en materia de generación de estadísticas sobre la discapacidad, adecuadas para los censos y las encuestas nacionales con el objetivo de proporcionar información básica sobre la discapacidad que sea comparable en todo el mundo.

Mencionaremos la lista breve, que es la misma utilizada en censos nacionales y encuestas donde el espacio es costoso y limitado, siendo más acorde en esos casos, por este motivo la elegimos para nuestra encuesta. La lista breve consta de las seis preguntas siguientes sobre el grado de dificultad para realizar actividades en seis esferas funcionales básicas.

1. ¿Tiene dificultad para ver, incluso cuando usa lentes?

2. ¿Tiene dificultad para oír, incluso cuando usa un audífono?

3. ¿Tiene dificultad para caminar o subir escalones?

4. ¿Tiene dificultad para recordar o concentrarse?

5. ¿Tiene dificultad para lavarse o vestirse (gestionar su autosuficiencia para el cuidado personal)?

6 ¿Tiene dificultad para comunicarse, por ejemplo, entender a los demás o que lo entiendan a usted, cuando se usa un lenguaje normal (habitual)?

Todas las preguntas tienen cuatro categorías de respuestas, que se leen a continuación de cada pregunta.

1. No, ninguna dificultad.

- 2. Sí, cierta dificultad.
- 3. Sí, mucha dificultad.

4. No puedo ver/oír en absoluto. / No puedo realizar esta actividad.

A partir de esta información elaboramos las siguientes preguntas para la encuesta a realizar:

- Primera pregunta: "¿Te reconoces como una persona con discapacidad?" Esta pregunta básica es para dividir a los encuestados entre aquellos que poseen alguna discapacidad y los que no, ya que esto evita tener que responder la sección en el caso que no sea necesario.
- Segunda pregunta: *"¿Tienes alguna dificultad para ver?"* Esta pregunta es para obtener una estadística de las personas con discapacidad visual.
- **Tercera pregunta**: *"¿Tienes alguna dificultad para moverse?"* Esta pregunta es para obtener una estadística de las personas con discapacidad motriz.
- Cuarta pregunta: "¿Tienes alguna dificultad para escuchar o comunicarse?" Esta pregunta es para obtener una estadística de las personas con discapacidad auditiva.
- Quinta pregunta: "¿Tienes alguna dificultad para aprender, concentrarse o relacionarse?" Esta pregunta es para obtener una estadística de las personas con discapacidad cognitiva.

A partir de estas preguntas sobre el perfil de la persona con discapacidad reunimos la siguiente información:



Discapacidad en las personas encuestadas

Gráfico 3.1. Resultados de la encuesta sobre las discapacidades de los usuarios de la web.

Del gráfico 3.1 podemos observar claramente que más de la mitad de los encuestados poseen alguna discapacidad por lo cual podemos concluir que la extensión podría ayudar a la mayoría de los usuarios de internet.

Continuando con la encuesta, en la siguiente sección "**uso de internet**" realizamos las preguntas que más información nos brindan para el desarrollo de la extensión, se busca obtener las funcionalidades más necesarias para los usuarios de la web.

• Primera pregunta: "¿Para qué tipo de tareas cotidianas necesita el uso de internet?" se analiza cuáles tareas son realizadas por los usuarios para las cuales es imprescindible el uso de internet.



Tareas cotidianas con uso de internet

Gráfico 3.2. Resultados de la encuesta sobre las tareas cotidianas que implican usar el navegador.

Del gráfico 3.2, concluimos que un mayor porcentaje de uso del internet se concentra en 3 aspectos, los cuales son, estudiar, informarse y realizar trámites. La mayoría de estas tareas se realizan en páginas gubernamentales y nacionales, sitios académicos y medios de comunicación, como los diarios del país.

 Segunda pregunta: "¿Qué Navegador utilizas?" De esta pregunta obtenemos el navegador más utilizado con el objetivo de desarrollar sobre ese navegador la herramienta propuesta.

Navegadores utilizados





Observando el gráfico 3.3 concluimos que elegimos bien la plataforma para nuestra herramienta de accesibilidad, ya que la mayoría utiliza el navegador Chrome.

 Tercera pregunta: "¿Cuáles herramientas de apoyo o tecnologías usas para acceder a la información o utilizar la computadora?" De esta pregunta podemos analizar cuáles herramientas son usadas y así adaptar nuestra extensión para cubrir una mayor cantidad de necesidades.



Tecnologia de asistencia que usan los encuestados

Gráfico 3.4. Resultados de la encuesta sobre las tecnologías de asistencia que utilizan los usuarios de la web.

Del gráfico 3.4, concluimos que un gran porcentaje de los encuestados presenta problemas visuales por los cuales una mejor accesibilidad web sería una gran mejora en el uso de estas tecnologías de asistencia.

 Cuarta pregunta: "¿Usas extensiones del navegador para ser asistido? ¿Cuáles?" De las respuestas obtenidas reunimos las extensiones que brindan ayuda a los usuarios para después analizarlas cada una y extraer las funcionalidades que podamos integrar en la extensión.



Gráfico 3.5. Resultados de la encuesta sobre las extensiones más utilizadas por los usuarios de la web.

Mediante el gráfico 3.5 podemos concluir que los encuestados utilizan las extensiones a modo de asistencia mientras navegan por la web.

 Quinta pregunta: "¿Qué páginas usas diariamente? Escriba 5 páginas, de ser posible." En base a las páginas recolectadas con esta pregunta podemos hacer validaciones sobre ellas con validadores de HTML, CSS, etc. y un diagnóstico de cuán accesible son. Además de analizar si las funcionalidades que proponemos para la extensión podrían ayudar a mejorar el acceso a la información.
Páginas mas utilizadas



Gráfico 3.6. Resultados de la encuesta sobre las páginas web más visitadas.

De esta pregunta reunimos aquellas páginas más visitadas por los usuarios y un perfil más detallado de las tareas que realizan cotidianamente. Según el gráfico 3.6 el mayor porcentaje es para el uso de las redes sociales y los diarios pero también podemos observar muchas páginas gubernamentales y no gubernamentales del país donde se realizan trámites o gestiones que son tareas indispensables para las personas. Otra de las páginas indispensables que se pueden observar en el gráfico son aquellas que se utilizan para los estudios, como las páginas universitarias ya sea de universidades nacionales como institutos privados.

 Sexta pregunta: "¿A qué páginas no podes acceder y por qué razón?" Esta pregunta nos ayuda a ver los problemas con los que se enfrentan los diferentes usuarios y tener otro punto de vista de la verdadera situación de accesibilidad en la web.

De esta pregunta obtuvimos respuestas diversas de las cuales podemos concluir que los sitios más inaccesibles son los diarios porque la gran mayoría visualizan muchas publicidades, pop-ups, formularios y/o elementos captcha, también poseen mucho contenido y se dificulta el acceso por estar mal organizado,

además de contener imágenes que entorpecen a la hora de usar lectores de pantalla ya que en muchos casos no llevan descripción asociada en el atributo alt.

Séptima pregunta: "¿Qué página le resulta fácil de usar?" De las páginas reunidas podemos comparar la usabilidad, como están dispuestos los elementos, el feedback, los colores, la fuente que se usa, qué componentes están más a la vista, en qué lugar se ubican, qué atajos tienen, qué ayuda brindan, si tienen tutoriales, si tienen la seccion de preguntas frecuentes, etc. Con la finalidad de saber que diseños les resultan más favorables a los usuarios.

De esta pregunta obtuvimos respuestas muy variadas de las cuales podemos concluir que las páginas más fáciles de usar son las redes sociales, estos sitios presentan un alto grado de madurez en cuanto a experiencia de usuarios y usabilidad, lo que produce una mejora constante del diseño que resulta ser sencillo y bien organizado, por lo cual es fácil aprender a usarlas.

 Octava pregunta: "Marque qué funcionalidades le resultaría útil" propusimos un listado de funcionalidades para buscar cuales son más necesarias y en base a eso organizarnos para dar hincapié a las más elegidas.



Funcionalidades elegidas

Gráfico 3.7. Resultados de la encuesta sobre las funcionalidades que los usuarios de la web desean utilizar.

Observando el gráfico 3.7 podemos notar que todas las opciones fueron elegidas y dentro de las más relevantes están quitar contenido extra, como imágenes, publicidades, multimedia que dificulta el acceso al contenido principal de la página, evitar interrupciones como pop ups, captcha o formularios de acceso y cambios de contraste para aquellas páginas que es difícil distinguir el texto.

Conclusión del capítulo

En síntesis, en este capítulo exponemos los errores de accesibilidad tanto desde el análisis de los validadores como desde la experiencia del usuario. La mayoría de las herramientas de validación analizan las páginas web basándose en las pautas de la WCAG y generan un informe detallado con los errores y advertencias encontrados. Aunque algunas herramientas pueden guiar al usuario (ya sea programador o diseñador web) en el proceso de corrección de errores, no realizan cambios en el código. Por otro lado, con el cuestionario realizado, recopilamos los problemas de accesibilidad que más afectan a los usuarios, así como las funcionalidades consideradas útiles para solucionar dichas problemáticas.

Capítulo 4

Aplicaciones de software sobre accesibilidad

En el capítulo anterior, realizamos un análisis detallado sobre la accesibilidad web y llevamos a cabo una encuesta dirigida a usuarios, con o sin discapacidades, con el objetivo de identificar los problemas más comunes y extraer las soluciones más útiles de las herramientas de validación vistas, además de las funcionalidades propuestas dentro del cuestionario.

En este capítulo analizamos complementos, aplicaciones, sistemas operativos y páginas web que resuelven problemas de accesibilidad web y que cumplen con algunas pautas de la WCAG para recopilar las funcionalidades más relevantes y necesarias para el usuario.

Complementos

Se analizaron diferentes herramientas que tratan temas relacionados a la accesibilidad, brindando asistencia y apoyo a cualquier usuario. Algunas de estas herramientas son aplicaciones web, de escritorio o móviles. Para el caso de las aplicaciones web, el análisis e investigación se centró en el navegador web Google Chrome, precisamente en relación a una de sus características, las extensiones.

Entre las herramientas más interesantes encontrados para su análisis e investigación, podemos destacar:

High Contrast

Es una extensión de Google Chrome que cambia o invierte la combinación de colores para que sea más fácil leer las páginas web, se aplica desde la barra de navegación en la sección de las extensiones a partir de un botón on/off.

Los sitios con bajo contraste pueden ser difíciles de leer para las personas con baja visión. Algunos sitios mal diseñados de la web tienen malas combinaciones de color, tales como enlaces de color azul sobre fondo negro, texto en color rojo sobre fondo verde, u otras combinaciones que no son fáciles a los ojos de cualquiera, pero especialmente difíciles para las personas con baja visión. Esta extension permite navegar por la web con su elección de varios filtros de color de alto contraste diseñados para facilitar la lectura del texto y evitar la fatiga ocular. Como funcionalidades disponibles, tiene habilitar/deshabilitar contraste alto, seleccionar un esquema de color (los esquemas disponibles son modo normal, contraste aumentado, escala de grises, color invertido , escala de grises invertida y amarillo sobre negro), establecer esquema de color por defecto y olvidar todas las personalizaciones de todos los sitios. Una de las cualidades que tiene la aplicación es que podemos elegir un contraste por defecto que se aplica a las nuevas pestañas, e independientemente de esto, podemos elegir un contraste exclusivo para una página en específico, este cambio queda guardado y no se pierde al recargar la página. Lamentablemente la herramienta no funciona en algunos sitios web como por ejemplo, Google Docs o el Chrome Web Store. Otro inconveniente es que también cambia los colores de las imágenes y los videos lo que resulta incómodo, estaría bueno tener una opción adicional para que los contrastes se apliquen o no para imágenes y videos [32]. Cumple con la pauta 1.4.3 de nivel AA y la pauta 1.4.6 de nivel AAA de la WCAG.

En la siguiente imagen podemos observar como queda aplicado el contraste color invertido en una página web.



Imagen 4.1. Menú de configuraciones de la extensión y ejemplo de página web con un filtro de alto contraste aplicado.

Font Size Increase & Decrease

Son extensiones de Google Chrome que aumenta o disminuye el tamaño de la fuente de cada elemento con tan solo un clic del mouse, a diferencia del zoom que se realiza en el navegador para ampliar la página. Se aplica desde la barra de herramientas del navegador con el botón A+ o el botón A-, que mediante los símbolos indica el aumento o disminución del texto. Lamentablemente dichas extensiones no funcionan en

muchos sitios como por ejemplo, WordPress, Outlook, Google Docs, etc. También se observó que no funciona en todos los elementos HTML, por ejemplo con los elementos span. Si se quiere restaurar el tamaño original de la fuente las únicas opciones son recargar la página o tener ambas extensiones lo que sería impreciso porque no hay manera de saber el valor original y la cantidad de veces que se aumento o disminuyo, estaría bueno que tenga una opción al menos en las opciones de la extensión [33][34]. Esta extensión cumple con la pauta 1.4.4 de nivel AA de WCAG.

La siguiente imagen a modo de ejemplo muestra cómo quedan los textos luego de aumentar o disminuir los tamaños de fuente.



Imagen 4.2. Ejemplos de aumentar y disminuir el tamaño de la fuente con las herramientas Font Size Increase & Decrease.

A+ cambiar de tamaño de texto

Es una extensión de Google Chrome y Microsoft Edge que permite al usuario personalizar el tamaño, la fuente y el color del texto, el color de fondo y la familia de fuentes en una página. Para el cambio del tamaño del texto, funciona similar a las extensiones vistas anteriormente, no realiza la acción zoom del navegador sobre la página. La herramienta ofrece 2 versiones, versión de prueba y versión paga, las funcionalidades mencionadas corresponden a la versión de prueba, para acceder a la versión paga se tiene que pagar US\$9.99. En la versión paga se desbloquean nuevas funcionalidades como atajos para aumentar/disminuir el tamaño del texto y la posibilidad de guardar las configuraciones hechas. Para utilizarla desde la barra de herramientas se

despliega un menú con todas las configuraciones que brinda la extensión. Una de las desventajas es que los cambios no se aplican a todas páginas ni tampoco quedan guardados al recargar y es necesario volver a configurar [35].

Esta extensión cumple con las pautas 1.4.1 de nivel A, 1.4.3 de nivel AA, 1.4.4 de nivel AA, 1.4.6 de nivel AAA de la WCAG.

La siguiente imagen muestra todas las funcionalidades disponibles con la versión paga de de la herramienta.



Imagen 4.3. Configuraciones de la herramienta A+ cambiador de tamaño de texto.

OpenDyslexic para Chrome

Open-Dyslexic es una fuente de código abierto creada para aumentar la legibilidad de los lectores con dislexia. Esta extensión anula todas las fuentes en las páginas web y las reemplaza con la fuente OpenDyslexic, también da formato a las páginas para que sean más fáciles de leer.

La fuente ayuda a reconocer correctamente las letras y las formas únicas de cada letra pueden ayudar a evitar confusiones al voltearlas e intercambiarlas o evitar que el cerebro las gire. Se puede activar mediante un botón on/off desde la barra de herramientas del navegador. También tiene la posibilidad de elegir las variantes de la fuente Open Dyslexic (regular, bold e italic), las cuales, junto con el botón mencionado, se pueden observar en la imagen 4.4. La configuración queda guardada y se aplica a la página en la que estas y todas las nuevas pestañas, en algunos casos, para que se aplique el cambio de la fuente es necesario recargar [<u>36</u>].



Imagen 4.4. Página web con la fuente OpenDyslexic aplicada mediante la herramienta para Chrome.

Ocultar imágenes

Oculta imágenes, imágenes de fondos, videos multimedia, elementos flash en cualquier página para mayor privacidad y legibilidad. Esta herramienta ayuda con varias discapacidades cognitivas en las que las imágenes suelen distraer, además ayuda a la lectura para las personas con visión disminuida. Una de las desventajas es que no puede ocultar las imágenes que están dentro de los elementos iframes, además hay que configurarlo en cada página y no guarda los cambios al recargar [<u>37</u>].

La siguiente imagen muestra el antes y el después de ocultar las imágenes con la herramienta.



Imagen 4.5. Ejemplo de pantalla con y sin imágenes usando la extensión de Ocultar imágenes.

ColorBlind - Dalton for Google Chrome

Dalton es un software que permite a las personas con diferentes tipos de daltonismo ver más colores o notar mejor la diferencia. La extensión te da la posibilidad de elegir entre los tres grupos de colores para los diferentes tipos de daltonismo (tritanopia, deuteranopia o protanopia), además permite modificar la intensidad del color y ajustar los colores. Estas funcionalidades solo cambian algunos elementos HTML y las imágenes, no afecta a los videos. Una de las ventajas es que queda guardada la configuración que hayas hecho en cada sitio web y no se modifica al recargar [38]. Cumple con la pauta 1.4.3 de nivel AA y la pauta 1.4.6 de nivel AAA de la WCAG.

La siguiente imagen muestra como quedaria aplicado en la pagina web el grupo de color para el tipo de daltonismo protanopia.



Imagen 4.6. Filtro aplicado de un tipo de daltonismo.

Aplicaciones con funciones de accesibilidad

También, en esta etapa de exploración fue necesario observar las funciones de accesibilidad provistas por distintas aplicaciones de software. Para ello, analizamos 3 ejemplos de aplicaciones como un visualizador de pdf, un gestor de videoconferencias y un reproductor de videos que presentan funciones de accesibilidad.

Adobe Acrobat Reader

La aplicación posee un asistente de configuración de accesibilidad que proporciona instrucciones en pantalla para configurar estas preferencias donde se pueden activar varias opciones [39]. Dentro de estas opciones están:

- Ampliadores de pantalla: es posible activar la lupa para ver en el tamaño deseado lo que se presenta, textos, imágenes, videos, etc.
- Configuración del contraste en los documentos: brinda diferentes opciones para visualizar el texto en los documentos como también da otras opciones de colores para los resaltados de texto.
- **Suavizar el texto**: es posible suavizar o redondear los bordes definidos de una selección con el suavizado de texto.

• Activar el lector de pantalla: estas preferencias permiten controlar cómo aparecen los archivos PDF en la pantalla y cómo los lee el lector de pantalla.

En la siguiente imagen se visualizan algunas de las opciones anteriormente mencionadas.

		_					
	Ayudante de configuración de accesibilidad, Pantalla 2 de 5 X						
4	Adobe Acrobat Reader puede anular los colores del texto del documento y mostrar éste con colores de alto contraste para facilitar su lectura. No active esta opción si usa una configuración de alto contraste en el sistema o en la tecnología de soporte: pueden surgir interferencias entre ambos que producirían resultados imprevistos.						
Re	Usar colores de alto contraste para el texto del documento						
lo	Combinación de colores de alto contraste: Texto verde sobre negro 🗸 🗸						
	Puede establecer los colores usados para resaltar los campos de formulario y para indicar los campos obligatorios. Si se cambian los colores usados en los documentos, puede ser conveniente personalizar estos colores también.						
	Color de resaltado de campo: Azul claro 🗸						
K	Color de resaltado para campo obligatorio: Rojo 🗸						
C	El suavizado de texto se puede desactivar durante la visualización para proporcionar texto de mayor contraste.						
P	Desactivar suavizado de texto						
	Si desea que el cursor de selección del teclado esté activo en el documento, especifique que Adobe Acrobat Reader siempre debe usar el cursor de selección del teclado.						
	Mostrar siempre el cursor de selección de teclado						
	Cancelar Anterior Siguiente						
6	6 Certificados						

Imagen 4.7. Asistente de configuración de accesibilidad de Adobe Acrobat Reader.

Zoom

La herramienta ofrece en su menú de configuración una sección accesibilidad donde se puede adaptar la interfaz a tus necesidades [40].

En esta configuración, se pueden adaptar los subtítulos modificando el tamaño de la fuente, el color del texto y el color de la traducción. También es posible atenuar el video de la pantalla compartida para evitar imágenes intermitentes o patrones visuales, como rayas en la imagen. Además, se puede aumentar o disminuir el tamaño del chat y agregar alertas de eventos que ocurren durante la llamada, como cuando el anfitrión detiene el video o silencia el audio. Esta configuración se puede observar en la imagen 4.8.

en cor	lingulacion	/	\sim
۵	actualice a Zoom Workplace Pro para	obtener reuniones ilimitadas, aplicaciones de productividad y más. Actualizar ahora	
Θ	General	Subtítulos	
0	Vídeo	Tamaño de fuente: (14)	
\bigcirc	Audio		
Ŷ	Compartir pantalla	Color de subtítulos:	
0	Chat en equipo	Color de la traducción: 🔵 🛑 🛑 🔵 🌍 🥱	
3	Zoom Apps		
	Fondo y efectos	Los subtítulos ocultos serán así Las traducciones serán así	
\bigcirc	Grabación		
8	Perfil	Mostrar subtítulos siempre ③	
	Estadísticas		
	Métodos abreviados de teclado	Atenuar vídeo de pantalla compartida Atenuar automáticamente el vídeo cuando se detecten imágenes intermitentes	
Ť	Accesibilidad	 o patrones visuales (como rayas). 	
		Tamaño de visualización del chat (Ctrl+/-)	
		Alertas del lector de pantallas Restablecer los valores predeterminados	

Imagen 4.8. Configuración de accesibilidad de Zoom.

VLC

La aplicación brinda diferentes configuraciones de accesibilidad, así como también permite agregar extensiones o complementos para una mayor accesibilidad [41].

La configuración de accesibilidad está disponible desde el menú de preferencias donde brinda diferentes instrucciones para cada parte de reproductor:

- Agregar subtítulos
- Configurar el audio
- Configurar la visualización de los videos
- Personalizar los atajos del teclado
- Agregar codecs propios
- Configurar la interfaz para mayor comodidad del usuario, reordenando los elementos en la pantalla o configurando la visibilidad de los elementos.

 También se destaca que es posible la navegación secuencial por los elementos del menú.

Algunas de las instrucciones mencionadas se pueden observar en la siguiente imagen, en este caso instrucciones orientadas en subtítulos y ajustes sobre pantalla.

Preferencias s	imples				_		2
4 Interfaz	Audio	Vídeo	Subtítulos / OSD Entra	da / Códecs Atajos	de teclado		
referencia	as de subtítu	ılos y aj	ustes sobre pant	alla			
Mostrar en par	ntalla						
🗹 Habilitar vis	sualización en pantall	a (OSD)					
Mostrar títi	ulo del medio al iniciar	el vídeo	Posición Abajo			•	•
✓ Habilitar subt	ítulos						
Idiomas de sub	títulos						
Idioma de subt	títulos preferido						
Codificación pr	edeterminada		Predeterminado (Windo	ows-1252)			•
Efectos de sub	títulos						
Fuente			Arial			```	7
Tamaño de fue	ente		Automático 🔹	Color de texto p	redeterminad	do	
Grosor del boro	de		Normal	Color del borde			
🖂 Añadir una	sombra		Añadir un fondo				
Forzar posición	n de subtítulos		0 px				
Mostrar aiuster	5						
i lood ar ajabtet				_			

Imagen 4.8. Configuración de las preferencias de VLC.

Funciones de configuración en sistemas operativos

Los sistemas operativos tanto móviles como de escritorio incluyen también funciones de accesibilidad para adaptarse ante requerimientos y necesidades de la persona usuaria. Comentamos dos ejemplos, Windows y Linux.

Accesibilidad en Windows

En la configuración del sistema operativo se ofrecen distintas opciones de accesibilidad relacionadas a visión, audición e interacción. Las más destacables son la configuración del contraste, la lupa, los filtros de colores (como se muestra en la imagen 4.9), la navegación por voz y el aumento del tamaño de la fuente. Para algunas de estas funcionalidades es necesario un complemento tecnológico, por ejemplo, para la navegación por control ocular es necesario conectar un sensor de movimiento ocular.

Configuración	
ය Inicio	Filtros de color
Buscar una configuración ${\cal P}$	Haz que las fotos y los colores sean más fáciles de ver al aplicar un filtro de color a la pantalla.
Accesibilidad	
	Usar filtros de color
Visión	Activar filtros de color
⊕ Lupa	Activado
Filtros de color	Permitir que la tecla de método abreviado active o desactive el filtro
.☆ Contraste alto	Presione la tecla del logotipo de Windows 배 + CTRL + C para activar o desactivar el filtro.
오 Narrador	
	Selecciona un filtro de color para ver mejor los elementos en la pantalla
Audición	O Invertido (invertir colores de la pantalla)
외까 Audio	O Escala de grises
🖾 Subtítulos	Escala de grises invertida
Interacción	O bien, selecciona un filtro de daltonismo para hacer que los nueve colores de la rueda sean más distintos.
🖵 Voz	Rojo y verde (verde suave, deuteranopía)

Imagen 4.9. Ejemplo de accesibilidad en windows 10 utilizando la opcion de filtros de color.

Accesibilidad en Linux

Dentro de la configuración del sistema operativo Linux, también se ofrecen distintas opciones de accesibilidad relacionadas a visión, audición e interacción. Las herramientas más destacables que incluye Linux son:

- Orca: un lector de pantalla que lee en voz alta el texto en la pantalla.
- Magnificador: una herramienta que amplía la pantalla para facilitar la lectura.

- Teclado en pantalla: una herramienta que permite a los usuarios escribir utilizando un teclado virtual en la pantalla.
- Contraste alto: una opción que aumenta el contraste de la pantalla para facilitar la lectura.
- Subtítulos y descripciones de audio: opciones que permiten a los usuarios con discapacidades auditivas seguir el contenido de los videos y las presentaciones.
- Control por voz: una herramienta que permite a los usuarios controlar el sistema operativo y las aplicaciones utilizando comandos de voz.

Además, muchas distribuciones de Linux ofrecen opciones de personalización para adaptar el sistema operativo a las necesidades específicas de cada usuario.

Páginas Web con opciones de accesibilidad

Las funciones de accesibilidad también pueden disponerse dentro de los propios sitios web, como explicamos en los siguientes ejemplos.

Banco Provincia

Esta página web utiliza una herramienta de accesibilidad llamada InSuit, que, a través de un botón en pantalla (como se muestra en la imagen 4.10), permite desplegar la navegación asistida por voz. Además, ofrece atajos de teclado para facilitar el desplazamiento entre los diferentes componentes de la página.



Imagen 4.10.Página web de la herramienta InSuit [42] .

Al cliquear sobre el botón herramientas de accesibilidad se reproduce un mensaje de bienvenida y se visualiza un tutorial (como se muestra en la imagen 4.11) para aprender a utilizar el menú de navegación mediante el uso de las teclas de función, entre otras. Como ayuda al usuario tiene un menú donde agrupa en categorías los distintos tipos de navegación disponible, navegación web, multimedia, pdf y formularios.



Imagen 4.11. Página del Banco Nación con el menu de navegacion InSuit.

Es una herramienta muy útil para la navegación por teclado que además integra un lector de pantalla pero tiene el inconveniente que es una herramienta de pago y que solo funciona en las páginas web que tengan activado el servicio, lo cual es decisión del propietario de la página, además, funciona con las tres últimas versiones de la mayoría de navegadores. A nivel de accesibilidad HTML mejora la página web respecto a las pautas de la WCAG 2.1, incluso sin necesidad de activar la navegación accesible que ofrece la herramienta. Como última observación, tanto la activación como la configuración de inSuit se mantiene al navegar entre las páginas que ofrecen el servicio.

Titicom

Esta página web utiliza una herramienta de accesibilidad que ofrece diversas funcionalidades útiles para diferentes usuarios. Al hacer clic en el botón con el símbolo de accesibilidad universal, se despliega un menú lateral (como se observa en la imagen 4.12) que cuenta principalmente con cuatro opciones: ajuste del color, ajuste de texto, asistente de contenido y dispositivos compatibles. Los cambios realizados en cualquiera de estas cuatro opciones pueden guardarse, permitiendo mantener las configuraciones elegidas, o bien restablecerse a los valores predeterminados. La herramienta también incluye opciones de internacionalización de idiomas, así como configuraciones adicionales, como cambiar el tema a oscuro o mover la barra de derecha a izquierda.



Imagen 4.12. Página web de titicom, empresa argentina de telecomunicaciones con el menú de la herramienta User Accessibility[43].

Esta herramienta es similar a la anterior en este sentido, que también es de pago y solo está disponible en las páginas web que han habilitado el servicio. La decisión de activarla depende del propietario de cada sitio.

Es una herramienta bastante completa ya que dispone de un amplio repertorio de ajustes y funcionalidades, el menú está bien estructurado y ubicado, es fácil de entender, se puede ajustar el sitio a tus preferencias y guardarlo para una visita posterior. En la experiencia utilizando la herramienta encontramos varios errores en algunas de sus funcionalidades, los mensajes de error son de un idioma distinto (hebreo) al elegido en la herramienta por lo que se dificulta entender qué está pasando. Uno de los errores que encontramos más notorio es que al querer escribir símbolos con el teclado virtual devuelve un bloque de código *span* que no se entiende para el usuario común.

ADC Asociación por los Derechos Civiles

Este sitio web brinda la posibilidad de aumentar o disminuir el tamaño de la fuente de cada sección mediante dos botones claramente identificados: A+ para aumentar y A-para disminuir, cumpliendo con la pauta 1.4.4 de nivel AA de la WCAG. [44] En la siguiente se muestra cómo estos botones se encuentran ubicados en la interfaz.



Imagen 4.13. Aumento de fuente integrado en la página web accesibilidad-digital.adc.org.ar

También ofrece la posibilidad de activar subtítulos en inglés y español, además de utilizar lenguajes de señas específicos para Argentina, Chile y Uruguay en los videos, cumpliendo con la pauta 1.2.2 de nivel A de la WCAG. En la siguiente imagen se muestra un video con la opción de cambiar entre los diferentes lenguajes de señas disponibles.



Imagen 4.14. Opciones de subtítulos, lenguajes de señas en los videos de la página web accesibilidad-digital.adc.org.ar

Tesinas de grado sobre accesibilidad

En el presente trabajo se analizan 3 tesinas vinculadas al tema central de esta investigación, accesibilidad web, con el objetivo de explorar diversas aproximaciones y enfoques que abordan esta problemática desde distintos ángulos. A través de su análisis, se busca comprender las funcionalidades provistas, las conclusiones alcanzadas y las implicaciones que tienen en la mejora de la accesibilidad en diferentes contextos.

Transita Web

Transita Web es un sitio web que en la actualidad no se encuentra activo en internet, es una herramienta que accesibiliza sitios web para que las personas que poseen alguna discapacidad puedan consultarlos de forma simple y rápida. La misma adapta el sitio a mostrar en base a la configuración realizada, aplicando reglas de accesibilidad y visualizando la información de forma accesible [45]. Dentro de las configuraciones disponibles de la herramienta tenemos:

- Accesibilizar las páginas web implementadas con código HTML5 aplicando las normas WCAG 2.0.
- Descartar las publicidades.
- Ocultar los logos de las redes sociales.
- Elegir la ubicación de los elementos del sitio, según la necesidad del usuario. Colocar el menú arriba, abajo o a los costados y, por consecuencia, reagrupar los contenidos principales del sitio web.
- Visualizar las páginas con alto contraste.
- Sacar o dejar las imágenes.
- Seleccionar el tamaño de letra con el que se desea mostrar la información, letra chica, mediana o grande.

La herramienta está centrada en discapacidades visuales ya que el contenido de las páginas web se accesibiliza de acuerdo a pautas de la WCAG 2.0 como por ejemplo cambiar el tamaño de fuente y aumentar el contraste, también se elimina el contenido innecesario e irrelevante de la página como imágenes y publicidades lo que favorece el procesamiento por parte de los lectores de pantalla.

HandsFree for web

HandsFree for web es una herramienta que le permite al usuario navegar por la web exclusivamente invocando comandos de voz que está actualmente en funcionamiento [46]. Dentro de las funcionalidades que se pueden realizar utilizando los comandos de voz están:

- **Navegar la web**: abrir sitios web, realizar clics y desplazarse verticalmente dentro de los sitios.
- **Completar formularios**: dictar textos, completar todo tipo de campos, copiar y pegar.
- Uso del navegador: gestión de favoritos, descargas, historial y pestañas.
- Controlar contenido multimedia: reproducir videos y audios
- Buscar en internet: buscar en Google y texto dentro de las páginas web.

Este modelo de interacción web dirigido por comandos de voz asiste a usuarios con y sin discapacidades motrices.

SiMor

SiMor, acrónimo de las palabras en inglés See (ver) y More (más), actualmente inactiva, es una herramienta de análisis de accesibilidad web, que se basa en un enfoque de análisis intensivo basado en reglas siguiendo las especificaciones de las WCAG 2.0 de la W3C [47]. Según su especificación, cada regla en SiMor corresponde a una pauta específica de las WCAG y define las condiciones para su cumplimiento, además de sugerir soluciones posibles en caso de que se detecten problemas. Provee de un conjunto de reglas específicas para evaluar la accesibilidad, lo que lo hace extensible ante actualizaciones o cambios en las WCAG. Esto le permite adaptarse fácilmente a nuevas especificaciones o pautas.

Para realizar un análisis continuo de la accesibilidad, SiMor necesita que los usuarios creen una cuenta de usuario. Esto posibilita hacer un seguimiento del progreso a lo largo del tiempo, ver comparaciones entre validaciones previas y nuevas, e identificar las causas de las violaciones de las reglas. Además, el sistema de versionado asegura que las validaciones no sean repetitivas, sino parte de un proceso evolutivo.

Cada vez que se valida un sitio web, se puede realizar un crawling (recorrido de todos los enlaces del sitio) y obtener los códigos HTML correspondientes en segundo plano. Esto permite una evaluación completa de todas las páginas vinculadas. También permite utilizar wildcards o comodines en expresiones regulares, lo que facilita la exclusión de enlaces o páginas que no se desean validar. Esta funcionalidad mejora la flexibilidad y control sobre el análisis.

Conclusión del capítulo

Después de analizar los puntos clave de cada software investigado en este capítulo, hemos identificado muchas funcionalidades útiles para el objetivo de esta tesis. Cada una de las prestaciones está orientada a resolver una problemática específica, pero las mismas pueden encontrarse en forma descentralizadas. Algunas de las herramientas con más funcionalidades integradas requieren una suscripción paga, lo que dificulta el acceso para el usuario. Otros inconvenientes que encontramos incluyen que algunas herramientas no guardan la configuración para futuras interacciones, mientras que otras requieren recargar la página para restaurar la configuración inicial. Además, algunas herramientas aplican cambios en elementos que no deberían o, en algunos casos, no aplican los cambios en forma completa, lo que genera confusión en el usuario.

A partir de lo investigado en los capítulos anteriores tomaremos en cuenta para el desarrollo de AccesaWeb las funcionalidades más relevantes vistas hasta este capítulo,

logrando que cumplan con el estándar de la WCAG mencionadas en el capítulo 2 y que den solución a los problemas más comunes de accesibilidad.

Capítulo 5

Desarrollo de la extensión AccesaWeb

En el capítulo anterior analizamos complementos del navegador, aplicaciones, sistemas operativos y páginas web que brindan accesibilidad con el objetivo de observar las funcionalidades más relevantes y características útiles de accesibilidad con sus pros y contras.

En este capítulo hacemos una introducción a nuestra herramienta bautizada como AccesaWeb, explicaremos la arquitectura sobre la cual está montada, las decisiones de diseño y el proceso de desarrollo llevado a cabo para lograr el objetivo de la tesina.

¿Qué es AccesaWeb?

Como mencionamos anteriormente, hemos investigado diversas herramientas que brindan accesibilidad teniendo en cuenta sus potencialidades como sus inconvenientes. Algunas de ellas son gratuitas, pero las opciones más útiles suelen estar en versiones de pago, lo que representa un impedimento para muchos usuarios. Además, hemos notado mediante ejemplos que no todas las herramientas funcionan correctamente en algunas páginas web, como el caso donde el contraste solicitado no se aplicaba a todos los elementos; algunas no ofrecen la posibilidad de guardar las configuraciones solicitadas por la persona con discapacidad, lo que provoca que se pierdan los cambios al recargar la página. Entre otros aspectos a considerar, estas herramientas no ofrecen la posibilidad de accesibilizar la página que se está visitando, cumpliendo con las pautas de la WCAG. En AccesaWeb se intenta solucionar estos inconvenientes incorporando e integrando mejoras de accesibilidad.

Además de considerar estos hallazgos en la investigación, se tuvo en cuenta también la encuesta de usuarios realizada que fue una fuente de información e inspiración para reunir, integrar y desarrollar las opciones de accesibilidad necesarias.

Entonces, AccesaWeb nace como una herramienta para brindar asistencia a personas con discapacidades diversas y limitaciones de interacción. Integra las funcionalidades vistas más útiles y relevantes para el usuario con o sin discapacidad, de manera totalmente gratuita brindando accesibilidad en una sola herramienta mucho más completa y eficaz. Además, Implementa y cumple con las pautas de WCAG, 2.0 y algunas más de la WCAG 2.1, asegurando un estándar alto de accesibilidad.

Para el desarrollo de AccesaWeb, es fundamental que la herramienta sea estable, segura, confiable y multiplataforma. Además, debe contar con un amplio soporte y documentación, no requerir una licencia comercial de pago y ser apta para un uso masivo. Por estas razones, decidimos crear una extensión para el navegador Google Chrome, ya que cumple con todas estas características. Además, Chrome es considerado uno de los

navegadores más utilizados en el mercado¹. Es de última generación, gratuito y permite una fácil distribución e instalación, lo que favorece su adopción masiva.

A continuación detallamos las decisiones de diseño y cada etapa del desarrollo, los aspectos funcionales y de implementación de AccesaWeb.

Implementación de AccesaWeb

AccesaWeb es una extensión para Google Chrome, su núcleo está basado en la plataforma manifest V2, a través de la cual, se puede programar pequeñas aplicaciones que modifican el navegador y las páginas web, permiten a los usuarios adaptar la funcionalidad y el comportamiento del navegador a las necesidades del usuario.

Las extensiones están hechas de diferentes componentes descritos en el archivo manifest.json, estos pueden incluir código para el contenido de las páginas, código de background, código para la página de opciones, elementos de la interfaz de usuario y varios archivos lógicos. Los componentes se crean con tecnologías de desarrollo web: HTML, CSS y JavaScript y según las funcionalidades de la extensión es posible que no se requiera el uso de todas las opciones. El registro de los componentes en el archivo manifest le dice a la extensión a qué archivo hacer referencia y cómo debe comportarse ese archivo.

A continuación, se presenta un ejemplo de archivo manifest.json basado en las especificaciones de Manifest V2.

```
{
"name": "Example",
"version": "1.0",
"description": "Extension",
"manifest version": 2,
"background": {
   "scripts": ["background.js"]
},
 "options page": {
   "scripts": "docs.html"
},
 "content scripts": {
   "css": ["body.css"],
   "js": ["content.js"],
}
}
```

Sección de código 5.1. Ejemplo de un archivo manifest.json simple

1

https://es.statista.com/estadisticas/600249/cuota-de-mercado-mensual-de-los-principales-navegadores-deinternet/

En este archivo se especifican los componentes básicos de la especificación Manifest V2 tales como el script de background, el script content que se ejecuta por cada pestaña en su contexto asociado, estilos CSS, HTML de la página de opciones.

Alcance de la extensión

Basándonos en esta información, la extensión se encargará de garantizar los principios de las pautas WCAG accesibilizando en lo posible, la página visitada y brindando una serie de opciones de configuración para que la persona usuaria pueda adaptar la misma. Como objetivo principal deberá lograr que los elementos de las páginas web puedan ser percibidos por los usuarios de manera sencilla. El contenido será mostrado en diferentes formas sin perder información o estructura. Asimismo, ofrecerá al usuario la posibilidad de guardar la configuración que desea para adaptar el navegador según sus necesidades en los posteriores accesos.

Arquitectura

La aplicación es una extensión del navegador Google Chrome de modalidad add/on, es decir, que se agrega al navegador para incrementar sus funcionalidades básicas. Se ejecuta del lado del cliente y se instala desde la Chrome Web Store, es gratuita y multiplataforma.

Se presenta la siguiente arquitectura:



Figura 5.1. Ejemplo del funcionamiento de la arquitectura de AccesaWeb al cambiar una configuración.

En la figura anterior, al realizar un cambio en la configuración, la extensión procesa esta modificación mediante las tecnologías correspondientes. La información se

guarda en el almacenamiento del navegador y los cambios se reflejan de inmediato en las páginas web.



Figura 5.2. Ejemplo del funcionamiento de la arquitectura de AccesaWeb al emitir un comando de voz.

En esta figura, el usuario emite un comando de voz, la extensión recibe los datos sonoros y los envía al servicio de reconocimiento de voz, Web Speech API. Allí se procesa el flujo de datos y se retorna la transcripción a texto; y por último, la extensión hace uso del resultado para ejecutar el comando correspondiente.

Tecnologías utilizadas

Además de la propia plataforma, Manifest V2, se utilizaron tecnologías de código abierto que serán descriptas a lo largo de esta sección.

HTML5

Es un lenguaje de etiquetas, utilizado para la estructuración y la presentación de contenido en los sitios web [48].

Como se muestra en la sección de código 5.2 la etiqueta especial DOCTYPE con el valor html indica que el documento será HTML5. La estructura básica cuenta con un header y un body dentro de los cuales se ubican las diferentes etiquetas.

```
<!DOCTYPE html>
<html>
<head>
    <title>Page Title</title>
    </head>
    <body>
        <h1>This is a Heading</h1>
        This is a paragraph.
     </body>
</html>
```

Sección de código 5.2. Estructura de HTML simple.

WAI-ARIA

El código HTML5 de las páginas web se puede accesibilizar mediante atributos, propiedades y roles provistos por la WAI-ARIA (Web Accessibility Initiative - Accesible Rich Internet Applications). WAI-ARIA es una especificación de la W3C que define como hacer que el contenido web y las aplicaciones web sean más accesibles para las personas con discapacidades. Está pensado para hacer más accesible el contenido dinámico como AJAX, JavaScript y sus tecnologías relacionadas. Sin WAI-ARIA, ciertas funciones utilizadas en los sitios web no están disponibles para algunos usuarios con discapacidades, especialmente las personas que dependen de los lectores de pantalla y las personas que no pueden usar un mouse. WAI-ARIA aborda estos desafíos de accesibilidad, por ejemplo, definiendo formas de proporcionar funcionalidad a la tecnología de asistencia. Algunos de los principales atributos y propiedades de WAI-ARIA son:

Tabindex: en HTML sólo pueden recibir el foco del teclado los enlaces, botones y
elementos de formulario. Por el contrario los elementos HTML tales como listas,
párrafos, capas div o span, etc nunca recibirán el foco del teclado al tratarse de
elementos con un propósito diferente, por lo que podemos utilizar el atributo
tabindex para que estos elementos puedan recibir el foco y por tanto ejecutar
acciones dinámicas, por ejemplo el evento onclick.

El siguiente código visualiza un ejemplo de uso de tabindex sobre una lista:

```
        tabindex="3">Elemento 3
        tabindex="1">Elemento 1
        tabindex="2">Elemento 2
```

Sección de código 5.3. Ejemplo de lista ordenada con atributo tabindex para cada ítem. Al utilizar la tecla Tab, el navegador enfoca los elementos de la lista según el orden especificado por el atributo tabindex, en lugar de seguir el orden en que estos aparecen en el código HTML.

Roles: proporcionan significado semántico al contenido de manera que las tecnologías de asistencia, como los lectores de pantalla, puedan interpretar y comunicar correctamente la función y el propósito de cada elemento a los usuarios con discapacidades. El rol o función de un elemento de nuestra página mediante ARIA es tan sencillo como añadir a la etiqueta del elemento role="[nombre del rol]". El siguiente código es un ejemplo sencillo de uso de role:

```
<div role="button" tabindex="0">
Haz clic aquí
</div>
```

Sección de código 5.4. Ejemplo de uso de role ARIA.

En este caso, se define que el elemento <div> se comporte como un botón, y el lector de pantalla lo interpretará de esa manera.

Existen dos tipos de roles, aquellos que definen elementos de la interfaz (árboles, alertas, sliders, etc.) y los que definen la estructura de la página (cabecera, navegación, pie). Dentro de los que definen la estructura de la página se distinguen los landmark roles, que se usan para identificar áreas separadas de la página y transmitir la naturaleza de las mismas.

Los roles de estructura y su función son:

- role="banner". Sólo debe existir uno en la página y es para marcar generalmente la cabecera de la página web donde se encuentra el título, logo, etc.
- role="navigation". Se debe utilizar para marcar los elementos que forman un menú, por ejemplo una lista ul.
- role="main". Marca dónde está el contenido principal de la página.
- role="complementary". Contenidos que no son imprescindibles el contenido, por ejemplo un slider, barra lateral con publicidad, etc.
- role="contentinfo". Para marcar una zona en la que hay información sobre la página y que se repite en todas, por ejemplo la información legal, enlaces de navegación de un sitio y demás información que generalmente se ubica en la parte inferior de la página web.
- o role="search". Para marcar las zonas donde hay un buscador en la página.
- role="form". Para marcar las zonas donde hay formularios.

 role="application". Se marca esta zona si hay una aplicación web interactiva, como juegos y similares que cambian las reglas del uso normal del teclado.

Es recomendable utilizar las etiquetas nativas de HTML5 con un significado semántico bien definido, como <nav>, <header>, <main>, <footer>, entre otras. Estos elementos semánticos facilitan el procesamiento y comprensión del contenido por parte de los lectores de pantalla. Sin embargo, no siempre son utilizados por desarrolladores que prefieren maquetar con elementos genéricos como <div> o tablas . En estos casos, se recomienda agregar un rol de estructura mediante el atributo role. Asimismo, es importante incluir el atributo aria-label para describir claramente el propósito del elemento, mejorando la experiencia de las tecnologías de asistencia como los lectores de pantalla. A continuación, se presenta un ejemplo de código para dar semántica al elemento:

```
<div role="navigation" aria-label="Menú principal">

        <a href="#inicio">Inicio</a>
        <a href="#nosotros">Nosotros</a>
        <a href="#servicios">Servicios</a>
        <a href="#contacto">Contacto</a>

    <//div>
```

Sección de código 5.5. Ejemplo de uso de role para dar semántica al elemento.

En este ejemplo, aunque el elemento <nav> sería más adecuado desde el punto de vista semántico para esta sección, se ha optado por utilizar un <div> junto con el atributo role="navigation" para indicar que el elemento pertenece a una región de navegación. Además, el atributo aria-label="Menú principal" proporciona una descripción accesible para los usuarios de tecnologías de asistencia, ayudándoles a comprender de manera clara el propósito del contenido.

- Atributos aria-label, aria-labelledby y aria-describedby: estos atributos nos permiten dar un nombre accesible, etiquetar o dar una descripción a los elementos HTML que lo necesiten.
 - Aria-label: sirve para dar nombre accesible a un elemento. Debe usarse sólo en ocasiones que lo requieran, por defecto el nombre de un elemento debe ser su contenido. En el siguiente ejemplo de código, el enlace utilizado para cerrar un cuadro modal se representa con una 'x', pero el atributo aria-label proporciona un nombre accesible para describir su función como 'Cerrar':

```
<a href="#" aria-label="Cerrar">x</a>
```

Sección de código 5.6. Ejemplo de atributo aria-label para un link de cierre de modal

 Aria-labelledby: como aria-label, sirve para dar un nombre a un elemento HTML, la diferencia es que con aria-label indicas directamente el texto y con aria-labelledby el id/ids del elemento/s de la página que actúan como contenido. El siguiente ejemplo de código presenta un formulario de contacto simple y accesible, acompañado de un encabezado que actúa como título:

```
<h1 id="titulo-formulario">Formulario de contacto</h1>
<form aria-labelledby="titulo-formulario">
<label for="correo">Correo:</label>
<input type="email" id="correo" name="correo">
<button type="submit">Enviar</button>
</form>
```

Sección de código 5.7. Ejemplo de uso de atributo aria-labelledby con un id de encabezado.

Este formulario está diseñado para ser accesible mediante el uso del atributo ARIA aria-labelledby, que vincula el formulario con el texto del encabezado 'Formulario de contacto' para describir su propósito.

 Aria-describedby: por otro lado, aria-describedby permite asociar una descripción larga a un elemento indicando el id del elemento que proporciona la descripción. El siguiente ejemplo de código presenta un campo de formulario acompañada de una descripción detallada:

```
<label for="nombre">Nombre:</label>
<input id="nombre" aria-describedby="nombre-ayuda"/>
<small id="nombre-ayuda">
Por favor, escribe tu nombre completo.
</small>
```

Sección de código 5.8. Ejemplo de uso del atributo aria-describedby con un id que proporciona una descripción detallada.

Este campo de entrada, mediante el uso del atributo ARIA aria-describedby, queda asociado a una descripción adicional, mejorando así la accesibilidad para tecnologías de asistencia.

Además de los atributos mencionados anteriormente, WAI-ARIA cuenta con un total de 48 atributos, que se dividen en distintas categorías según su propósito, como estados, propiedades y roles. Entre ellos destacan aria-hidden para ocultar contenido a tecnologías de asistencia, aria-expanded para elementos expandibles, y aria-live para

anunciar cambios dinámicos. Estos atributos adicionales, como aria-modal o aria-busy, complementan las capacidades semánticas de HTML, garantizando experiencias accesibles e inclusivas para todos los usuarios.

CSS3

Es el lenguaje de estilos utilizado para describir la presentación de documentos HTML o XML [49]. CSS describe cómo debe ser renderizado el elemento en la pantalla.

El **selector** apunta al elemento HTML que se desea modificar.

El **bloque de declaración** contiene una o más declaraciones separadas por punto y coma. Cada declaración incluye un nombre de propiedad CSS y un valor, separados por dos puntos. Los bloques de declaración están rodeados por llaves. La imagen 5.1 muestra un ejemplo práctico: una declaración de estilos CSS aplicada a un párrafo (), mostrando cómo se define un selector y su correspondiente bloque de declaración.



Imagen 5.1. Estructura de una declaración CSS sobre un párrafo.

A continuación, se presenta un ejemplo básico en CSS para mejorar la accesibilidad de los textos, utilizando medidas relativas, buen contraste y espaciado adecuado:

```
body {
  font-size: 100%; /* Tamaño de fuente relativo al tamaño base del navegador
*/
  line-height: 1.6; /* Espaciado entre líneas para facilitar la lectura */
  color: #000; /* Contraste adecuado para el texto */
 background-color: #fff; /* Fondo claro para mejor legibilidad */
}
p {
  font-size: 1em; /* Tamaño de fuente ajustado en relación al elemento padre
*/
  margin-bottom: 1.5em; /* Espaciado claro entre párrafos */
}
h1, h2, h3 {
 font-size: 1.5em; /* Tamaño de fuente más grande para encabezados */
 line-height: 1.2; /* Espaciado más compacto en los títulos */
  color: #333; /* Color ligeramente diferente para diferenciarlos del texto
```

```
estándar */
}
```

Sección de código 5.9. Ejemplo de estilos CSS básicos para mejorar la accesibilidad de los textos.

Bootstrap

Es un framework de CSS y JavaScript para HTML, está orientado al enfoque de desarrollo "mobile first" y al diseño responsive [50]. Proporciona hojas de estilos CSS y scripts de JavaScript compilados, así como también archivos minimizados. Además incluye archivos de fuentes Glyphicons, al igual que temas Bootstrap que pueden ser seteados en la página. A continuación, se presenta en la figura 5.3, la estructura de Bootstrap.

bootstrap/
├── css/
bootstrap.css
│
bootstrap-theme.css
bootstrap-theme.min.css
├── js/
│
│ └── bootstrap.min.js
└── fonts/
└─── glyphicons-halflings-regular.eot
<pre>glyphicons-halflings-regular.svg</pre>
<pre>glyphicons-halflings-regular.ttf</pre>
<pre>glyphicons-halflings-regular.woff</pre>
└── glyphicons-halflings-regular.woff2

Figura 5.3. Estructura de archivos y directorios de Bootstrap.

Para incluir Bootstrap en el código fuente, se utiliza la etiqueta <link> dentro de la sección <head> del documento HTML. Esto permite cargar la hoja de estilos CSS de Bootstrap, facilitando el uso de sus clases predefinidas. En el siguiente ejemplo, se muestra cómo se integra Bootstrap y se visualiza un botón primario estilizado utilizando sus clases CSS:

```
<body>
        <div class="container">
            <button type="button" class="btn btn-primary">Primary</button>
        </div>
        </body>
        </html>
```

Sección de código 5.10. Ejemplo de botón primario con Bootstrap.

React

Es un framework de código abierto del lenguaje JavaScript que sirve para renderizar interfaces de usuario (UI por sus siglas en inglés) [51]. La UI se construye a partir de pequeñas unidades como botones, texto e imágenes. React te permite desarrollar componentes reutilizables y anidables. Desde sitios web hasta aplicaciones de teléfonos, todo en la pantalla se puede descomponer en componentes. En sí React se presenta como una excelente alternativa para desarrollar todo tipo de aplicaciones web, SPA (Simple Page Application) y aplicaciones para dispositivos móviles.

Los componentes de React son funciones comunes de JavaScript que retornan código en formato JSX, el cual se asemeja al lenguaje HTML. Por ejemplo, el siguiente componente representa un botón:

```
function MyButton() {
  return (
      <button>I'm a button</button>
  );
}
```

Sección de código 5.11. Esta función retorna un componente botón llamado MyButton.

Los nombres de los componentes de React siempre deben comenzar con mayúscula, mientras las etiquetas HTML se escriben en minúscula. Esto es importante porque React distingue entre los componentes definidos por el usuario y los elementos HTML nativos del DOM. En el siguiente ejemplo de código, se utiliza JSX para retornar el componente MyButton anidado dentro de otro componente llamado MyApp:.

Sección de código 5.12. Se retorna el componente MyButton anidado dentro de otro componente llamado My App.

En el código anterior, se encuentran las palabras claves export default que permiten que el componente MyApp sea reutilizado en otras partes del proyecto.

Aunque existen diversas maneras de codificar componentes en React, la mayoría de los desarrolladores prefieren utilizar la sintaxis extendida de JSX, ya que esta se asemeja mucho al marcado HTML. Esto hace que sea más intuitiva y fácil de leer

JSX

Es una extensión de sintaxis para el lenguaje JavaScript que permite escribir marcado similar a HTML dentro de un archivo JavaScript [52]. JSX se parece muchísimo a HTML, pero es un poco más estricto y puede mostrar información dinámica. El uso de JSX no es obligatorio en React aunque la mayoría de los desarrolladores lo prefieren porque es fácil de entender, tiene todo el poder de JavaScript y es más intuitivo para crear componentes de UI.

En el siguiente ejemplo de código declaramos una variable llamada name y luego la usamos dentro de una expresión de JSX envolviéndola entre llaves.

```
const name = 'Josh Perez';
const element = <h1>Hello, {name}</h1>;
```

Sección de código 5.13. Declaración de variable.

El siguiente ejemplo de código muestra cómo insertar el resultado de llamar a la función de JavaScript formatName(user) dentro de un elemento <h1>:

```
function formatName(user) {
    if (user) {
        return `${user.firstName} ${user.lastName}`;
    }
    return 'Stranger';
}

const user = {
    firstName: 'Harper',
    lastName: 'Perez'
};

const element = (
    <h1>
        Hello, {formatName(user)}!
        </h1>
);
```

Sección de código 5.14. Inserta el nombre del usuario dentro de un elemento <h1> utilizando la función formatName(user).

Redux

Redux es una librería diseñada para gestionar el estado de las aplicaciones web de manera sencilla, consistente entre el cliente y el servidor, fácil de probar y que ofrece una excelente experiencia de desarrollo. Se basa en tres principios fundamentales:

- Única fuente de la verdad: el estado de toda la aplicación se almacena en un solo árbol dentro de un único store, es decir, el estado de la aplicación no solo se almacena en un árbol, sino en un objeto centralizado (el *store*), que es la fuente única de la verdad para toda la aplicación.
- El estado es de solo lectura: la única manera de modificar el estado es mediante la emisión de una acción, un objeto que describe qué ha sucedido.
- Los cambios se realizan con funciones puras: para especificar cómo el estado se transforma a raíz de las acciones, se utilizan reducers puros, que son funciones que determinan el nuevo estado sin modificar el estado original.

Además, Redux permite usar middlewares para manejar lógica compleja, como operaciones asíncronas, y ofrece herramientas de desarrollo para facilitar la depuración. Es ampliamente utilizado junto con React, a través de la librería react-redux, y fomenta la normalización de datos para un manejo más eficiente.

JavaScript ES6 (ECMAScript 6)

Es un lenguaje de programación que se aplica a un documento HTML para añadir características interactivas [54]. Se ejecuta del lado del cliente, es decir en el navegador web. En la actualidad el estándar ECMAScript 6 es la versión más reciente para JavaScript y agrega las siguientes características al lenguaje:

- Let y Const: la introducción de variables de ámbito de bloque.
- Funciones de flechas: sintaxis abreviada para crear funciones.
- Template literals: interpolación de cadenas y cadenas multilínea.
- Desestructuración: extracción de valores de matrices u objetos.
- **Parámetros predeterminados:** especificación de valores predeterminados para los parámetros de función.
- Operadores Rest y Spread: combinando arrays o separándolos.
- Clases: un azúcar sintáctico para la programación orientada a objetos.
- Módulos: forma estándar de organizar el código en unidades reutilizables.
- Map and Set: estructuras de datos mejoradas para almacenar colecciones.
- **Promesas:** una mejor manera de manejar el código asíncrono.

En este ejemplo de script, se utiliza el método querySelector para buscar en el documento el primer elemento <h1> y asignarlo a la constante miTitulo. Luego, se modifica el contenido de texto de ese elemento para que muestre '¡Hola mundo!'.



JQuery

Es una biblioteca multiplataforma de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol DOM, manejar eventos, desarrollar animaciones y agregar interacción con la técnica AJAX a páginas web [55].

Para incluir jQuery, es necesario agregar el enlace a la librería dentro de una etiqueta <script> en la sección <head> del documento. En el siguiente ejemplo de código, la función utiliza el evento click para ocultar el texto del selector cuando se detecta un clic sobre él.

```
<!DOCTYPE html>
<html>
 <head>
   <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.6.4/jquery.min.js">
   </script>
   <script>
      $(document).ready(function() {
        $("p").click(function() {
          $(this).hide();
        });
      });
   </script>
 </head>
  <body>
    Si haces clic aquí, el texto desaparecerá.
```

Sección de código 5.16. Ejemplo de una función JQuery.

Web Speech API

Esta API permite incorporar datos de voz en aplicaciones web. La Web Speech API tiene dos partes: SpeechSynthesis (Texto a voz) y SpeechRecognition (Reconocimiento de voz asíncrono) [56].

- Se accede al reconocimiento de voz a través de la interfaz "SpeechRecognition", que brinda la capacidad de reconocer el contexto de voz de una entrada de audio (normalmente a través del servicio de reconocimiento de voz predeterminado del dispositivo que se esté utilizando) y responder adecuadamente. Por lo general, usará el constructor de la interfaz para crear un nuevo objeto "SpeechRecognition", que tiene varios controladores de eventos disponibles para detectar cuándo se ingresa voz a través del micrófono del dispositivo. La interfaz "SpeechGrammar" representa un contenedor para un conjunto particular de gramática que su aplicación debería reconocer. La gramática se define utilizando el formato de gramática JSpeech (JSGF).
- Se accede a la síntesis de voz a través de la interfaz "SpeechSynthesis", la cual permite que los programas lean su contenido de texto (normalmente a través del sintetizador de voz predeterminado del dispositivo que se esté utilizando). Los diferentes tipos de voz están representados por objetos "SpeechSynthesisVoice" y las diferentes partes del texto que desea pasar a voz están representados por objetos "SpeechSynthesisUtterance". Estos últimos pueden pasarse al método SpeechSynthesis.speak() para convertirlo a voz.

Storage

localStorage y sessionStorage son propiedades de HTML5 que acceden al objeto Storage del navegador y tienen la función de almacenar datos de manera local, muy similar a como lo hacen las cookies [57].

Session storage

Almacena información mientras la pestaña donde se esté utilizando siga abierta. Una vez cerrada, la información se elimina.

API storage de Google

Para la persistencia de datos, no nos basamos en un base de datos relacional o no relacional, sino que, se decidió utilizar la API de almacenamiento de Google Chrome (chrome.storage) para almacenar datos de manera centralizada y garantizar su
sincronización entre las pestañas [58]. Esta API, exclusiva para extensiones de Chrome, es más eficiente y robusta que opciones como localStorage o sessionStorage. Su principal ventaja es la capacidad de sincronizar datos entre diferentes dispositivos del usuario a través de su cuenta de Google, lo que permite que la configuración de la extensión se mantenga consistente. Además, todos los componentes de la extensión pueden acceder a esta API.

Para la interfaz gráfica de la página de configuración utilizamos el framework de JavaScript, React. Adicionalmente utilizamos Bootstrap para estilizar los diferentes componentes.

La extensión consume recursos de Web Speech API, es gestionada por Redux para mantener su estado (qué comandos se están ejecutando y en qué alcance).

Para hacer el raspado de las páginas, y aplicar la mayoría de los cambios de las configuraciones utilizamos JQuery.

Con CCS3 creamos los estilos para los diferentes contrastes que después se aplican al activar la configuración.

Diseño de la herramienta

En el diseño de la interfaz de AccesaWeb, nos guiamos por principios de usabilidad como facilidad de uso, consistencia, minimizar costos de ejecución y memorización, con el fin de facilitar al usuario la comprensión y uso de las diferentes funcionalidades de la extensión. Además, es de por sí una herramienta accesible siendo coherente con las pautas establecidas por la WCAG.

Para enriquecer la funcionalidad de la herramienta, realizamos una exhaustiva recopilación de características presentes en las herramientas investigadas. A partir de este análisis, seleccionamos las funcionalidades más relevantes, priorizando aquellas que surgieron de la retroalimentación obtenida en las entrevistas con los usuarios. Este enfoque centrado en el usuario nos permitió identificar sus necesidades más urgentes y las expectativas de quienes utilizarán AccesaWeb.

La interfaz de configuración está compuesta por una sola pantalla, en la que se encuentran diversos componentes visuales para organizar la funcionalidad. A la izquierda, se ubica un menú lateral desplegable, que ocupa toda la altura de la pantalla, es desplazable y agrupa las diferentes configuraciones organizadas en módulos para un fácil acceso. En la parte superior derecha, se encuentra el encabezado con el logotipo y el título de la herramienta. El cuerpo de la página contiene un botón fijo y siempre visible para restablecer todas las configuraciones a sus valores predeterminados. A continuación, se despliegan las opciones de configuración, cada una con un título y una breve descripción que explica su funcionamiento, junto con un formulario para almacenar y aplicar los ajustes correspondientes. Uno de los factores más importantes a tener en cuenta son los colores, elegimos las combinaciones de colores blanco sobre celeste y negro sobre blanco, basándonos en el logotipo de accesibilidad universal [59] que respeta la norma 1.4.3 de la WCAG sobre el contraste mínimo. Esta norma tiene el objetivo de proporcionar suficiente contraste entre el texto y su fondo para que pueda ser leído por personas con una visión moderadamente baja, sin deficiencias de color, el tono y la saturación. Además elegimos usar algunos tonos grises de fondo para resaltar la información más relevante sobre las configuraciones.

Para la construcción del logo de AccesaWeb, nos inspiramos en el logotipo de accesibilidad web y realizamos modificaciones para adaptarlo a las iniciales de la extensión. El resultado final del diseño puede apreciarse en la imagen 5.2, que muestra el logo de AccesaWeb..



Imagen 5.2. Logotipo de accesibilidad web universal.



Logotipo de AccesaWeb.

El diseño de la herramienta pasó por varios estadios de prueba y de refinamientos a lo largo de la etapa de desarrollo como por ejemplo la decisión del menú de dos niveles que detallamos a continuación

Menú de dos niveles

Una vez que contamos con varios módulos agregados en el menú tuvimos la necesidad de implementar varios cambios, el primero de ellos fue agrupar las opciones en categorías ya que el menú se veía muy extenso y engorroso, es decir, era difícil encontrar alguna funcionalidad en particular a la hora de hacer las pruebas. Este cambio, aportó simplicidad, una mejor estructura y una visual más global de las características que brinda la herramienta.

Otra de las dificultades que surgieron haciendo pruebas de las distintas funcionalidades fue que al momento de desactivar las configuraciones resultaba muy tedioso la búsqueda de los cambios, además de que no recordábamos bien los valores por defecto. Por este motivo decidimos agregar un botón para restablecer todas las configuraciones. Este cambio implicó agregar la lógica necesaria para restablecer por defecto todas las configuraciones, en consecuencia resultó ser un agregado muy útil porque simplificó en gran medida la tarea.

Siguiendo estas ideas, realizamos un mockup de la interfaz de AccesaWeb, el cual se puede observar en la imagen 5.3.

Mi configuración		AccesaWeb
Accesibilidad	v	Restablecer
Personalizar	v	Accesibilidad
Formato	~	
Multimedia	<	Accesibilidad HTML
Comandos	v	
AccesaWeb	v	Encender
Acerca de	v	○ Apagar
		Reconocimiento de voz
		Encender
		O Apagar

Imagen 5.3. Pantalla de las opciones de la extensión.

A continuación, detallamos todas las funcionalidades de la herramienta por módulo, tal como se mostró en la imagen 5.3.

Funcionalidades

La extensión AccesaWeb consta de una sola pantalla de inicio que está conformada por un menú de configuraciones con sus respectivas secciones donde el usuario puede seleccionar las características que desee cambiar en forma centralizada.

La decisión de agrupar por módulos o secciones la configuración se debe a que queríamos un menú más intuitivo y no abrumar al usuario con tantas opciones dispersas.

Estos módulos son 5 y se describen a continuación.

Pantalla de configuraciones de la extensión AccesaWeb.

Esta pantalla incorpora en su barra superior las opciones para restablecer la configuración, como también la internalización de idiomas, lo cual queda fija para acceder en todo momento.

Mi configuración	Español Restablecer Configuración Inglés Español			
🕄 Accesibilidad 🗸 🗸	Herramienta de ayuda para las personas con discapacidad.			
Accesibilidad HTML	Accesibilidad			
Reconocimiento de voz	Accesibilidad HTML			
🖋 Personalizar 🗸 🗸 🗸	Accesibilitza HTML para avudar al lactor de pastalla			
Contraste	Activar			
Publicidades	O Desactivar			
Redes Sociales				
A Formato 🗸 🗸	Reconocimiento de voz			
Alineacion	Encender/apagar el reconocimiento de voz			
• Espaciado	C Encender			
 Espaciado entre parrafos 	Apagai			
Fuente	Personalizar			
Tamaño de fuente				
Multimedia ~	Contraste AccesaWet	, ₊		

Imagen 5.4. Página donde se puede acceder a la configuración de AccesaWeb.

Módulo "accesibilidad" de la extensión AccesaWeb.

En el módulo de accesibilidad se pueden encontrar las opciones de configuración para accesibilizar las páginas web visitadas y para activar o desactivar la navegación por comandos de voz. Este módulo de accesibilidad se puede observar en la imagen 5.5.

Accesibilidad



Imagen 5.5. Módulo de accesibilidad con sus funciones.

Módulo "personalizar" de la extensión AccesaWeb.

En el módulo personalizar se pueden encontrar las opciones de configuración para cambiar el contraste y mostrar u ocultar las publicidades y redes sociales. Este módulo personalizar se puede observar en la imagen 5.6.

Personalizar						
• Contraste						
Cambia el contraste Selecciona un contraste:						
Ninguno ~						
Publicidades						
Mostrar/ocultar publicidades Mostrar O Ocultar						
Redes Sociales						
Mostrar/ocultar redes sociales Mostrar 						

Imagen 5.6. Módulo personalizar con algunas de sus funciones.

Módulo "formato" de la extensión AccesaWeb.

En el módulo de formato se pueden encontrar las configuraciones para cambiar la alineación del texto, el espaciado entre párrafos y/o renglones, la fuente y el tamaño de la fuente de las páginas visitadas. Este módulo de formato se puede observar en la imagen 5.7.

F	Formato									
	■ Alineacion									
	Alinear el texto de las paginas visitadas									
	Ninguno ≡Izquierda ≡Centrado ≡Justificar ≡Derecha									
	TI Espaciado									
	Agregar espaciado entre renglones Selecciona un espaciado:									
	Ninguno 🗸									
	¶ Espaciado entre parrafos									
	Agregar espaciado entre parrafos									
Selecciona un espaciado:										

Imagen 5.7. Módulo formato con algunas de sus funciones.

Módulo "multimedia" de la extensión AccesaWeb.

En el módulo multimedia se puede encontrar la configuración para ocultar o mostrar las imágenes de las páginas visitadas. Este módulo Multimedia se puede observar en la imagen 5.8.

Multimedia

Imagenes

Mostrar/ocultar imagenes

Mostrar

Ocultar

Imagen 5.8. Módulo multimedia con sus funciones.

Módulo "comandos" de la extensión AccesaWeb.

En la documentación sobre los comandos de voz se puede encontrar todos los comandos disponibles con sus ejemplos y explicaciones de cómo usarlos.

Documentación sobre los comandos de voz, contiene una breve explicación de cada comando sobre cómo se usa y que hace. La siguiente imagen muestra la documentación sobre algunos de los comandos de voz.

Comandos de voz

Abrir un sitio web							
Este módulo permite acceder a un sitio web.							
Contexto Principal							
Abrir							
Comando	Descripción						
abrir	Abrir un sitio web						
Accesibilidad							
Accesibilidad Este modulo permite act	ivar la accesibilidad HTML						
Accesibilidad Este modulo permite act Contexto Principal	ivar la accesibilidad HTML						
Accesibilidad Este modulo permite act Contexto Principal Accesibilidad	ivar la accesibilidad HTML						

Imagen 5.9. Documentación sobre algunos comandos de voz de la página de configuración.

Funcionalidades recuperadas y mejoradas de tesis analizadas

De las tesinas que analizamos, seleccionamos aquellas que presentaban funcionalidades valiosas y enriquecedoras, las cuales podrían resultar muy útiles para el objetivo de AccesaWeb. Para integrarlas a nuestra herramienta, llevamos a cabo un proceso que incluyó etapas de análisis, estudio y adaptación de código, con el fin de recuperar y actualizar esos trabajos. Como primer paso, nos pusimos en contacto con sus autores para solicitar documentación e información relevante que nos ayudará en nuestros propósitos. Agradecemos la colaboración de Victoria Mussa y Javier Pérez, quienes compartieron con nosotros sus trabajos TransitaWeb y Handsfree for Web.

Para el desarrollo de AccesaWeb reutilizamos el código de la primera versión productiva de la herramienta Handsfree for Web, precisamente la versión 1.0, la cual nos

fue compartida por su autor para fines y alcance de esta tesis porque sigue la arquitectura y stack tecnológico propuesto. La herramienta nos provee de un listado de comandos de voz los cuales serán parte de AccesaWeb, además agregamos los comandos respectivos a las funcionalidades de la extensión.

También nos fue compartido el código de TransitaWeb pero por diferencias tecnológicas no se pudo reutilizar, ya que, está basado en un stack de tecnologías relacionadas con PHP, por ejemplo CakePHP, Simple DOM HTML y PhantomJS. Además es una arquitectura cliente/servidor la cual difiere con nuestra arquitectura add-on que es propia de una extensión. Nos pareció interesante algunas de sus funcionalidades por lo que decidimos agregarlas desde cero en nuestra herramienta, por ejemplo ocultar imágenes de publicidades.

Desarrollo de la herramienta

Como primera medida, incorporamos el logo, el título y los colores propios de AccesaWeb que sigue el estándar de accesibilidad universal.

Con la parte visual definida, analizamos la estructura de los componentes del proyecto y decidimos comenzar por modificar el árbol de ítems del menú, añadiendo las funcionalidades junto con su respectiva información y formularios. Entre las primeras funcionalidades implementadas se encuentran:

Ocultar imágenes

Esta funcionalidad muestra/oculta imágenes de la página web. La siguiente función (mostrada en el código 5.17) está diseñada para ocultar imágenes y elementos gráficos en una página web utilizando JQuery.

```
function hideImages() {
 $('img').hide();
  $('[style*="url("]').filter(function() {
   var style = $(this).attr('style');
    return
/\.(jpg|jpeg|png|gif|bmp|webp|tiff|svg|heic|heif|raw)\b/i.test(style);
  }).hide();
 $('svg').hide();
 $('canvas').hide();
 $('iframe').each(function() {
   var iframe = $(this);
   var iframeContent = iframe.contents();
    iframeContent.find('img').hide();
    iframeContent.find('[style*="url("]').filter(function() {
      var style = $(this).attr('style');
      return
/\.(jpg|jpeg|png|gif|bmp|webp|tiff|svg|heic|heif|raw)\b/i.test(style);
```

```
}).hide();
iframeContent.find('svg').hide();
iframeContent.find('canvas').hide();
});
}
```

Sección de código 5.17. Función hidelmages().

Se buscan todas las imágenes contenidas en elementos con las etiquetas , <svg>, <canvas>, y en aquellos elementos que tienen un atributo style cuyo valor incluye la cadena url(, con una ruta o URL que apunte a imágenes en formatos como jpg, jpeg, png, gif, bmp, webp, tiff, svg, heic, heif y raw. Además, esta búsqueda se extiende a los contenidos de todos los elementos <iframe>. Las imágenes encontradas se ocultan mediante el uso de la función hide() de JQuery.

La siguiente función realiza una búsqueda similar a la utilizada para ocultar imágenes, pero en este caso, emplea la función show() de JQuery para volver a mostrarlas. Esto incluye imágenes en elementos con etiquetas , <svg>, <canvas>, aquellos con estilos que contienen la cadena url y las imágenes dentro de elementos <iframe>, como se visualiza en el siguiente código 5.18.

```
function showImages() {
  $('img').show();
  $('[style*="url("]').filter(function() {
   var style = $(this).attr('style');
   return
/\.(jpg|jpeg|png|gif|bmp|webp|tiff|svg|heic|heif|raw)\b/i.test(style);
  }).show();
  $('svg').show();
  $('canvas').show();
  $('iframe').each(function() {
   var iframe = $(this);
   var iframeContent = iframe.contents();
   iframeContent.find('img').show();
   iframeContent.find('[style*="url("]').filter(function() {
      var style = $(this).attr('style');
      return
/\.(jpg|jpeg|png|gif|bmp|webp|tiff|svg|heic|heif|raw)\b/i.test(style);
    }).show();
   iframeContent.find('svg').show();
   iframeContent.find('canvas').show();
  });
}
```

Sección de código 5.18. Función showlmages().

Si el valor de la configuración correspondiente es false, se invoca la función hide() de JQuery para ocultar las imágenes. En caso contrario, se invoca la función show() de JQuery para mostrarlas. La siguiente función implementa esta lógica de manera clara y concisa:

```
function showHideImages(param) {
   param ? showImages() : hideImages();
}
```

Sección de código 5.19. Función showHidelmages().

Ocultar imágenes o mostrarlas puede resultar complejo cuando están dentro de iframes. Es importante tener en cuenta que los src de los iframes pueden pertenecer a un dominio externo al que está alojado el sitio web, lo que impide modificar el contenido HTML del iframe mediante JavaScript, ya que esto generaría un error por restricciones de seguridad. Solo se podría modificar el contenido de un iframe si este pertenece al mismo dominio que la página. Esta problemática podría ser objeto de análisis en trabajos futuros, ya que de su resolución depende lograr una accesibilización aún más completa de los sitios web.

Cambiar contraste

Antes de empezar con esta funcionalidad se vio necesario modificar el diseño de la interfaz de la extensión para garantizar accesibilidad. Para esto utilizamos Bootstrap por su adaptación al diseño responsivo, la gama de colores elegida fue a partir del diseño del logo de la aplicación basado en el logo oficial de accesibilidad web.

A la hora de elegir los colores para los cambios de contraste fue necesario realizar una investigación sobre los diferentes tipos de daltonismo, qué combinación de colores le son difíciles de distinguir para evitarlos y qué combinación de colores resulta ser más apta para esta problemática. En base a esto decidimos que la gama del color amarillo era el menos conflictuoso, además de que se acerca al color blanco, adecuado para otros casos.

Otra de las fuentes de información que aportaron en la decisión fue el análisis de las herramientas que brindan accesibilidad estudiadas en el capítulo 4.

A partir de toda la informacion que recopilamos decidimos que lo mejor sería agregar las combinaciones más simples y más utilizadas, como lo son, la escala de grises, la escala de grises invertida, el amarillo sobre negro y el negro sobre amarillo. Otra decisión fue dejar solo estas opciones estáticas y no dar la libertad de cambiar los colores a elección del usuario ya que podría no cumplir la pauta de la WCAG en la cual nos estamos basando. Las opciones brindadas cumplen con los criterios 1.4.3, 1.4.6 y 1.4.11 de la WCAG.

Para implementar esta funcionalidad creamos un documento de estilo contrast.css que se agrega a la página visitada de la siguiente manera:

```
const id = 'link-contrast-accesa';
let url = window.chrome.extension.getURL('contrast.css');
if (url) {
 let style = document.getElementById(id);
 if (!style) {
      style = document.createElement('link');
      style.rel = 'stylesheet';
      style.id = id;
      style.type = 'text/css';
   (document.head || document.documentElement).appendChild(style);
  }
 style.href = url;
} else {
 $('#' + id).prop('disabled', true);
  $('#' + id).remove();
}
```

Sección de código 5.20. Código para agregar la hoja de estilos de contraste en la página web.

En el documento de estilos css se definen las siguientes variables que se muestran en la sección de código 5.21, para parametrizar el uso de los distintos contrastes seleccionados posteriormente por la configuración que el usuario desee.

```
:root {
    --color: #{$color};
    --background: #{$background};
    --border: #{$border};
    --upper-color: #{$upper-color};
    --selection-bg: #{$selection};
    --selection-c: #{$selection-c};
}
```

Sección de código 5.21. Variables CSS para el contraste.

Luego, según el valor de la configuración, se asignan los colores a las variables del documento CSS previamente definidas. Estas configuraciones se realizan utilizando la estructura switch, como se muestra en el código 5.22, donde se aplica un esquema de colores específico configurando las variables --color, --background y --border con el

método setProperty. En el caso de que el valor de la configuración sea 'none', las variables se eliminan mediante el método removeProperty, restaurando el estado original..

```
switch (param) {
 case 'blackWhite':
   documentElementStyle.setProperty('--color', 'black');
   documentElementStyle.setProperty('--background', 'white');
   documentElementStyle.setProperty('--border', 'silver');
   break;
  case 'blackYellow':
   documentElementStyle.setProperty('--color', 'black');
   documentElementStyle.setProperty('--background', 'yellow');
   documentElementStyle.setProperty('--border', 'gold');
   break:
 case 'yellowBlack':
   documentElementStyle.setProperty('--color', 'yellow');
   documentElementStyle.setProperty('--background', 'black');
   documentElementStyle.setProperty('--border', '#333');
   break;
  case 'whiteBlack':
   documentElementStyle.setProperty('--color', 'white');
   documentElementStyle.setProperty('--background', 'black');
   documentElementStyle.setProperty('--border', 'grey');
   break:
 case 'none':
   documentElementStyle.removeProperty('--color');
   documentElementStyle.removeProperty('--background');
   documentElementStyle.removeProperty('--border');
   url = '';
   break;
}
```

Sección de código 5.22. Estructura switch de cambio de colores según contraste seleccionado.

Al diseñar los estilos para garantizar un contraste adecuado, seguimos la pauta 1.4.3 [AA] sobre contraste mínimo, que establece que *"los textos deben tener una relación de contraste entre el primer plano y el fondo de al menos 4.5:1"*. Basándonos en este criterio, implementamos opciones de alto contraste que superan ampliamente estos requerimientos. Por ejemplo, la combinación 'negro sobre blanco' y su inversa alcanzan un contraste de 21:1, mientras que la opción 'negro sobre amarillo' y viceversa tienen un contraste de 15.72:1, proporcionando una excelente legibilidad.

Es importante destacar que no incluimos todos los estilos desarrollados para los cambios de contraste en este documento, ya que la lista es extensa. Sin embargo, estas

configuraciones reflejan nuestro compromiso por hacer que la interfaz sea accesible y adaptativa para diversos usuarios.

Cambiar el tamaño de la fuente

Esta funcionalidad permite aumentar o disminuir el tamaño de la fuente proporcionalmente en relación con el tamaño original de la fuente. Como se ilustra en el código 5.23, la función changeFontSize() ajusta dinámicamente el tamaño de todas las fuentes de texto visibles en la página web.

```
function changeFontSize(param) {
  const rootFontSize =
parseFloat(getComputedStyle(document.documentElement).fontSize);
  // Obtener el tamaño de fuente del elemento raíz
  $('body *').filter(function() {
    return $(this).text().trim().length > 0;
  }).each(function() {
    if (!$(this).data('original-font-size')) {
      $(this).data('original-font-size', $(this).css('font-size'));
    }
    const originalFontSize =
parseFloat($(this).data('original-font-size')) / rootFontSize;
    const newFontSize = (originalFontSize * param) / 100;
    $(this).css('font-size', param === 'none' ?
$(this).data('original-font-size') : newFontSize + 'rem');
  });
}
```

Sección de código 5.23. Función changeFontSize().

El tamaño de la fuente de cada elemento se podrá disminuir hasta en un 50% respecto a su tamaño original de fuente y se podrá aumentar hasta un 200%, en cada intento se aumenta o disminuye gradualmente un 10%. A nivel de implementación se modifica el tamaño de fuente en base a medidas rem, como referencia se utiliza la medida del elemento root el cual no será modificado.

Para cambiar el tamaño de fuente tuvimos en cuenta la pauta 1.4.4 [AA] que se refiere a la "Pantalla de Texto Ajustable". Esta pauta establece que, excepto para las leyendas y las imágenes de texto, el texto debe poder agrandarse hasta en un 200% sin perder contenido o funcionalidad.

Espaciado entre renglones

La funcionalidad para ajustar el espaciado entre renglones permite aumentar o disminuir el espacio entre ellos. Esta funcionalidad es aplicable a contenedores de texto, tanto en línea como en bloque, de HTML, tales como: p, span, h1, h2, h3, h4, h5, h6, body, div, main, article, section, header, footer, aside y nav. La función changeLineSpacing del código 5.24 es la encargada de realizar este ajuste. A través de esta función, se puede establecer un valor específico para el espaciado o restaurar los valores originales según las necesidades.

```
function changeLineSpacing(param) {
  $('body,p,span,div,main,article,section,header,footer,aside,nav,h1,h2,h
  3,h4,h5,h6').filter(
    function() {
      return $(this).text().trim().length > 0;
  }).each(function() {
      const elem = $(this);
      if (!elem.data('original-line-height')) {
        elem.data('original-line-height', elem.css('line-height'));
      }
      elem.css('line-height', param === 'none' ?
elem.data('original-line-height') : param + 'em');
    });
}
```

Sección de código 5.24. Función changeLineSpacing().

La propiedad CSS line-height, utilizada para establecer la altura de las líneas, puede modificarse ajustando su valor. Para este objetivo, es posible aumentar o disminuir dicho valor empleando unidades em. Las medidas contempladas son las siguientes:

```
const options = [
    {
        label: getI18nText('none'),
        value: 'none'
    },
    {
        label: '1,5',
        value: '1.5'
    },
    {
        label: '1,75',
        value: '1.75'
},
```

```
{
    label: '2,0',
    value: '2'
}
];
```

Sección de código 5.25. Medidas em para el espaciado entre renglones.

Para el espaciamiento entre renglones, se tuvo en cuenta la pauta WCAG 1.4.12, que establece que este debe ser al menos 1.5 veces el tamaño del texto. Por lo tanto, hemos definido un valor mínimo de 1.5 em.

Espaciado entre párrafos

Esta funcionalidad permite aumentar o disminuir el espaciamiento entre párrafos. Para ello, se contempla exclusivamente el contenedor de bloque p, ya que su uso semántico en HTML está destinado a la representación de párrafos. La función changeParagraphSpacing correspondiente al código 5.26, se encarga de modificar la propiedad CSS margin-bottom de los elementos p. A través de esta función, se puede ajustar el espaciamiento de los párrafos mediante valores expresados en unidades em, o restaurar su valor original en caso de ser necesario.

```
function changeParagraphSpacing(param) {
    $('p').filter(function() {
        return $(this).text().trim().length > 0;
    }).each(function() {
        const elem = $(this);
        if (!elem.data('original-margin-bottom')) {
            elem.data('original-margin-bottom', elem.css('margin-bottom'));
        }
        elem.css('margin-bottom', param === 'none' ?
elem.data('original-margin-bottom') : param + 'em');
    });
}
```

Sección de código 5.26. Función changeParagraphSpacing().

La propiedad CSS margin-bottom, encargada de definir el margen inferior de un contenedor, puede ser ajustada. En este caso, modificamos su valor aumentando o disminuyendo unidades em específicamente para los elementos . Las medidas contempladas se visualizan en el siguiente código 5.27.

```
const options = [
  {
    label: getI18nText('none'),
    value: 'none'
  },
  {
    label: '2.0',
    value: '2'
  },
  {
    label: '2.25',
    value: '2.25'
  },
  {
    label: '2.5',
    value: '2.50'
  }
];
```

Sección de código 5.27. Medidas em para el espaciado entre párrafos.

Para el espaciamiento entre párrafos, se tuvo en cuenta la pauta WCAG 1.4.12, que establece que este debe ser, al menos, 2 veces el tamaño del texto. Por lo tanto, se definió un valor mínimo de 2 em.

Ocultar publicidades

Las publicidades suelen ser invasivas y molestas para el usuario, ya que pueden visualizarse en cualquier momento, tanto en forma de imágenes como de videos que ocupan total o parcialmente la pantalla.

Esta funcionalidad permite ocultar o mostrar las publicidades que aparecen dinámicamente en la página web, ya sea antes o después de la carga completa del documento HTML. Se contemplan publicidades generadas por medios de comunicación como El Día, El Plata, La Nación, Clarín, entre otros, así como aquellas generadas por el navegador Google Chrome.

Las funciones encargadas de mostrar y ocultar estas publicidades se puede observar en el código 5.28.

```
//mostrar publicidades
function showAds() {
  $("ins[class='adsbygoogle'],
    a[href*='ads.e-planning.net'],
    a[href*='adclick.g.doubleclick.net'],
```

```
div[id*='google_ads_iframe']").show();
}
//ocultar publicidades
function hideAds() {
  $("ins[class='adsbygoogle'],
    a[href*='ads.e-planning.net'],
    a[href*='adclick.g.doubleclick.net'],
    div[id*='google_ads_iframe']").hide();
}
```

Sección de código 5.28. Funciones para mostrar y ocultar publicidades.

Existe una dificultad en identificar elementos que son publicidad, no hay una manera unificada para identificarlas como tal, una manera de identificarlas fue mediante los valores de los atributos id, class y href, validamos que contengan la palabra ads (publicidades en inglés).

Hemos identificado un problema con algunas publicidades: ciertas imágenes, como los GIFs, no se bloquean después de que el documento HTML ha terminado de cargarse. Esto ocurre porque, desafortunadamente, no es posible modificar elementos img que pertenecen a dominios externos al servidor. En términos de seguridad del navegador y políticas de CORS (Cross-Origin Resource Sharing), no se permite la modificación de elementos que provienen de un dominio diferente al del servidor original. Esto significa que cualquier intento de manipular directamente las imágenes alojadas en dominios externos será bloqueado por el navegador para proteger contra posibles ataques y violaciones de seguridad. Esta misma problemática también ocurrirá en las funcionalidades ocultar imágenes y ocultar redes sociales.

Ocultar redes sociales

En muchos sitios se ofrece un listado de redes sociales que se visualiza como iconos o links, para reducir y facilitar el procesamiento de los lectores de pantalla se ocultan estas menciones a redes sociales. Generalmente las redes sociales que son visualizadas tienen un link a su red social asociada, en HTML5 los links se implementan mediante elementos a, la url a redireccionar se encuentra como valor del atributo href. Para nuestra búsqueda sólo contemplamos las siguientes redes sociales:

```
const socialNetworks = ['facebook', 'twitter', 'whatsapp', 'youtube',
'instagram', 'telegram', 'linkedin', 'qzone', 'pinterest', 'badoo',
'tiktok', 'github'];
```

Sección de código 5.29. Listado de redes sociales.

Para la búsqueda de todas redes sociales de la página, filtramos todos los elementos a cuyo valor de href no esté vacío y sea alguna de las redes sociales contempladas anteriormente. Por cada elemento del resultado ocultamos y liberamos su espacio en pantalla mediante las función de JQuery hide(), para que aparezcan nuevamente invocamos a la función de JQuery show().

La funcionalidad encargada de mostrar las publicidades se encuentra implementada en la función del código 5.30.

```
function showSocialNetworks() {
  $("a[href]:not([href=''])").filter(function() {
    let hrefValue = $(this).attr('href');
    hrefValue = hrefValue.toLowerCase();
    return socialNetworks.some(function(socialNetwork) {
        return hrefValue.includes(socialNetwork);
    });
  }).show();
}
```

Sección de código 5.30. Función showSocialNetworks().

Cambiar fuente

Existen muchas fuentes en la web como Arial, Courier New, Times New Roman, Calibri, Comic Sans MS, entre otros, pero solo algunas son tipografías recomendadas para sistemas accesibles. Las fuentes recomendadas para brindar una mayor accesibilidad son aquellas que no poseen serifa como por ejemplo Arial, Verdana y Helvetica. Nosotros decidimos agregar una de estas para cubrir la funcionalidad y agregar una fuente destinada para aquellas personas que tienen dislexia, como lo es OpenDyslexic [33]. En el código 5.31, se contemplan las fuentes: Arial y OpenDyslexic.

```
const options = [
{
    label: getI18nText('none'),
    value: 'none'
    },
    {
    label: getI18nText('arial'),
    value: 'arial'
    },
    {
    label: getI18nText('dyslexic'),
    value: 'opendyslexic'
    }
];
```

Sección de código 5.31. Tipos de fuentes disponibles.

La implementación de esta funcionalidad se encuentra en la función changeFont, descrita en el código que se muestra a continuación.

```
function changeFont(param) {
  // Seleccionar todos los elementos
  $('*').filter(function() {
    // Verificar si el elemento contiene texto no vacío
   return $(this).text().trim().length > 0;
  }).each(function() {
    // Almacenar el valor original de font-family si no está ya
almacenado
   if (!$(this).data('original-font-family')) {
      $(this).data('original-font-family', $(this).css('font-family'));
   }
    // Cambiar el tipo de fuente
    $(this).css('font-family', param === 'none' ?
$(this).data('original-font-family') : param);
 });
}
```

Sección de código 5.32. Función changeFont().

Para cambiar la fuente modificamos la propiedad de CSS font-family con los valores mencionados, arial y opendyslexic. Para el caso de opendyslexic no es una fuente nativa del navegador por lo que importamos archivos con extensión woff y otf que fueron descargados de la página oficial de OpenDyslexic.org.

Accesibilizar HTML

Nuestra implementación provee la funcionalidad de adaptar y accesibilizar la página web visitada tanto desde las opciones brindadas a las personas usuarias que aplican criterios de accesibilidad como también internamente a nivel de código (en lo que a nivel de automatización pueda ser posible).

Uno de los puntos considerados, en el método accesibilizar HTML fue añadir cada etiqueta de ARIA recomendadas por la WCAG. En el siguiente código (código 5.33) se muestra la incorporación de ARIA:

```
// Agregar roles ARIA a elementos específicos
addAriaAttribute('nav:not([role]), nav[role=""]', 'role',
'navigation');
addAriaAttribute('header:not([role]), header[role=""]', 'role',
'banner');
addAriaAttribute('footer:not([role]), footer[role=""]', 'role',
```

```
'contentinfo');
 addAriaAttribute('main:not([role]), main[role=""]', 'role',
'main');
 addAriaAttribute('aside:not([role]), aside[role=""]', 'role',
 'complementary');
 addAriaAttribute('section:not([role]), section[role=""]', 'role',
 'region');
 addAriaAttribute('form:not([role]), form[role=""]', 'role',
'form');
 addAriaAttribute('article:not([role]), article[role=""]', 'role',
 'article');
 addAriaAttribute('button:not([role]), button[role=""]', 'role',
 'button');
 addAriaAttribute('ul:not([role]), ul[role=""]', 'role', 'list');
 addAriaAttribute('ol:not([role]), ol[role=""]', 'role', 'list');
 addAriaAttribute('li:not([role]), li[role=""]', 'role',
'listitem');
 addAriaAttribute('table:not([role]), table[role=""]', 'role',
'table');
 addAriaAttribute('th:not([role]), th[role=""]', 'role',
'columnheader');
 addAriaAttribute('td:not([role]), td[role=""]', 'role', 'cell');
 addAriaAttribute('tr:not([role]), tr[role=""]', 'role', 'row');
 addAriaAttribute('img:not([role]), img[role=""]', 'role', 'img');
 addAriaAttribute('a:not([role]), a[role=""]', 'role', 'link');
 // Agregar atributos ARIA a elementos con identificadores
específicos
 addAriaLabelIfContains('[id!=""]', /captcha/i, 'captcha');
 // Agregar atributos ARIA a elementos con clases específicas
 addAriaLabelIfContains('[class!=""]', /captcha/i, 'captcha');
 // Agregar atributos ARIA a elementos de formulario
 $('input[id!=""], textarea[id!=""],
select[id!=""]').each(function() {
   const id = $(this).attr('id');
   const label = $(`label[for="${id}"]`);
   if (label.length > 0 && !$(this).attr('aria-labelledby')) {
      $(this).attr('aria-labelledby', id).addClass(CLASS ARIA ADDED);
    }
  });
 // Agregar atributos ARIA a botones y enlaces
 $('button,a').each(function() {
```

```
const text = $(this).text().trim();
    if (text && !$(this).attr('aria-label')) {
      $(this).attr('aria-label', text).addClass(CLASS ARIA ADDED);
    }
 });
 // Agregar atributos ARIA a imágenes
 $('img').each(function() {
    if (!$(this).attr('alt')) {
      $(this).attr('aria-hidden', 'true').addClass(CLASS ARIA ADDED);
    }
 });
 // Agregar atributos ARIA a elementos interactivos
 addAriaAttribute('[tabindex]', 'role', 'button');
 // Agregar atributos ARIA a elementos con estados
 $('[aria-expanded]').each(function() {
    const expanded = $(this).attr('aria-expanded');
    $(this).attr('aria-expanded', expanded === 'true' ? 'true' :
'false')
    .addClass(CLASS ARIA ADDED);
 });
 // Agregar aria-live a elementos dinámicos
 addAriaAttribute('[data-dynamic]', 'aria-live', 'polite');
 // Agregar aria-controls a elementos que controlan otros elementos
 $('[data-controls]').each(function() {
   const controls = $(this).attr('data-controls');
    if (controls && !$(this).attr('aria-controls')) {
      $(this).attr('aria-controls',
controls).addClass(CLASS ARIA ADDED);
    }
 });
 // Agregar aria-haspopup a elementos que abren menús o diálogos
 addAriaAttribute('[data-has-popup]', 'aria-haspopup', 'true');
 // Agregar aria-current a elementos que representan el estado
actual
 addAriaAttribute('.current', 'aria-current', 'page');
 // Agregar role="alert" a elementos que muestran mensajes
importantes
```

addAriaAttribute('.alert', 'role', 'alert');

Sección de código 5.33. Código para accesibilizar HTML mediante atributos y roles ARIA.

Es importante aclarar que uno de los puntos que se intentó desarrollar, fue agregar en forma automática textos alternativos a las imágenes. Durante nuestra investigación sobre diferentes API de reconocimiento automático de imágenes para completar los atributos alt de las imágenes, encontramos que los resultados no siempre eran lo suficientemente certeros. Por ejemplo, al intentar procesar una imagen de una pareja pintando una habitación de color celeste para su futuro bebé, la IA la interpretaba erróneamente como una pareja en un día soleado. Debido a que los resultados requerían intervención humana para alcanzar un grado de certeza adecuado, decidimos no incorporar esta funcionalidad por el momento. Además, se consideró la integración de una API de reconocimiento de imágenes basada en tecnología OCR. Entre las opciones evaluadas, destacan Google Cloud Vision y Microsoft OCR, pero debido a las limitaciones de las transacciones mensuales, decidimos no implementarlas en la versión actual. Todo lo que se agrega a la herramienta pasa por un proceso de validación para asegurar que cumple con un grado de certeza confiable.

Activar comandos de voz

Originalmente en la extensión estaba siempre activo el reconocimiento de voz y daba la posibilidad de cambiar el modo de uso, por reconocimiento continuo o reconocimiento mientras se presiona la tecla "control". Como nuestra tesis está destinada a diferentes tipos de accesibilidad decidimos agregar la posibilidad de desactivar el reconocimiento de voz y activarlo en el caso que sea necesario. Esta funcionalidad fue implementada en el código 5.34, la cual introduce un grupo de botones tipo 'radio' para alternar entre los estados de activación y desactivación. Cuando el reconocimiento está activo, se despliega el formulario de voz correspondiente (SpeechForm), y cuando está desactivado, dicho formulario desaparece, garantizando mayor flexibilidad para los usuarios.

```
<div>
  <div className='radio'>
        <label>
        <input type='radio' name={'recognitionMode'} checked=
{isOnRecognition}
        onChange={() => onChangeOnRecognition(true)}/>
        <span>{getI18nText('on')}</span>
```

```
</label>
</div>
<label>
<label>
alabel>

alabel>
alabel>

alabel>
alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

alabel>

<l
```

Sección de código 5.34. Formulario destinado a la activación y desactivación del reconocimiento de voz.

Como se mencionó anteriormente, en esta funcionalidad están las dos opciones de encender y apagar el reconocimiento de voz, además, como se puede observar al final del formulario agregamos la funcionalidad para ocultar la selección del modo de uso cuando está desactivado lo cual nos parecio mas comodo a la hora de usarlo. En la imagen 5.10, el reconocimiento de voz está desactivado, por lo que no se muestran los modos de reconocimiento de voz. Por otro lado, en la imagen 5.11, el reconocimiento de voz está activado, lo que permite visualizar las opciones de los modos de uso disponibles.



Imagen 5.10. Ejemplo de reconocimiento de voz desactivado.



Imagen 5.11. Ejemplo de reconocimiento de voz activado con las opciones de modo de uso.

Comandos de voz agregados

Para facilitar el acceso a las configuraciones mediante navegación por voz, hemos incorporado comandos específicos para cada funcionalidad de nuestra extensión, lo que optimiza y acelera el acceso.

Los comandos incorporados se agregaron en el contexto root y son los siguientes:

- Activar/desactivar accesibilidad (enable/disable accessibility)
- mostrar/ocultar imágenes (show/hide image)
- mostrar/ocultar publicidades (show/hide ads)
- mostrar/ocultar redes sociales (show/hide social networks)
- **Cambiar contraste** (change contrast): este comando permite ajustar el contraste, recibiendo como parámetro una opción específica. Para ello, se ha creado un contexto llamado "cambiar contraste" que facilita la selección de la opción mediante voz.
- **Cambiar alineación** (change align): este comando ajusta la alineación y, al igual que el anterior, cuenta con un contexto denominado "cambiar alineación" para recibir la opción seleccionada por voz.
- **Cambiar espaciado** (change spacing): este comando permite modificar el espaciado entre renglones, y su contexto correspondiente es "cambiar espaciado", que recibe la opción deseada por voz.
- **Cambiar espaciado entre párrafos** (change paragraph spacing): similar al comando anterior, este comando ajusta el espaciado entre los párrafos y se gestiona a través del contexto "cambiar espaciado entre párrafos".

- **Cambiar fuente** (change font): este comando permite seleccionar la fuente deseada, y el contexto "cambiar fuente" se utiliza para recibir la opción por voz.
- Cambiar tamaño de fuente (change font size): este comando ajusta el tamaño de la fuente y, al igual que los demás, tiene un contexto específico llamado "cambiar tamaño de fuente" para facilitar su selección por voz.

Para todos los comandos con contexto se ha descrito una ayuda que muestra las opciones disponibles, con el fin de evitar posibles equivocaciones. Si el usuario menciona una opción no válida, el sistema marcará el comando como no disponible.

Este enfoque ha requerido modularizar la lógica de las funcionalidades, garantizando que tanto el acceso a través del formulario de configuración como mediante los comandos de voz se realice de manera eficiente.

Sin embargo, nos encontramos con un inconveniente relacionado con el almacenamiento y el contexto de cada pestaña del navegador. Cada pestaña tiene su propio almacenamiento, lo que significa que los cambios realizados mediante los comandos de voz solo afectan a la pestaña activa, sin aplicarse en las demás. Inicialmente, consideramos usar el local storage nativo del navegador, pero nos topamos con esta limitación.

La primera alternativa que exploramos fue intentar sincronizar el almacenamiento entre pestañas. Sin embargo, no encontramos una API nativa del navegador que permitiera hacerlo, y la implementación de una solución personalizada resultaba en un código demasiado complejo.

Tras investigar otras opciones, decidimos utilizar la API de almacenamiento de Google Chrome, que es global a todos los contextos de las pestañas. Esto nos permite almacenar los cambios de forma centralizada y asegurarnos de que se sincronicen correctamente entre todas las pestañas, simplificando significativamente la implementación. El almacenamiento chrome.storage es una API proporcionada por Google Chrome para almacenar datos de manera persistente en las extensiones de Chrome, y se diferencia del almacenamiento tradicional de los navegadores como localStorage o sessionStorage debido a su capacidad para manejar datos de forma más eficiente y robusta en el contexto de las extensiones.

La característica más relevante de *chrome.storage* es su capacidad para almacenar datos que se pueden sincronizar entre las distintas instancias del navegador de un usuario a través de su cuenta de Google. Esto resulta especialmente útil cuando se desea que la configuración de una extensión permanezca consistente en diferentes dispositivos en los que el usuario esté autenticado con su cuenta de Google. Además, todos los componentes de la extensión pueden acceder a esta API de almacenamiento.

Una ventaja importante de chrome.storage es que sus operaciones son asincrónicas, lo que significa que no bloquean el hilo principal de ejecución. Esta

99

característica es beneficiosa, ya que permite realizar lecturas y escrituras de datos sin afectar el rendimiento de la extensión.

Una vez resuelto el problema del almacenamiento, nos enfocamos en establecer una comunicación eficiente entre los distintos contextos para aplicar los cambios de manera coherente en todas las pestañas. Dado que no existe una solución nativa para la comunicación entre dominios (pestañas), optamos por emplear el paso de mensajes entre contextos mediante la API de mensajería de Google.

Esta API permite enviar mensajes a cada contexto, como pestañas, scripts de fondo (background scripts) o scripts de contenido (content scripts), de manera que puedan ejecutar los cambios correspondientes. El proceso conocido como "message passing" o paso de mensajes es una técnica clave para permitir la comunicación asíncrona entre los diferentes componentes de la extensión, como scripts de fondo, scripts de contenido, scripts de ventanas emergentes (popup scripts) y las propias pestañas. Este enfoque permite la transmisión de información de manera asíncrona entre distintos contextos aislados dentro de la extensión.

La comunicación entre pestañas no se puede realizar directamente, pero es posible lograrlo mediante un background script. Este script actúa como un intermediario para recibir notificaciones desde un contexto (una pestaña) y reenviarlas a otras pestañas activas. Los pasos son los siguientes:

Notificar al background script desde el content script

Cuando ocurre un cambio en el contexto de una pestaña, es necesario notificar al background script para que realice las acciones necesarias. Esto se logra enviando un mensaje mediante chrome.extension.sendMessage, modularizado en la función executeBackgroundAction (mostrada en el código 5.35), que recibe un parámetro con los detalles de la acción. En el siguiente código se muestra cómo realizar una acción de modificación del DOM. En este caso, se cambia la fuente utilizando un valor que representa el tipo de fuente:

```
export function changeFont(value) {
    executeBackgroundAction({
        modifyDOM: {
            action: 'changeFont',
            param: value
        }
    });
}
export function executeBackgroundAction(action) {
    window.chrome.extension.sendMessage(action); // Enviar mensaje
```

```
al background
```

```
}
```

Sección de código 5.35. Notificación desde el content script al background script para realizar un cambio en la fuente del DOM.

Recibir y manejar mensajes de los content script en el background script
 El background script funciona como intermediario entre los content scripts y las
 pestañas activas. Utiliza un listener chrome.runtime.onMessage.addListener
 (mostrado en el código 5.36) para recibir los mensajes enviados desde los content
 scripts y reenviarlas según sea necesario. Este listener identifica el manejador para
 la acción solicitada, en este caso modificar el DOM. Para ello, se apoya en el
 objeto actionHandlers, que actúa como un mapeo de las acciones disponibles y los
 controladores correspondientes.

```
chrome.runtime.onMessage.addListener(handleMessageFromContent);
function handleMessageFromContent(actions, sender, sendResponse)
{
    _.each(actions, (actionValue, actionName) => {
      actionHandlers[actionName](sender, actionValue,
sendResponse);
    });
    return true; // Mantener el canal abierto para enviar una
respuesta asincrónica
}
const actionHandlers = {
    modifyDOM, // Incluido en el mapeo de acciones
    // Otras acciones podrían incluirse aquí
};
```

Sección de código 5.36. Listener del background script para recibir y manejar mensajes de los content script.

Reenviar mensajes a todos los content script desde el background script
 El background script actúa como intermediario para enviar mensajes a todas las
 pestañas activas (los content scripts) para que ejecuten una acción específica en el
 DOM. Antes de reenviar el mensaje, guarda en el almacenamiento (storage) los
 detalles de la acción (action) y el parámetro (param). Para reenviar el mensaje a las
 pestañas, utiliza la función chrome.tabs.sendMessage (mostrada en el código
 5.37), que permite enviar mensajes desde el background script a los content scripts
 correspondientes.

```
function modifyDOM(sender, actionValue, sendResponse) {
  // Actualizar propiedades anidadas en el almacenamiento local
  updateNestedProperty('userSettings', actionValue.action,
actionValue.param, function() {
    // Reenviar la solicitud a todas las pestañas activas
    sendDataToTabs({ modifyDOM: actionValue });
  });
}
// Función para enviar datos a todas las pestañas activas
function sendDataToTabs(data) {
  chrome.tabs.query({}, function(allTheTabs) { // Obtener todas
las pestañas
    allTheTabs.forEach(tab => {
      chrome.tabs.sendMessage(tab.id, data, response => {
        // Manejar posibles errores al enviar mensajes
        if (chrome.runtime.lastError) {
          console.warn(`Error al enviar mensaje a la pestaña
${tab.id}:`, chrome.runtime.lastError);
        }
      });
    });
  });
}
```

Sección de código 5.37. Reenviar petición modificar DOM a todos los content script .

• Recibir un mensaje y ejecutar una acción en el content script

El content script utiliza un listener chrome.extension.onMessage.addListener como se muestra en el código 5.38, para recibir los mensajes enviados desde el background script. Cuando se detecta un mensaje con la propiedad modifyDOM, se busca en el actionMap la función correspondiente a la acción solicitada. Para nuestro caso, se destaca la acción changeFont y, si se encuentra, se ejecuta utilizando el parámetro recibido, el cual indica el nuevo tipo de fuente. En el siguiente código se puede observar este comportamiento:

```
// Listener para recibir mensajes enviados desde el background
script
window.chrome.extension.onMessage.addListener(changeBackgroundHan
dler);
const changeBackgroundHandler = data => {
  // Se omiten otras lógicas (por ejemplo, manejo de módulos
importados o de reconocimiento)
  // Lógica específica para modificar el DOM: búsqueda y
ejecución de la acción solicitada
  if ( .has(data, 'modifyDOM')) {
    // Buscar en el actionMap la función asociada a la acción
(por ejemplo, "changeFont")
    const action = actionMap[data.modifyDOM.action];
    if (action) {
      // Ejecutar la función asociada, pasándole como argumento
el parámetro (por ejemplo, el tipo de fuente)
      action(data.modifyDOM.param);
    } else {
      console.warn(`No se encontró la acción:
${data.modifyDOM.action}`);
    }
  }
};
// Mapeo de acciones a funciones
export const actionMap = {
  changeFont: changeFont, // Acción específica para cambiar la
fuente
  ... // Acciones restantes
};
```

Sección de código 5.38. Recibir mensaje y modificar el DOM en el content script .

En resumen, los 4 pasos descritos establecen un flujo de comunicación robusto y modular entre los content scripts y el background script en nuestra extensión de Chrome. Primero, la solicitud se envía al background, que actualiza el almacenamiento y reenvía el mensaje a todas las pestañas mediante chrome.tabs.sendMessage. Luego, cada content script escucha la notificación y utiliza el actionMap para ejecutar la acción correspondiente (por ejemplo, changeFont), aplicando el parámetro recibido.

Cabe resaltar, aunque pueda parecer redundante, que tanto los comandos de voz como los formularios de configuración invocan esta misma lógica; es decir, tanto el disparo de una solicitud como la modificación del DOM se realizan de la misma manera en ambos casos. Esta uniformidad garantiza consistencia en el manejo y sincronización de las acciones en toda la extensión.

Guía de instalación

Para el desarrollo de nuestra página web estática, hemos utilizado tecnologías como HTML5, CSS y JavaScript, creando una estructura clara y funcional que proporciona a los usuarios toda la información relevante sobre nuestra aplicación AccesaWeb. En esta página, los visitantes encontrarán guías detalladas sobre cómo utilizar las diferentes funcionalidades de la extensión, así como instrucciones para instalar y acceder a la aplicación, con opciones de descarga tanto en formato ZIP como en archivo de instalación (.setup).

Además, hemos incluido una sección sobre nosotros, los desarrolladores, donde compartimos nuestra misión y visión detrás del proyecto. También hemos integrado un enlace directo al repositorio de GitHub, permitiendo que la comunidad acceda al código fuente, ya que la aplicación es completamente libre y gratuita, promoviendo así la colaboración y la expansión del trabajo de manera abierta y transparente. La versión de prueba se encuentra en el enlace:

https://miguel-alejandro-castillo.github.io/PageAccesaWeb/.

Alcance de la accesibilidad

La idea de la herramienta siempre estuvo orientada a que cubra las necesidades de todo tipo de discapacidades por lo que su alcance en cuanto a la accesibilidad es muy extenso, incorpora la mayoría de las normas de accesibilidad WCAG y brinda ayudas para discapacidades visuales, auditivas, motrices y cognitivas. También todas las funcionalidades pueden ser útiles para personas que no poseen discapacidad y quieran navegar cómodamente adaptando las páginas a sus necesidades.

Conclusión del capítulo

Como conclusión de este capítulo, en la etapa de desarrollo de AccesaWeb logramos crear una interfaz sencilla, intuitiva y accesible, gracias a un enfoque de diseño centrado en el usuario. Hemos desarrollado una aplicación add-on que resuelve diversos problemas de accesibilidad a través de sus múltiples funcionalidades, permitiendo a los usuarios configurar cada página según sus necesidades específicas. Este enfoque no solo mejora la experiencia de uso, sino que también facilita la personalización y optimización de la accesibilidad en cada página web.

Capítulo 6

Pruebas de usabilidad

En el capítulo anterior realizamos una introducción a la herramienta propuesta, mencionamos las decisiones de diseño, la arquitectura sobre la que está montada, tecnologías utilizadas y describimos todo el proceso de desarrollo y funcionalidades soportadas.

En este capítulo presentamos las pruebas en cuanto a lo trabajado en la etapa de desarrollo en el cual evaluamos aspectos de usabilidad. Como punto de partida planteamos objetivos de prueba, también se presenta la selección de usuarios, la recolección de resultados y el posterior análisis de los mismos.

Es importante aclarar que a su vez AccesaWeb fue utilizada en la ExpoCiencia 2022 organizada por la Fac.de Informática, la cuál fue probada por los visitantes (a razón de unas 44 personas) y salió como el trabajo del laboratorio Linti más votado por el público. Además de ser aprobada para su publicación en WICC 2024.

Objetivos de la prueba

Para nuestro propósito consideramos de gran importancia realizar pruebas de usabilidad con diferentes fines pero principalmente para tener una retroalimentación obtenida de los diferentes tipos de usuarios. Ya que la funcionalidad es muy variada planteamos los siguientes objetivos:

- Analizar la utilidad de las funcionalidades de la herramienta.
- Detectar errores o sugerencias de cambio a partir de la experiencia que tuvo el usuario.
- Validar si la herramienta puede ser una solución a las discapacidades visuales, auditivas, motrices y cognitivas.
- Conocer el perfil del usuario con datos de conocimientos previos, como el nivel de uso del navegador, uso de herramientas de asistencia y el uso de extensiones.
- Determinar el nivel de satisfacción de los usuarios en cuanto a la funcionalidad.
- Confirmar si la documentación es informativa.
- Validar la correcta elección de la plataforma.
- Evaluar el aprendizaje en cuanto a la dificultad de instalación y el uso de la extensión, si le resultó fácil las actividades, si resolvió fácilmente cada una, si tuvo alguna dificultad, cómo la resolvió, si necesito ayuda externa.
- Analizar el nivel de interés que presentan los usuarios en la herramienta.

A partir de los objetivos armamos un esquema de pruebas que inicialmente consiste en un sondeo de conocimientos previos de parte del usuario, es decir, indagar en conocimientos sobre utilización y comprensión de ciertas herramientas y plataformas como lo son las extensiones del navegador. Seguido de una serie de actividades que el

usuario podría realizar, estas actividades van llevando al usuario por cada uno de los módulos de configuración de la herramienta, como así también enseña a restablecer todas las configuraciones y cambiar el idioma. Finaliza con una encuesta de satisfacción en la cual preguntamos sobre utilidad, satisfacción, errores, sugerencias, entendimiento sobre documentación y facilidad de uso con la interfaz.

Adicionalmente a la prueba y como parte de la documentación de la herramienta, elaboramos un manual de instalación con imágenes y descripción para que el usuario inexperto aprenda a instalar una extensión desde la store de Google Chrome (anexo)

Selección de participantes

Basamos dicha selección en participantes de diferentes edades con o sin discapacidades diversas, dado que esto nos dará un diferente nivel de uso de la tecnología en cuanto a la manera de aprendizaje o entendimiento de la interfaz de la extensión.

Es importante destacar que, aunque las discapacidades de los usuarios pueden variar, durante las pruebas, cada participante interactúa con todas las funcionalidades de la herramienta, independientemente de si estas estaban específicamente orientadas a su situación particular. Esto permitió que todos tuvieran la oportunidad de explorar cada aspecto de la herramienta y ofrecer retroalimentación valiosa sobre su usabilidad y efectividad.

Nombre	Edad	Discapacidad
Norma	57	Disminución de la vista
Adriana	68	Disminución de la vista y de motricidad
Esteban	25	Disminución de la vista
Paula	29	Sin discapacidad
Natalia	34	Disminución de la vista
Viviana	55	Dislexia
Josefina	20	Sin discapacidad

Se presenta el siguiente listado de participantes:

Análisis de los resultados

Como primer punto de la prueba, realizamos una breve encuesta para recopilar los perfiles de los usuarios. A continuación, les presentamos una serie de actividades orientadas a la instalación y uso de la extensión. Dado que las funcionalidades son numerosas, decidimos que probaran una o dos de cada sección de la configuración. Como último paso, se les pidió restablecer toda la configuración y utilizar la internacionalización de idiomas. Además, si el usuario estaba interesado, podía explorar otras funcionalidades a su elección.

Presentamos el listado de las actividades llevadas a cabo durante las pruebas de usabilidad:

- 1. Instalar la extensión, si es necesario leer el manual de instalación.
- 2. Encender el reconocimiento de voz utilizando la opción modo control. Este modo permite activar el reconocimiento después de apretar la tecla "ctrl".
- Utilizar el comando "buscar" esperar 3 segundos y decir la palabra "Accesibilidad". Esto buscará la palabra accesibilidad en el buscador. Al finalizar apagar el reconocimiento de voz.
- 4. Cambiar el contraste a blanco sobre negro. Si desea puede revertirlo seleccionando la opción "sin seleccionar".
- 5. Seleccionar la opción ocultar publicidades y visitar el sitio: <u>https://www.eldia.com/</u> para comprobar su correcto funcionamiento.
- 6. Seleccionar la alineación centrada.
- 7. Selecciona el espaciado entre renglones 2.0.
- 8. Ocultar las imágenes y volver a visitar el sitio del diario el día para comprobar su correcto funcionamiento.
- 9. Restablecer las configuraciones y probar el modo en inglés.

A continuación describimos los resultados obtenidos con respecto a los objetivos de la prueba mencionados anteriormente y las conclusiones a las que llegamos luego del análisis:

Analizar la utilidad de las funcionalidades de la herramienta

Durante las pruebas se observó que las funcionalidades cumplieran su propósito, los usuarios probaron en las páginas web que usualmente visitan y comprobaron que funcionaban correctamente.

Detectar errores o sugerencias de cambio a partir de la experiencia que tuvo el usuario

El único error que detectamos en las pruebas fue que el micrófono no detectaba correctamente la voz cuando el usuario hablaba muy bajo al realizar la actividad de ejecutar comandos de voz.

Validar si la herramienta puede ser una solución a las discapacidades visuales, auditivas, motrices y cognitivas

Hemos comprobado que la herramienta satisface las discapacidades visuales, motrices y cognitivas. Aún se necesita trabajar en las discapacidades auditivas y agregar algunas funcionalidades más para las discapacidades cognitivas.

Conocer el perfil del usuario con datos de conocimientos previos, como el nivel de uso del navegador, uso de herramientas de asistencia y el uso de extensiones.

La mayoría de los usuarios ya utilizaba el navegador Google Chrome para uso cotidiano, alguno de ellos nunca había utilizado extensiones ni herramientas de asistencia por desconocimiento, ya que, el store de extensiones no es fácilmente accesible ni visible al instalar el navegador. Agregar extensiones o utilizar herramientas de asistencia es algo que implica una previa investigación por parte de los usuarios.

Para aquellos usuarios que no conocían el uso de las extensiones fue sencillo la instalación de AccesaWeb gracias a la documentación y ayuda brindada durante la prueba, además se interesaron en el uso de estas.

Determinar el nivel de satisfacción de los usuarios en cuanto a la funcionalidad

Se comprobó que la herramienta mejoró el uso de la web cuando los usuarios probaban las distintas funcionalidades, las funcionalidades que más brindaron ayuda fueron quitar las publicidades, agrandar la fuente, cambiar el tipo de fuente y la navegación por voz. El 100% de los usuarios mostraron conformidad con las funcionalidades e interés en otras que no pertenecían a la prueba. También, aportaron sus ideas para funcionalidades futuras.

Confirmar si la documentación es informativa

El 100% de los usuarios precisaron leer la documentación detallada en la herramienta y algunos leyeron la documentación adicional para instalar la extensión. El 90% indicó que la documentación es clara y sencilla, y el restante solicitó una explicación más clara en la sección de los comandos de voz.

Validar la correcta elección de la plataforma

El 100% de los usuarios utiliza el navegador Google Chrome para todas las tareas que realizan en la web, por lo cual, fue correcta la elección de la plataforma.

Evaluar el aprendizaje en cuanto a la dificultad de instalación

El 100% de los usuarios de la prueba logró instalar correctamente la extensión en su primer intento. Aquellos usuarios que previamente no tenían el conocimiento sobre instalación manifestaron que el manual provisto era simple y se entendía.

Evaluar el aprendizaje en cuanto al uso de la extensión

El 71,4% de los usuarios completó 6 de las 7 tareas en su primer intento y el porcentaje restante en su segundo intento. Esto fue debido a que surgieron problemas de conectividad y voz baja por parte del usuario en la tarea de ejecutar comandos de voz.

Analizar el nivel de interés que presentan los usuarios en la herramienta

El 100% de los usuarios utilizará la herramienta y el 42% de los usuarios se vieron interesados en las otras funcionalidades que no formaron parte de la prueba.
En resumen, la prueba de usabilidad reveló que los usuarios encontraron varias funcionalidades de la extensión intuitivas; sin embargo, también enfrentaron dificultades al acceder a ciertas configuraciones. Afortunadamente, estas dificultades se resolvieron con la ayuda de la documentación proporcionada.

Por último, queremos agradecer la participación y el tiempo de todos los participantes, sus contribuciones nos resultaron valiosas y nos ayudan a mejorar la herramienta.

Capítulo 7

En el capítulo anterior realizamos pruebas de usabilidad con distintos perfiles de usuario con el objeto de obtener un feedback sobre los objetivos previamente planteados. Al involucrar a un grupo heterogéneo de participantes, buscamos obtener una visión amplia de sus necesidades y expectativas, así como detectar cualquier dificultad que pudieran enfrentar al utilizar la herramienta.

En este capítulo se abordan las funcionalidades que no se implementaron en la extensión desarrollada y se sugieren diversas mejoras y tecnologías adicionales que podrían integrarse en futuras versiones para ampliar su alcance y eficiencia.

Conclusiones y trabajos futuros

Dentro de los objetivos planteados se logró implementar la extensión AccesaWeb como una herramienta que considera, asiste e incluye a la persona usuaria en su contexto de discapacidad, ya sea visual, motriz y/o cognitiva, aceptando sus distintas formas de interacción y navegación. Esto le permite a la persona usuaria acceder sin restricciones a la información web gracias a la aplicación de las configuraciones que el usuario requiera.

En este trabajo abordamos diferentes aspectos relacionados con la accesibilidad en la web de los cuales hemos incluido funcionalidades relevantes, las cuales surgieron de toda la investigación realizada. La componente funcional de AccesaWeb está basada en encuestas realizadas con diferentes perfiles de usuarios de la web, también de antecedentes de tesinas y de funciones de otras herramientas de accesibilidad analizadas.

AccesaWeb representa un aporte significativo al cumplimiento de la ley de accesibilidad web, garantizando los derechos de acceso a la información y al contenido de las páginas, y alineándose con las normas de la WCAG que se han logrado automatizar. La aplicación es completamente libre y gratuita, con código abierto, lo que permite a la comunidad acceder, modificar y mejorar el proyecto. Los resultados de las pruebas de usabilidad fueron muy satisfactorios, con un alto grado de aceptación por parte de los usuarios.

Quedan funcionalidades que se pueden agregar como trabajos a futuro, debido a que la tecnología avanza y las normas WCAG se van modificando, para comienzos de esta tesina, la última versión disponible era 2.1, actualmente se encuentra vigente la versión 2.2 que fue publicada el 5 de octubre de 2023.

A continuación, enumeramos los cambios que pueden extender este trabajo:

• Reconocimiento de imágenes mediante la tecnología OCR y completar el atributo alt en elementos img. Esta funcionalidad pertenece a la opción accesibilizar HTML,

para implementarlo se debería integrar una API o librería con OCR que sea gratuito para seguir garantizando el libre acceso a la extensión.

- Extender los menús: esta funcionalidad quita los menú desplegables formando un solo menú donde se pueda leer todas las opciones, ya sea para ayudar a los lectores de pantalla como para asistir a personas con dificultades cognitivas.
- Agregar opciones para la visualización del menú, arriba, abajo, derecha e izquierda.
- Agregar subtítulos a los videos. Para esta funcionalidad habría que integrar una librería o API que trabaje haciendo transcripciones de voz a texto y agregarlas en el video.
- Agregar más opciones en la internalización para que más personas puedan tener acceso a la extensión.
- Adicionar más funciones de accesibilidad HTML correspondiendo a las futuras versiones de las pautas de la WCAG.
- Resolver problemas de accesibilidad en iframes de dominios ajenos al servidor de la página visitada.
- Para la funcionalidad de ocultar imágenes, no basta con tratar únicamente los elementos o aquellos con un atributo background-image cuyo valor sea distinto de "none". También se deben revisar los objetos <canvas> y <svg> de HTML5, ya que es probable que estos muestren imágenes de forma alternativa. Más adelante, conviene analizar estos casos con mayor detalle.
- Migración del aplicativo de la herramienta a manifest V3 y subir la nueva versión a la store de Google Chrome.
- Crear versiones de AccesaWeb para otros navegadores como Mozilla y Edge.

Bibliografía

- 1. W3C, World Wide Web Consortium https://www.w3.org/Consortium/
- WCAG Working Group, W3C 2018. Web Content Accessibility Guidelines (WCAG) 2.1

https://www.w3.org/TR/2018/REC-WCAG21-20180605/

 Ley nacional n° 26.653, 2010. Ley de accesibilidad de la información en las páginas web, Argentina. <u>http://servicios.infoleg.gob.ar/infolegInternet/anexos/175000-179999/175694/norma.</u> htm

- Ley nacional n° 26.378, 2008. Convención sobre los derechos de las personas con discapacidad y su protocolo facultativo, Argentina. <u>http://servicios.infoleg.gob.ar/infolegInternet/anexos/140000-144999/141317/norma.</u> <u>htm</u>
- 5. WAI Web accessibility initiative. https://www.w3.org/WAI/
- 6. ISO, International Organization for Standardization. https://www.iso.org/home.html
- 7. ISO/IEC 40500:2012 Information technology. https://www.iso.org/standard/58625.html
- 8. Accesibilidad Web. WCAG 2.1 de forma sencilla, Olga Revilla Muñoz y Olga Carreras Montoto, 2018.
- ONTI oficina nacional de tecnologías de información. <u>https://www.argentina.gob.ar/jefatura/innovacion-publica/onti</u>
- 10. Resolución 69/2011, Argentina. http://servicios.infoleg.gob.ar/infolegInternet/anexos/180000-184999/184102/norma. html
- 11. Ley provincial n° 15115 https://normas.gba.gob.ar/ar-b/ley/2019/15115/2361
- 12.Ley Provincial D Nº4532 Adhesión de la provincia de Rio Negro a la Ley Nacional nº 26.378.

https://web.legisrn.gov.ar/digesto/normas/ver?id=2010060006

- 13. Disposición 2/2014, Argentina. Reemplazada por la disposición 6/2019. /<u>http://servicios.infoleg.gob.ar/infolegInternet/anexos/230000-234999/233667/norma</u>.<u>.htm</u>
- 14. Ley provincial n°8.654. Accesibilidad web a personas con discapacidad. <u>https://e-legis-ar.msal.gov.ar/htdocs/legisalud/migration/html/25988.html#:~:text=Me</u> <u>ndoza%2C%20sanciona%20con%20fuerza%20de%20Ley:%20Art%C3%ADculo.fa</u> <u>ciliten%20el%20acceso%20a%20sus%20contenidos%20a</u>
- 15. Disposición 6/2019, Argentina. http://servicios.infoleg.gob.ar/infolegInternet/anexos/325000-329999/329284/norma. htm
- 16. Section 508 Amendment to the Rehabilitation Act of 1973, Estados Unidos. <u>https://www.access-board.gov/law/ra.html#section-508-federal-electronic-and-information-technology</u>
- 17. Ley Equality Act 2010, Reino Unido. https://www.legislation.gov.uk/ukpga/2010/15/contents
- 18. eEurope 2002: una sociedad de la información para todos.

https://eur-lex.europa.eu/legal-content/ES/TXT/PDF/?uri=CELEX:52002DC0263

19.Ley 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico, España.

https://www.boe.es/buscar/doc.php?id=BOE-A-2002-13758

- 20. Real Decreto 1494/2007, del 12 de noviembre de 2007, España. https://www.boe.es/buscar/doc.php?id=BOE-A-2007-19968
- 21.Ley 56/2007, de 28 de diciembre, de Medidas de Impulso de la Sociedad de la Información.

https://www.boe.es/buscar/doc.php?id=BOE-A-2007-22440

- 22. UNE 139803:2012 Requisitos de accesibilidad para contenidos en la web, Europa. https://www.une.org/encuentra-tu-norma/busca-tu-norma/norma?c=N0049614
- 23. Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público, España. https://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-12699
- 24. Accessibility Insights for Web https://accessibilityinsights.io/docs/web/overview/
- 25. TAW, herramienta de análisis de accesibilidad. https://www.tawdis.net/
- 26. WAVE, conjunto de herramientas de análisis. <u>https://chrome.google.com/webstore/detail/wave-evaluation-tool/jbbplnpkjmmeebjpij</u> <u>fedlgcdilocofh?hl=es</u>
- 27. AChecker, herramienta para validar accesibilidad. https://achecks.org/
- 28. WCAG color contrast checker https://chromewebstore.google.com/detail/plnahcmalebffmaghcpcmpaciebdhgdf
- 29. EqualWeb Accessibility Checker <u>https://chrome.google.com/webstore/detail/equalweb-accessibility-ch/imemciokfejbn</u> <u>onkkinhcdfigdilcllg</u>
- 30. Siteimprove Accessibility Checker <u>https://chrome.google.com/webstore/detail/siteimprove-accessibility/djcglbmbegfleh</u> <u>mbfleechkjhmedcopn</u>
- 31. Grupo de Washington https://www.washingtongroup-disability.com/
- 32. High Contrast

https://chrome.google.com/webstore/detail/high-contrast/djcfdncoelnlbldjfhinnjlhdjlik mph?hl=es

33. Font Size Increase

https://chrome.google.com/webstore/detail/font-size-increase/ombpcpigmndepfckcif dblemkabaoihk?hl=es

34. Font Size Decrease
https://chrome.google.com/webstore/detail/font-size-decrease/mpajngnpcmjjeoflljdjp
nehcfaldcia?hl=es
35.A+ cambiar de tamaño de texto
https://chrome.google.com/webstore/detail/a%2B-fontsize-changer-lite/ckihgechpah
hpompcinglebkgcdgpkil?hl=es
36. OpenDyslexic
https://chrome.google.com/webstore/detail/opendyslexic-for-chrome/cdnapgfjopgag
<u>gbmfgbiinmmbdcglnam?hl=es</u>
37. Ocultar imágenes
https://chrome.google.com/webstore/detail/hide-imagesvideos/idmcpoiccfmffdglijpia
pgienjemppo?hl=es
38. ColorBlind - Dalton for Google Chrome
https://chrome.google.com/webstore/detail/colorblind-dalton-for-goo/afcafnelafcgjink
aeohkalmfececool?hl=es
39. Adobe Acrobat Reader
https://get.adobe.com/es/reader/
40.Zoom
https://www.zoom.com/es
41.VLC
https://www.videolan.org/vlc/index.es.html
42. InSuit
https://www.insuit.net/es/
43. User Accessibility
https://user-a.co.il/es/pagina-principal/
44. ADC, Asociación por los derechos civiles
https://adc.org.ar/
45. TransitaWeb
http://sedici.unlp.edu.ar/handle/10915/59952
46. Handsfree for web
http://sedici.unlp.edu.ar/handle/10915/63851
47. Simor
https://sedici.unlp.edu.ar/handle/10915/50072
48.HTML5
https://developer.mozilla.org/es/docs/Glossary/HTML5
49.CSS3
https://developer.mozilla.org/es/docs/Web/CSS
50. Bootstrap
https://getbootstrap.com/

51.React

https://es.react.dev/

- 52. JSX <u>https://es.legacy.reactjs.org/docs/introducing-jsx.html</u>
- 53.Redux https://redux.js.org/
- 54. JavaScript https://developer.mozilla.org/es/docs/Web/JavaScript
- 55. Jquery https://jquery.com/
- 56. Web Speech API https://developer.mozilla.org/en-US/docs/Web/API/Web_Speech_API
- 57. Storage del navegador https://developer.mozilla.org/es/docs/Mozilla/Add-ons/WebExtensions/API/storage
- 58. Storage de Google Chrome <u>https://developer.chrome.com/docs/extensions/reference/api/storage?hl=es-419</u>
- 59. Logo de accesibilidad universal ONU, 2015 https://web.archive.org/web/20160310131220/http://www.un.org/webaccessibility/lo go.shtml
- 60. Message Passing https://developer.chrome.com/docs/extensions/mv2/messaging?hl=es-419

Anexos

Manual de instalación

Instalar desde la chrome web store

1. Abra una pestaña nueva e ingrese el siguiente link <u>chrome://extensions/</u> en la barra de direcciones.



2. Otra forma de acceder es desplegando el menú del navegador, ve a la sección "Extensiones" y haga clic en la opción "Gestionar extensiones".

						-	Ø	×
						☆		:
				Nueva pestaña			Ctr	1 + T
		Ct	Nueva ventana			Ctrl + N		
			뤟	Nueva ventana de incó	gnito	Ctrl +	Mayús	+ N
			•	Persona 1	No has	iniciado se	esión.	>
			®u	Contraseñas y Autocom	npletar	ar		>
			U	Historial				>
16	5		Ŧ	Descargas			Ctr	l + J
J .		☆	Marcadores y listas				>	
			88	Grupos de pestañas				>
	Ð	Gestionar extensiones	Ð	Extensiones				>
Activa el r	r= nodo	desarrollador	Ē	Plantan data da anto	*	сы . M		c
Q Buscar extens	siones						Modo	Desarro

nsión) (Actualizar	
<i>,</i> , , , , , , , , , , , , , , , , , ,	

4. Se desplegarán varias opciones. Selecciona la opción "Cargar descomprimida"

Extensiones	Q Buscar extensiones
Cargar descomprimida Emp	aquetar extensión Actualizar
D Mis extensiones	
Accesos directos	Todas las extensiones

5. Seleccione la carpeta AccesaWeb, haga clic en abrir y listo.

📀 Selecciona el directorio de la extensión.	×
← → → ↑ 🚞 ≪ AccesaWeb-Frontend → build	✓ O Buscar en build
Organizar 👻 Nueva carpeta	
Mombre 🖈 🔨 Nombre	Fecha de modificación Tipo
AccesaWeb 🖈 🚞 AccesaWeb	27/10/2024 17:09 Carpeta de archi
images page-AccesaWe	
Este equipo ↓ Descargas	
E Desktop	
Documentos Imágenes	 hr
🕖 Música	.cr
🇊 Objetos 3D 🗸 🧹	> di
Carpeta: AccesaWeb	
	Seleccionar carpeta Cancelar
	Quitar

6. Ya puede acceder a AccesaWeb desde la sección de extensiones en la barra de navegación.

			_	٥	×
	\$	Ð	=		:
	Extensiones	×	Desarroll	ador	
	No se necesita acceso Estas extensiones no necesitan ver ni cambiar información en este sitio web.				
	🖧 Accesa Web 📮	÷			^
	Gestionar extensiones]		
cesibilizar y	Adobe Acrobat: herramientas para co 24.10.2.0 Saque el máximo partido de Google Chrome con las herramientas para PDF de Adobe Acrobat.		-		