

**PROYECTO FINAL DE INGENIERÍA**

**WCARG – LIBRERÍA DE ACCESIBILIDAD WEB  
ARGENTINA**

**Fuentes, Gonzalo Ariel – LU1048232**

**Ingeniería en Informática**

**Lema, Rodrigo Damián – LU1049834**

**Ingeniería en Informática**

Tutor:

**Mg. Pandolfo, Pablo Miguel Ángel, UADE**

**2022**



**UNIVERSIDAD ARGENTINA DE LA EMPRESA  
FACULTAD DE INGENIERÍA Y CIENCIAS EXACTAS**

## **Agradecimientos**

Queremos agradecer a todas las personas que de una forma u otra hicieron posible el desarrollo de este proyecto final de ingeniería, WCARG. Sin embargo, no queremos dejar de mencionar,

A nuestro tutor, Pablo Pandolfo por hacer posible el desarrollo de este proyecto y estar siempre involucrado.

A nuestras familias y amigos, por siempre alentarnos y acompañarnos en este trayecto académico.

A nuestro lugar de estudios, la Universidad Argentina de la Empresa y todos sus colaboradores, por darnos la posibilidad de formarnos tanto académica como profesionalmente.

A nuestros profesores de PFI, Alicia Mogliani, Mara Ares, Maximiliano Dos Santos y Pablo Inchausti. Por hacer de este año, un año de aprendizaje y estar siempre para nosotros.

Finalmente, nos agradecemos a nosotros, como compañeros, colegas y amigos en este proyecto final de ingeniería, y a todo aquel que esté leyendo este documento por permitirnos ser parte de su formación intelectual

## **Resumen**

En 2010, se sancionó la ley 26.653 de Accesibilidad de la Información, que requiere que los sitios web del sector público (entes públicos estatales y no estatales, empresas privadas concesionarios de servicios públicos y empresas prestadoras o contratistas de bienes y servicios) respeten las normas y requisitos de accesibilidad recomendados por la ONTI (Oficina Nacional de Tecnologías de la Información).

En este trabajo final se desarrolló una herramienta que permite identificar problemas de Accesibilidad Web en forma automática de acuerdo con la norma vigente en Argentina desde 2019.

## **Abstract**

In 2010, law 26653 of Accessibility of Information was passed. This law requires that Public Sector's web sites (state and non-state public entities, private companies, public services concessionaires and providing companies or contractors of goods and services) respect the rules and accessibility requirements recommended by the ONTI (National Office of Information Technology).

In this final project, a tool was developed. This tool allows to identify problems in Web Accessibility in an automatic way according to the current rules in Argentina from 2019.

## Índice

1. Introducción	8
1.1. Objetivos	8
1.1.1. Objetivo General	8
1.1.2. Objetivos Específicos	8
1.2. Alcance	8
1.3. Descripción	9
2. Antecedentes	11
2.1. Marco Teórico	11
2.1.1. Discapacidad	11
2.1.2. Deficiencias visuales	12
2.1.3. Deficiencias auditivas	13
2.1.4. Diseño Web	14
2.1.5. Accesibilidad	16
2.1.6. Accesibilidad Web	18
2.1.6.1. Barreras que impiden el acceso a la Web	19
2.1.6.2. Ventajas y dificultades de implementar Accesibilidad Web	21
2.1.6.3. Marco legislativo	23
2.1.6.4. Organismos regulatorios de la Accesibilidad Web	24
2.1.6.5. Componentes en Accesibilidad Web	25
2.1.6.6. Niveles de orientación de las WCAG 2.0	25
2.1.6.7. Pautas de Accesibilidad Web	26
2.1.6.8. Conformidad en Accesibilidad Web	28
2.1.6.9. Tecnologías donde la accesibilidad es aplicable	32
2.1.7. DevOps	38
2.2. Estado del Arte	42
2.2.1. Herramientas para validar la Accesibilidad Web	42
2.2.2. Conclusiones	48
2.3. User Research	49
2.3.1. Informe de investigación	49
2.3.2. User Persona	56
2.3.2.1. Desarrollador DevOps	56
2.3.2.2. Trabajador tecnológico que posee una discapacidad	57
2.3.2.3. Usuario de páginas web que posee una discapacidad	58
3. Descripción	59
3.1. Requerimientos Funcionales	59
3.1.2. Épicas	59
3.1.2.1. Integración continua	59

3.1.2.2. Validar criterios WCAG 2.0	59
3.1.2.3. Página accesible	59
3.1.2.4. Información de criterios	60
3.1.2.5. Warnings	60
3.1.2.6. Reporte	60
3.1.2.7. Múltiples URLs	60
3.1.2.6. Pipeline	60
3.2. Elección de librerías	60
3.2.1. Facilidad de configuración	61
3.2.2. Usabilidad	62
3.2.3. Recursos	65
3.2.4. Feedback	68
3.2.5. Costo	70
3.2.6. Limitaciones y fortalezas	70
3.2.7. Conclusiones	72
4. Metodología de Desarrollo	74
4.1. Scrum	74
4.2. Herramientas utilizadas	74
4.2.1 Visual Studio Code	74
4.2.2 Jenkins	75
4.2.3 Trello	76
4.2.4 GitHub	77
4.2.5 CANVAS	78
4.3. Frameworks, Lenguajes y Librerías	78
4.3.1 NodeJS	78
4.3.2 Pal1y	79
5. Desarrollo	80
5.1. Introducción	80
5.2. Funcionamiento	80
5.3. Diagrama de Arquitectura	83
5.4. Diagrama de Clases	84
5.5. Logo	84
6. Conclusión	85
6.1. Discusión	85
6.1. Futuras líneas de investigación	86
7. Anexos	87
7.1. Anexo A: Tabla de criterios de conformidad	87
7.1. Anexo B: Cronograma	96
9. Glosario	97

10. Bibliografía

102

## 1. Introducción

### 1.1. Objetivos

#### 1.1.1. Objetivo General

Evaluar la Accesibilidad Web en las páginas Web de la República Argentina en 2022 y desarrollar una solución tecnológica de testeo automático para determinar el grado de cumplimiento de los requisitos de accesibilidad Web dispuestos por la ONTI (Oficina Nacional de Tecnología Informática) como legislación vigente en Argentina.

#### 1.1.2 Objetivos Específicos

Implementar una librería de testeo automático para determinar si una Web es accesible o no. Esta librería se podrá integrar a un proceso de integración continua por medio de una configuración standard.

### 1.2. Alcance

El desarrollo del producto para la evaluación del cumplimiento de los requisitos de Accesibilidad Web será parte de un proceso de integración continua para el desarrollo de aplicaciones Web (por ejemplo, un pipeline en Jenkins), en donde analizará y mostrará problemas de accesibilidad Web en el caso de que la aplicación que se quiere desplegar los tenga y no supere el umbral de 30 criterios (criterios nivel A y nivel AA) de acuerdo con las Pautas de Accesibilidad para el Contenido Web (WCAG, Web Content Accessibility Guidelines) en su versión 2.0 de acuerdo con la legislación argentina vigente.

El producto informara los criterios que no fueron aceptados emitiendo un reporte de criterios a los desarrolladores de forma automática por cada criterio no aceptado.

Este entregable formara parte de un proyecto de investigación dentro del INTEC.

Características:

- Analizar los criterios de aceptación de nivel A y nivel AA (criterios alcanzados por la disposición de la ONTI en Argentina)

- Validar que el sitio Web analizado cumpla con 30 o más criterios de aceptación según la disposición de la ONTI en Argentina.
- Integrar a un pipeline o proceso de integración continua mediante una configuración standard.
- Identificar uno a uno los criterios de aceptación que no fueron aceptados en el testeado de la Web informando el número de criterio y bajo qué principio se encuentra.
- Identificar e informar el segmento de código donde la Web no es accesible.
- Emitir reportes a desarrolladores con los errores encontrados en distintos formatos.
- Permitir además de identificar errores, advertencias (warnings) y avisos (notices) sobre la página analizada.

### 1.3. Descripción

A nivel internacional el W3C (World Wide Web Consortium), en donde un grupo independiente que define los protocolos y estándares para la Web lanzó una de sus iniciativas llamada WAI (Web Accessibility Initiative), la cual estableció diferentes pautas o guías que explican cómo se tienen que crear las páginas Web para que sean accesibles. Estas guías son llamadas WCAG (Web Content Accessibility Guidelines) de las cuales existen las versiones 1.0, 2.0, y actualmente se encuentran en la versión 2.1. Estas pautas son las que generalmente suelen ser tomadas en cuenta por la mayoría de los países que buscan mejorar los niveles de accesibilidad.

Las Pautas de Accesibilidad al Contenido en la Web 1.0 (WCAG 1.0) fueron aprobadas en mayo de 1999 y es una versión estable y de referencia. Sin embargo, las WCAG 2.0 han sido desarrolladas para aplicarse a diferentes tecnologías y, a su vez, para que su utilización y comprensión sea simple, y para que su comprobación sea más precisa.

La WCAG 1.0 utiliza 15 pautas. Cada pauta tiene varios checkpoints, y cada checkpoint tiene un nivel de accesibilidad (A, AA o AAA).

La WCAG 2.0 se divide en 4 principios robustos. Cada principio tiene varias pautas, y cada pauta tiene criterios, que a su vez tienen 3 niveles de éxito. Cada nivel de éxito tiene varios puntos a cumplir.

En WCAG 2.0, el nivel 1 se llama “A”, el nivel 2 se llama “AA” y el nivel 3 se llama “AAA”.

Los cuatro principios de la WCAG 2.0 son:

- El contenido debe ser perceptible.
- Los elementos de la interfaz en el contenido deben ser manejables.
- El contenido y los controles deben ser comprensibles.
- El contenido debe ser suficientemente robusto para funcionar con las tecnologías actuales y las del futuro.

Una ventaja importante es que en las WCAG 2.0, cada criterio va acompañada de un párrafo explicando a qué tipo de usuarios ayuda y con varios ejemplos de uso.

La WCAG 2.0 se publicó el 11 de diciembre del 2008, mientras que la WCAG 2.1 se publicó el 5 de junio del 2018, incluyendo 17 criterios nuevos.

El objetivo de esta nueva versión de las WCAG (2.1) es mejorar las pautas de accesibilidad para tres grupos específicos de usuarios:

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

## 2. Antecedentes

Se lleva cabo un proceso de investigación con el objetivo de comprender la problemática elegida a fondo, caracterizar su estado actual en el territorio y conocer las soluciones que ya se encuentran disponibles en el mercado para los potenciales usuarios del proyecto.

### 2.1. Marco Teórico

Para realizar un estudio de accesibilidad, es necesario tener claro determinados conceptos. A continuación, se detallan los términos más comunes, los cuales tienen diferente significado dependiendo del contexto donde se aplique.

#### 2.1.1. Discapacidad

Dado que el eje principal de este análisis está puesto en la accesibilidad y usabilidad de sitios Web para personas con discapacidad, es necesario plantear algunos conceptos que sirvan de eje para la lectura interpretativa. Para empezar, se ha delimitado los términos Deficiencia, Discapacidad y Minusvalía del mismo modo en que es definido por la Organización Mundial de la Salud (OMS). (RODRÍGUEZ, 2011)

**Deficiencia.** Toda pérdida o anomalía de una estructura o función psicológica, fisiológica o anatómica.

**Discapacidad.** Toda restricción o ausencia, debida a una deficiencia de la capacidad para realizar una actividad en la forma o margen que se considera normal.

**Minusvalía.** Situación de desventaja social como consecuencia de una deficiencia o discapacidad, que limita o impide el desempeño de un rol que es normal en su caso, en función de edad, sexo, factores sociales y culturales.

En base a los conceptos expuestos, cabe destacar que el término correcto a tratar en esta investigación es discapacidad, pues determina una interdependencia entre la deficiencia que posee el ser humano y su dificultad para desenvolverse en el entorno, en este caso un entorno virtual que es el sitio Web.

Por una parte, está la limitación física, la observable en el cuerpo o en el andar, y por otra, los obstáculos que me imponen el entorno físico construido según convenciones sociales. Y es justamente ahí donde se hace presente la discapacidad: en aquello a lo que no se puede acceder, de lo que quedó excluido, y no por decisión propia ni por acción consciente de nadie en particular, sino seguramente por la ignorancia, el incumplimiento de leyes, o la fuerza de la costumbre, por el peso de lo habitual, de lo ideológicamente establecido y plasmado en la manera de diseñar y construir el espacio. (CORIA, 2010)

Existen distintos tipos de discapacidad: física, cognitiva, psicosocial y sensorial. Dentro las de tipo sensorial, se destacan las deficiencias visuales y auditivas.

### 2.1.2. Deficiencias visuales

Se habla de personas con deficiencia visual cuando se refiere a individuos que, con la mejor corrección médica posible, como la utilización de lentes o una cirugía ocular, podrían ver o distinguir, aunque con gran dificultad, algunos objetos a una distancia muy corta. (VILLOSLADA SÁNCHEZ, 2011)

Algunas de estas personas con deficiencia visual pueden lograr leer o visualizar una publicación cuando la tipografía es del tamaño adecuado, o sus colores no afectan a los problemas de daltonismo o acromatopsia, aunque por lo general suelen realizarlo con un mayor esfuerzo. Los tipos de deficiencia visual más comunes son:

**Baja visión:** Se entiende por baja visión, a la deficiencia en el funcionamiento visual que, aún después de tratamiento y corrección óptica, posee una agudeza visual inferior a lo establecido como normal por la Organización Mundial de la Salud. (AQUINO ZÚÑIGA, GARCÍA MARTÍNEZ y IZQUIERDO, 2012)

Las personas con baja visión poseen dificultad para realizar tareas cotidianas, como desplazarse en un área o navegar en internet, pero el remanente visual les permite aún planear o ejecutar algunas de ellas.

**Acromatopsia:** La acromatopsia, es la incapacidad para diferenciar los colores. Un sector de los individuos que padecen esta condición visualiza los colores únicamente como

blanco, negro y varios tonos de grises. Sin embargo, la mayoría de ellos no posee acromatopsia total sino parcial, también conocida como daltonismo. (SOTIL y CALVO, 2015)

**Daltonismo:** El Daltonismo o acromatopsia parcial, es una enfermedad congénita y hereditaria que se caracteriza por la incapacidad para percibir un determinado color o una gama, siendo la afección más común el no poder distinguir correctamente los colores rojo o verde. (SOTIL y CALVO, 2015)

**Ceguera:** Se habla de personas con ceguera, para referirnos a aquellas que no ven nada en absoluto o que solamente tienen una ligera percepción de luz. Con frecuencia estas personas pueden distinguir entre luz y oscuridad, pero no la forma propia de los objetos. (VILLOSLADA SÁNCHEZ, 2011)

### 2.1.3. Deficiencias auditivas

Se entiende como la carencia de la capacidad parcial o total para percibir la información que ingresa por la vía auditiva, y le impide desenvolverse en su vida diaria y desarrollarse personalmente con plena autonomía en igualdad de condiciones. (TUTUY et al., 2012)

**Hipoacusia:** Es la pérdida o disminución parcial de la capacidad auditiva. Las personas con hipoacusia pueden poseerla en grado leve, medio o severo y en la mayoría de los casos la audición puede verse mejorada mediante implementos médicos. (GARROTE ROJAS y PALOMARES RUIZ, 2014)

**Cofosis:** La cofosis o sordera, es la pérdida total de la capacidad auditiva. Las personas con cofosis deben precisar de métodos no auditivos de comunicación. (Garrote Rojas y Palomares Ruiz, 2014)

Es debido a este tipo de limitaciones, que posee una persona con discapacidad, que el correcto diseño de un sitio Web es fundamental para no condicionar su acceso a la información y favorecer su inclusión

## 2.1.4. Diseño Web

Un sitio Web es un conjunto de páginas Web que poseen relación entre sí y pertenecen a un mismo dominio Web específico. Por lo tanto, se define Diseño Web como la actividad de planificar, diseñar e implementar sitios Web.

Esta disciplina requiere un conjunto de conocimientos técnicos, conceptuales y estéticos, que el diseñador web debe tener presente, estos son:

**Interfaz gráfica:** La interfaz es la parte del sitio Web que se muestra al usuario, con la que va a interactuar. Es el entorno y presentación visual que permite la comunicación entre el internauta y la Web. (ALONSO y CAMACHO, 2010)

La interfaz gráfica está compuesta por elementos visuales que permiten al internauta relacionar esos elementos con objetos y acciones de la vida real, con el fin de realizar actividades y explorar contenidos. En base a esto, Subiela menciona estos cuatro elementos como los principales componentes visuales en la composición de una interfaz gráfica. (SUBIELA HERNANDEZ, 2010)

**Tipografía:** Es la técnica del diseño, manejo y selección de tipos, originalmente de plomo para trabajos de impresión, y actualmente trasladado también al campo digital.

En términos específicamente visuales, la tipografía es considerada un elemento tan importante como las imágenes o el color. Su importancia es tal que una mala elección en el tipo de letra puede significar dar un mensaje equivocado y personalidad diferente al objetivo a comunicar. (CADENAS, 2010)

**Color:** Es la percepción creada por el sistema visual, en respuesta al estímulo de la luz en la región visible del espectro electromagnético. (STOCKMAN y BRAINARD, 2010)

Cada color posee características propias, dependiendo del color utilizado se afecta la percepción e incluso el comportamiento del usuario. Antes de emprender la lectura de un texto o analizar una imagen, el color ya ha transmitido un mensaje. (Velasco Santos et al., 2011)

**Formas:** Es el conjunto de líneas y superficies que determinan el contorno o límite de un producto visual. Subiela, sobre la importancia de las formas menciona: “formará parte intrínseca de dicha composición, ya que textos, imágenes e incluso colores deberán ubicarse dentro de cajas, cajas que presentarán una forma determinada. Y aún más, dentro de cada caja, se presentarán a su vez distintas imágenes, configuradas por distintas formas”. (SUBIELA HERNANDEZ, 2010)

**Imagen:** Es la representación figurativa de una persona, cosa o situación real o abstracta. Los gráficos son de alto impacto visual, inciden rápidamente en el sentido de la vista. Las imágenes explican, llenan espacios, complementan y agilizan la información. (VELASCO SANTOS et al., 2011)

Los aspectos para considerar en el diseño de la interfaz gráfica son los siguientes:

**Consistencia:** Un sitio Web usualmente está formado por varias páginas y secciones. Se debe lograr un diseño consistente entre las distintas partes del sitio. Si el sitio Web resulta familiar al usuario, lo persuade a navegar más tiempo en él. (TUCH et al., 2012)

Las formas, colores, tipografías e imágenes drásticamente diferentes entre sí pueden crear un efecto poco agradable para el internauta. Si el mismo elemento de un sitio Web es usado de manera distinta en otra sección del sitio, es una nueva función que el usuario debe aprender.

La consistencia visual, por su parte, impide que el internauta tenga que aprender nuevas funciones y crea una sensación de comodidad y estabilidad que da confianza al usuario.

**Balance:** Es la armonía que debe existir entre los diferentes elementos visuales que componen el sitio Web. Imágenes, tipografía, elementos de gran tamaño o de tamaños reducidos, el balance compositivo hace referencia al concepto de equilibrio visual, incluye la organización de los elementos de tal forma que ningún área del diseño domine a otra. Todo funciona y encaja como un todo, los elementos individuales contribuyen al todo, pero no son el todo. (VEGA PÉREZ, 2016)

Una composición sin balance puede crear tensión. El sitio Web debe estar diagramado de tal manera que no solo resulte estética, sino que además logre comunicar información sin entorpecer al usuario.

**Contraste:** El contraste es una forma de distinguir unos elementos de otros para mejorar la comunicación. Se logra mediante diferentes tipografías, diferencia en las formas de los elementos, botones o menús, diferencias de tamaño, texturas y fondos. Los contrastes pueden emplearse para enfocar la atención del usuario y facilitar la visualización del contenido. (VELÁSQUEZ HENAO, 2012)

**Énfasis:** Énfasis o acentuación gráfica es una forma de distinguir o destacar un elemento. Un ejemplo de esto es un fragmento a color en una imagen en blanco y negro, para de esta manera facilitar la comprensión de lo que se quiere comunicar, destacando un elemento en particular. (SUBIELA HERNANDEZ 2010)

**Ritmo:** El ritmo o repetición es la manera de establecer patrones con los que se ordenan los elementos. Si los elementos del sitio Web siguen un patrón determinado, resulta más sencillo para el usuario comprender el contenido y acceder a la mayor parte del mismo en poco tiempo. (GONZÁLEZ ORELLANA, 2012)

**Proximidad:** Este principio hace referencia a colocar juntos o próximos los elementos del sitio Web que estén relacionados, para así conformar unidades visuales que ofrezcan coherencia al contenido de la página, permitiendo un mejor acceso a los usuarios. (MARCOS ALBA y OLIVAREZ RUIZ, 2014)

### 2.1.5. Accesibilidad

La accesibilidad es la cualidad de fácil acceso para que cualquier persona, incluso aquellas que tengan limitaciones en la movilidad, en la comunicación o el entendimiento, pueda llegar a un lugar, objeto o servicio.

Se la puede relacionar con los siguientes términos:

- **Accesibilidad Arquitectónica:** Referida a edificios públicos y privados.
- **Accesibilidad Urbanística:** Referida al medio urbano o físico.
- **Accesibilidad en el Transporte:** Referida a los medios de transporte públicos.
- **Accesibilidad en la Comunicación:** Referida a la información individual y colectiva.

• **Accesibilidad Digital:** Referida a la facilidad de acceso a las TIC y a contenidos en internet, para cualquier persona sea con discapacidad, de edad avanzada o por privación circunstancial.

La accesibilidad universal es aquella condición que deben cumplir los entornos, procesos, bienes, productos y servicios, así como los objetos o instrumentos, herramientas y dispositivos, para ser comprensibles, utilizables y practicables por todas las personas en condiciones de seguridad, comodidad y de la forma más autónoma y natural posible. (OBSERVATORIO DE LA ACCESIBILIDAD, 2022)

Actualmente, la accesibilidad ha dejado de ser sinónimo de supresión de barreras físicas para adoptar una dimensión preventiva y amplia, generalizable a todo tipo de espacios, productos y servicios. Por otra parte, se trata de una variable fundamental para garantizar el cumplimiento del principio de igualdad de oportunidades, convirtiéndose paulatinamente en un reconocimiento general, como mejora de la calidad de vida de todos los ciudadanos. (OBSERVATORIO DE LA ACCESIBILIDAD, 2022)

Desde el punto de vista poblacional, según “Design for All Foundation” se indica que la accesibilidad es fundamental para un 10% de la población, para un 40% es necesario y para el 100% es comfortable.

La accesibilidad es un derecho de la persona y, como tal, debe ser tratado para conseguir la equiparación de oportunidades. Asimismo, la persona tiene derecho a la autonomía y la movilidad personal, como correspondencia al logro de una vida plenamente independiente. Por último, un aspecto fundamental es la normativa, que desde hace unos años obliga a que todo bien, producto o entorno sea accesible para todas las personas.

Como se puede ver en la tabla Nro.1.1, muchas de las ayudas diseñadas para usuarios con necesidades especiales pueden ser útiles para personas sin necesidades especiales que se encuentran en situaciones especiales.

Tabla 1.1. La accesibilidad, una necesidad general (tomada de "La interacción persona-ordenador". Abascal, 2001)

<b>Problemática</b>	<b>Necesidades especiales</b>	<b>Ejemplos de situaciones especiales</b>
<b>Sin visión</b>	Ciegos	Personas <ul style="list-style-type: none"> <li>• con ojos ocupados (por ejemplo, conduciendo o en navegación telefónica)</li> <li>• en la oscuridad</li> </ul>
<b>Poca visión</b>	Personas con limitaciones visuales	Personas con un visualizador pequeño
<b>Operable sin poder oír</b>	Personas sordas	Entornos ruidosos <ul style="list-style-type: none"> <li>• oídos ocupados</li> <li>• silencio forzado (bibliotecas, etc.)</li> </ul>
<b>Oído limitado</b>	Personas duras de oído (personas cuya pérdida de audición es entre leve y grave)	Personas en entornos ruidoso
<b>Operable con manualidad limitada</b>	Personas con limitaciones motrices	Personas <ul style="list-style-type: none"> <li>• con vestidos especiales</li> <li>• o que van en un vehículo que se balancea</li> </ul>
<b>Operable con cognición limitada</b>	Personas con cognición limitada	Personas distraídas <ul style="list-style-type: none"> <li>• con pánico</li> <li>• o bajo la influencia del alcohol</li> </ul>
<b>Operable sin lectura</b>	Personas con problemas cognitivos	Personas que <ul style="list-style-type: none"> <li>• no conocen ese lenguaje</li> <li>• visitantes,</li> <li>• no tienen sus anteojos</li> </ul>

### 2.1.6. Accesibilidad Web

La palabra acceso se puede definir como entrada o paso, o la acción de llegar o acercarse. Accesibilidad significa proporcionar flexibilidad para acomodarse a las necesidades de los usuarios y sus limitaciones o preferencias. (DICCIONARIO DE LA LENGUA ESPAÑOLA, 2022)

En el contexto de Internet, la accesibilidad Web es la facilidad para que cualquier persona pueda acceder al contenido de un sitio en diferentes condiciones. (HONORABLE CONGRESO DE LA NACIÓN ARGENTINA, 2010)

La accesibilidad Web debe verse como el diseño universal las personas, sin importar si tienen alguna discapacidad física y sin importar sus condiciones ambientales.

La Accesibilidad Web permite tratar y entender a las personas en su diversidad y no en su minusvalía. (FERNANDEZ, 2015)

La accesibilidad alcanza (WCAG 2.0, 2008):

- Personas con discapacidad.
- Personas mayores y niños.
- Personas con equipos anticuados.
- Personas con bajos recursos económicos.
- Conexiones lentas.

Desarrollar una página Web accesible no supone tener que dejar de usar imágenes y colores. Algunos sitios Web desarrollan una versión normal de la página y otra versión accesible para personas con algún tipo de discapacidad. No se debe ver la accesibilidad como algo anormal que discrimine a las personas que no puedan acceder a la versión normal. (REVILLA MUÑOZ, 2013)

En ocasiones, la versión accesible es una versión reducida en la que no está disponible todo el contenido de la Web original. No tiene sentido crear una versión poco accesible y otra accesible. Si se va a realizar un sitio Web, lo mejor es realizar un único sitio.

### **2.1.6.1. Barreras que impiden el acceso a la Web**

Se pueden encontrar barreras para acceder a la Web, pero algunas están más relacionadas con unas discapacidades que con otras (LUJAN MORA, 2006):

Las barreras para las personas invidentes son:

- Imágenes sin texto alternativo para describir su contenido. Las personas ciegas usan lectores de pantalla que les van leyendo el contenido de la Web. Si no se incluye un texto alternativo, los lectores de pantalla no pueden acceder a esa información.
- Imágenes complejas, como gráficos que representan datos sin una descripción en texto de lo que se muestra visualmente.

- Elementos multimedia, como pueden ser vídeos, sonidos o animaciones, sin descripción textual o sonora.
- Tablas en las que el contenido es incomprensible cuando se lee de forma secuencial.
- Formatos no accesibles de documentos que pueden dar problemas a los lectores de pantalla, por ejemplo, un documento PDF que no cumpla con las normas de accesibilidad.

Las barreras para las personas con problemas de visión son:

- Tamaño de letras definido a través de las hojas de estilo CSS (font-size) con medidas absolutas que no permiten redefinirlo.
- Diseño de página que, al modificar el tamaño de fuente, dañan la maquetación y hacen difícil la navegación.
- Poco contraste entre el fondo, las imágenes y el texto.
- Texto agregado mediante imágenes, que dificulta su lectura al aumentar el tamaño.
- Uso de color para resaltar texto sin usar otro elemento de formato adicional (como cursiva, negrita o subrayado).
- Poco contraste en las imágenes con texto o entre el texto y el color de fondo de la página.
- Navegadores que no soportan el uso de hojas de estilo definidas por el usuario.

Las barreras para las personas con sordera o hipoacusia son:

- La falta de subtítulos o de transcripciones de los contenidos.
- La falta de imágenes que ayuden a la comprensión del contenido de las páginas. Las imágenes favorecen la comprensión a las personas cuyo idioma principal es el lenguaje de señas.
- Necesidad de entrada de voz en algunos sitios Web. Por ejemplo, cuando se pide al usuario que hable a través del micrófono.

Las barreras para las personas con alguna discapacidad motriz son:

- Iconos, botones, enlaces y otros elementos de interacción demasiado pequeños, que dificultan su uso a personas con poca destreza en sus movimientos.
- Falta de independencia de dispositivos, que impide usar correctamente la Web con el teclado en vez de con el mouse.
- Tiempos de respuesta limitados para interactuar con la página. Por ejemplo, sitios donde hay que realizar un test con un tiempo máximo de finalización.
- Elementos de interacción cuya navegación se realiza en orden no secuencial.

Las barreras para las personas con discapacidad cognitiva y neurológica son:

- Los elementos visuales o sonoros que no se puedan desactivar cuando se desee y que puedan distraer a las personas con déficit de atención.
- La falta de una organización clara y coherente de la información que ayude a las personas con problemas de memoria o con poca capacidad cognitiva.
- El uso de un lenguaje complejo.
- La ausencia de gráficos en los sitios Web que complementen la información textual.
- Tamaño de letra fijo que no se puede aumentar.
- Destellos o parpadeos con altas frecuencias que pueden provocar incluso ataques de epilepsia.

### **2.1.6.2. Ventajas y dificultades de implementar Accesibilidad Web**

El hecho de mejorar el posicionamiento de una página Web en los buscadores es una razón para mejorar la accesibilidad de las páginas Web. Pero existen otras razones que también son importantes. (LUJAN MORA, 2022)

#### **Incrementa la cuota de mercado y audiencia de la Web**

- Mejora la usabilidad de la Web para todo tipo de usuarios.
- Permite mejorar el acceso a los contenidos Web a las personas mayores.
- Permite la reutilización de contenidos por múltiples formatos o dispositivos.

- Incrementa el soporte para el mercado internacional (subtítulos, idiomas alternativos, contenidos universales...).
- Ayuda a reducir la llamada brecha digital.

### **Mejora la eficiencia y el tiempo de respuesta**

- Reduce los costes de desarrollo y mantenimiento del sitio.
- Mejora del motor de búsqueda del sitio.
- Reutilización de contenidos.
- Gestiona la carga del servidor (hojas de estilo, vínculos mejor definidos, alternativas textuales...).
- Gestiona el ancho de banda del servidor.

### **Demuestra responsabilidad social**

- Refuerza positivamente la imagen de las entidades enunciadas en el art 1 de la ley 26.653.
- Permite diferenciarse de la competencia.
- Posibilita llegar a una proporción influyente de la población.

Algunas dificultades que se encuentran en la implementación de la accesibilidad Web son las siguientes (Carreras Montoto, 2006):

- Desconocer la composición del público de un sitio.
- Desconocer cómo ofrecer la información de forma accesible.
- Considerar que la accesibilidad Web implica más trabajo o mayores costos.
- Considerar incompatible un buen diseño y un sitio accesible.

### 2.1.6.3. Marco legislativo

En los orígenes de Internet no existía ningún tipo de regulación, ni legal ni técnica, sobre accesibilidad, pero en los últimos años han surgido multitud de iniciativas con la intención de normalizar la accesibilidad en Internet. En Argentina, la legislación vigente comprende:

- Decreto 1172/2003 de Acceso a la Información Pública que establece que toda persona tiene derecho a requerir, consultar y recibir información de cualquier organismo público. El decreto señala como información: documentos escritos, fotográficos, grabaciones, soporte magnético, digital y que haya sido creada u obtenida por organismos públicos o que obre en su poder o bajo su control. En ese sentido se prevé que la información debe estar adecuadamente organizada, sistematizada y disponible para garantizar un amplio y fácil acceso. (PODER EJECUTIVO NACIONAL, 2003)
- Decreto 378/2005 del Plan Nacional de Gobierno Electrónico donde se impulsa el uso intensivo de las Tecnologías de la Información y las Comunicaciones (TICs) para mejorar la relación del gobierno con los ciudadanos, e incrementar eficiencia de la gestión y la transparencia y la participación. (PODER EJECUTIVO NACIONAL, 2005)
- Ley 26.653 que garantiza y amplía el acceso a la información pública a las personas con discapacidad. Mediante esta ley, el Estado Nacional debe respetar en los diseños de sus páginas Web las normas y requisitos sobre accesibilidad de la información que faciliten el acceso a sus contenidos para todas las personas con discapacidad, con el objeto de garantizarles la igualdad real de oportunidades y trato, evitando así todo tipo de discriminación. (HONORABLE CONGRESO DE LA NACIÓN ARGENTINA, 2010)

La ley comprende a:

- El Estado nacional, y sus organismos descentralizados o autárquicos.
- Entes públicos no estatales.
- Empresas del Estado.
- Empresas privadas concesionarias de servicios públicos.

- Empresas prestadoras o contratistas de bienes y servicios.
- Instituciones u organizaciones de la sociedad civil que sean beneficiarias o reciban subsidios, donaciones o condonaciones, por parte del Estado o celebren con el mismo contrataciones de servicios.
- Decreto 656/2019 que designa a la ONTI, en su rol de Autoridad de Aplicación de la Ley, monitorea y evalúa la accesibilidad de sitios Web. (PODER EJECUTIVO NACIONAL, 2019)
- Decreto 6/2019, en donde en los anexos complementarios a se incorporan las Pautas de Accesibilidad de Contenido Web y los Criterios de Conformidad aplicables de acuerdo con dichas pautas (38 criterios, 25 tipo A o “Must”, y 13 Tipo AA o “Should”), que consisten básicamente en requisitos técnicos que deben cumplimentarse a la hora de desarrollar un sitio Web para que este resulte accesible. (PODER EJECUTIVO NACIONAL, 2019)

La disposición recomienda al cumplimiento de un nivel mínimo de 30 criterios.

#### **2.1.6.4. Organismos regulatorios de la Accesibilidad Web**

Existen diversos organismos que regulan la accesibilidad Web. Algunos son tenidos en cuenta a nivel mundial y otros son propios de los países. (REVILLA MUÑOZ, 2013)

A nivel internacional, el W3C es una organización fundada en el año 1994 y formada por empresas y organismos de distintos países y sectores que trabajan para desarrollar estándares Web. La misión del W3C es la de guiar la Web hacia su máximo potencial a través del desarrollo de protocolos de uso común y pautas que propicien el crecimiento y evolución, asegurando la interoperabilidad de la Web para fomentar su universalidad. (W3C, 1994)

En el año 1998, la W3C presentó la Iniciativa de Accesibilidad Web (WAI, Web Accessibility Initiative) que es un grupo interno de trabajo permanente de la W3C, y que se enfoca en hacer la Web accesible. (WAI, 1998)

La WAI colabora con organizaciones de todo el mundo, mejorando la accesibilidad en la Web a través de cinco áreas de trabajo (WAI, 1998):

1. Tecnología.
2. Pautas.
3. Herramientas de validación y reparación.
4. Educación y difusión.
5. Investigación y desarrollo.

### **2.1.6.5. Componentes en Accesibilidad Web**

Los componentes esenciales de Accesibilidad Web son los siguientes (WAI, 1998):

- Contenido, es la información presente en una página Web.
- Navegadores Web, reproductores multimedia y otros agentes de usuario.
- Tecnología de ayuda (AT), por ejemplo: lectores de pantalla, teclados alternativos, software de escaneo, etc.
- Herramientas de autor, se utilizan para crear sitios Web.
- Herramientas de evaluación, se utilizan para evaluar la Accesibilidad Web, por ejemplo: validadores HTML, validadores CSS, etc.

### **2.1.6.6. Niveles de orientación de las WCAG 2.0**

Las WCAG 2.0 se han desarrollado en cooperación con individuos y organizaciones de todo el mundo, con el fin de proporcionar un estándar compartido para la accesibilidad del contenido Web que satisfaga las necesidades de personas, organizaciones y gobiernos a nivel internacional. Las WCAG 2.0 se han diseñado para ser aplicadas a una amplia gama de tecnologías Web ahora y en el futuro, y para ser verificables con una combinación de pruebas automatizadas y evaluación humana. (REVILLA MUÑOZ, 2013)

Los individuos y organizaciones que emplean las WCAG son un grupo amplio y variado que incluye diseñadores y desarrolladores Web, profesores y estudiantes. Para poder satisfacer las necesidades tan variadas de esta audiencia, se proporcionan varios niveles de orientación: principios, pautas, criterios de conformidad verificables y una amplia colección de técnicas suficientes, técnicas recomendables y fallos comunes documentados con ejemplos, enlaces a

recursos adicionales y código. (WCAG 2.0, 2008)

- **Principios:** en el nivel más alto se sitúan cuatro principios que proporcionan los fundamentos de la accesibilidad Web: perceptible, operable, comprensible y robusto.
- **Pautas:** por debajo de los principios están las pautas que proporcionan los objetivos básicos que los autores deben lograr. Las pautas no son verificables, pero proporcionan el marco y los objetivos generales que ayudan a los autores a comprender los criterios de conformidad y a implementar mejor las técnicas.
- **Criterios de Conformidad:** para cada pauta se proporcionan los criterios de conformidad verificables que permiten emplear las WCAG 2.0 en aquellas situaciones en las que existan requisitos y necesidad de evaluación de conformidad como: especificaciones de diseño, regulación o acuerdos contractuales. Con el fin de cumplir con las necesidades de los diferentes grupos y situaciones, se definen tres niveles de conformidad: A (el más bajo), AA y AAA (el más alto).
- **Técnicas suficientes y recomendables:** las técnicas son informativas y se agrupan en dos categorías: aquellas que son suficientes para satisfacer los criterios de conformidad, y aquellas que son recomendables. Las técnicas recomendables van más allá de los requisitos de cada criterio de conformidad individual y permiten a los autores afrontar mejor las pautas. Algunas de las técnicas recomendables tratan sobre barreras de accesibilidad que no han sido cubiertas por los criterios de conformidad verificables.

Todos estos niveles de orientación (principios, pautas, criterios de conformidad y técnicas suficientes y recomendables) actúan en conjunto para proporcionar una orientación sobre cómo crear un contenido más accesible. (REVILLA MUÑOZ, 2013)

### 2.1.6.7. Pautas de Accesibilidad Web

Las Pautas de Accesibilidad de Contenido Web 2.0 están organizadas alrededor de cuatro principios de diseño accesible que son perceptibilidad, operatividad, comprensibilidad y robustez.

## Perceptibilidad

Los sitios Web deben ser fáciles de percibir. Se debe presentar la información y la interfaz de usuario de forma que todos los usuarios puedan percibirlas. Las pautas asociadas con este principio son (WCAG 2.0, 2008):

- **Pauta 1.1. Alternativas textuales.** Se deben proporcionar alternativas textuales para cualquier contenido no textual, para que el contenido pueda ser transformado a otro formato en función de las necesidades de las personas, como, por ejemplo, aumentar el tamaño de letra, pasarlo a formato braille, voz, símbolos o un lenguaje más simple.
- **Pauta 1.2. Contenido multimedia dependiente del tiempo.** Se deben proporcionar alternativas para los contenidos multimedia que dependen del tiempo.
- **Pauta 1.3. Adaptabilidad.** Se debe crear contenido que pueda ser presentado de maneras diferentes (por ejemplo, un diseño más simple) sin perder información ni estructura.
- **Pauta 1.4. Distinguible.** Se debe facilitar a los usuarios ver y oír el contenido, incluyendo la separación entre el primer plano y el fondo en la que se proporcionan algunos consejos sobre el uso del color, del contraste y del tamaño del texto.

## Operabilidad

Los sitios Web deben ser operativos. Los componentes de la interfaz usuario y la navegación deben ser fáciles de operar. Las pautas asociadas con este principio son (WCAG 2.0, 2008):

- **Pauta 2.1. Accesibilidad a través del teclado.** Se debe conseguir que toda la funcionalidad esté disponible a través del teclado.
- **Pauta 2.2. Tiempo suficiente.** Se debe proporcionar al usuario tiempo suficiente para leer y usar el contenido.
- **Pauta 2.3. Ataques.** No se deben diseñar contenidos que se sepa que puedan provocar ataques con convulsiones.
- **Pauta 2.4. Navegable.** Se deben proporcionar formas de ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran en todo momento.

## Comprensibilidad

Un sitio Web debe ser fácil de entender. La información y el manejo de la interfaz de usuario deben ser comprensibles. Las pautas asociadas con este principio son (WCAG 2.0, 2008):

- **Pauta 3.1. Legible.** Se debe hacer que el contenido textual sea legible y comprensible.
- **Pauta 3.2. Predecible.** Hacer páginas Web predecibles en su apariencia y su funcionamiento.
- **Pauta 3.3. Ayuda a la entrada de datos.** Ayudar a los usuarios a evitar y corregir errores.

## Robustez

Un sitio Web debe ser robusto. El contenido debe ser lo suficientemente robusto para que pueda ser correctamente interpretado por una gran variedad de agentes de usuario, incluidas las tecnologías de apoyo. La pauta asociada con este principio es (WCAG 2.0, 2008):

- **Pauta 4.1. Compatible.** Se debe maximizar la compatibilidad con agentes de usuario actuales y futuros, incluyendo tecnologías de apoyo.

### 2.1.6.8. Conformidad en Accesibilidad Web

El documento de las WCAG 2.0 ofrece un estándar técnico sobre accesibilidad para los contenidos Web. Está acompañado por otros documentos de apoyo, que lo complementan y facilitan la evaluación e implementación de la accesibilidad (WCAG 2.0, 2008):

- **Cómo cumplir con las WCAG 2.0.** Es un documento de referencia rápida que se puede personalizar. Incluye una lista con todas las pautas, criterios de conformidad y técnicas que se pueden emplear para desarrollar y evaluar contenido Web.
- **Comprender las WCAG 2.0.** Es una guía para comprender e implementar cada pauta y criterio de conformidad.
- **Técnicas para las WCAG 2.0.** Es una colección de técnicas y errores que se cometen frecuentemente. Existen técnicas suficientes y recomendadas y son generales o específicas para alguna tecnología. También se incluye la descripción de los errores, ejemplos, código y pruebas.

- **Los documentos de las WCAG 2.0.** Es un diagrama y una descripción de cómo se relacionan entre sí estos documentos técnicos.

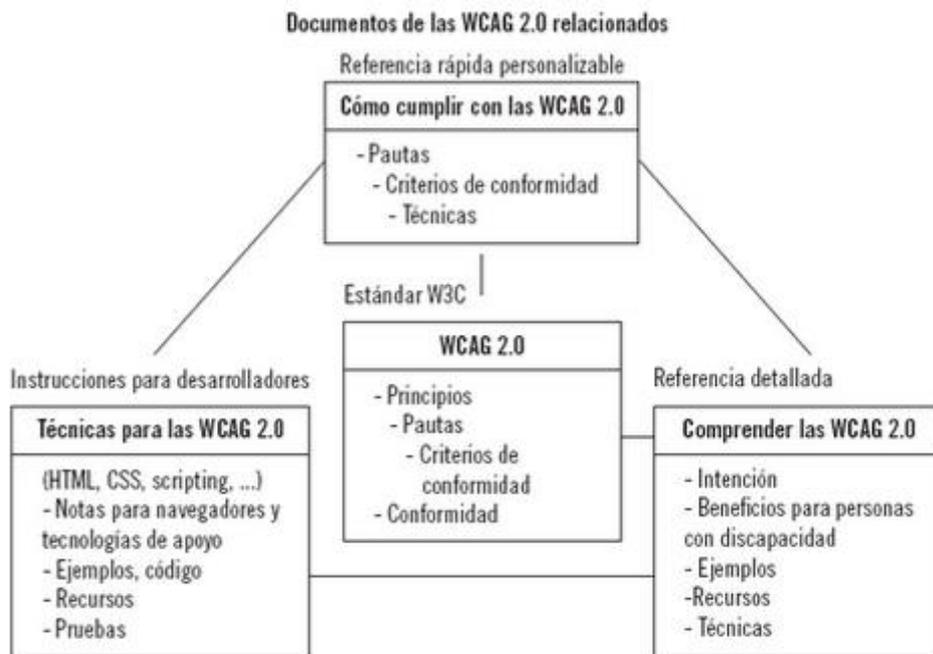


Figura 1.1: Documentos de la WCAG 2.0 relacionados. [WCAG, 2008]

## Criterios de conformidad en Accesibilidad Web

Las directrices de WCAG 2.0 establecen un conjunto de criterios de conformidad para verificar el cumplimiento de las pautas. Un criterio de conformidad es una afirmación comprobable que puede ser verdadera o falsa cuando se aplica a un contenido Web específico. (REVILLA MUÑOZ, 2013)

### Estructura del documento de las WCAG 2.0

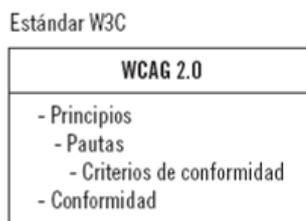


Figura 1.2: Estructura del documento de las WCAG 2.0. (REVILLA MUÑOZ, 2013)

Cada una de las pautas de la WCAG 2.0 tiene un grupo de criterios de conformidad. Y al mismo tiempo, cada uno de los criterios de conformidad pertenece a un nivel de conformidad. En el Nexa A, se describen los criterios de conformidad relacionados con cada una de las pautas. (WCAG 2.0, 2008)

## Requisitos de conformidad

Los requisitos de conformidad establecidos por la WCAG 2.0 (REVILLA MUÑOZ 2013) son:

- **Nivel de conformidad:** Debe cumplir completamente los requisitos de alguno de los niveles de conformidad, A, AA o AAA.
- **Páginas completas:** La conformidad y el nivel de conformidad se debe cumplir en la página completa. No se puede alcanzar si se excluye parte de la página.
- **Procesos completos:** Cuando una página es parte de un proceso (uno de los pasos de una secuencia), todas las páginas del proceso deben ser conformes con el nivel especificado.
- **Uso de tecnologías sólo con métodos que sean compatibles con la accesibilidad:** Para cumplir los criterios de conformidad deben utilizarse las tecnologías que sean compatibles con la accesibilidad. Si una información o funcionalidad está disponible de forma no accesible, debe proporcionarse una alternativa que sí sea compatible con la accesibilidad.
- **Sin interferencia:** Si las tecnologías se utilizan de una forma incompatible con la accesibilidad, o no cumplen los requisitos de conformidad, no se debe impedir a los usuarios acceder al contenido del resto de la página. Además, es necesario que la página Web, como un todo, siga cumpliendo con los requisitos de conformidad cuando cualquier tecnología de la que no se depende está activada o no es soportada por una aplicación de usuario.

Además, se aplican los siguientes criterios de conformidad a todo el contenido de la página:

- Control de audio.
- Teclado no bloqueado.
- Tres destellos o por debajo del umbral.

- Pausar, detener, ocultar.

## Metodología de evaluación de la accesibilidad en sitios Web

El grupo WAI de la W3C ofrece recursos y consejos para la evaluación de la accesibilidad de un sitio Web. En primer lugar, se puede utilizar el documento “Easy Checks – A First Review of Web Accessibility” (SHAWN LAWTON, 2014). Este documento proporciona una serie de pasos rápidos para ayudar a evaluar si un sitio Web es accesible. Permite realizar una revisión preliminar de los aspectos más evidentes, antes de un informe más profundo.

En el año 2012, la W3C publicó el documento de la “Metodología de Evaluación de Conformidad con la Accesibilidad en sitios Web” (WCAG-EM). En este documento se tiene también en cuenta aplicaciones Web y sitios Web para dispositivos móviles. Es independiente del tamaño del sitio Web, de la tecnología utilizada, de las herramientas de evaluación de accesibilidad, de navegadores Web o de productos de apoyo específicos. (WCAG-EM, 2012)

Normalmente, se selecciona una parte representativa de un sitio Web para su evaluación y no la totalidad del sitio. En el documento WCAG-EM se especifican los pasos a seguir para definir el alcance de la evaluación, auditar la muestra seleccionada y mostrar los resultados de la evaluación creando informes bien estructurados y uniformes.

A continuación, se describe el procedimiento para evaluar la conformidad en cinco pasos (WCAG-EM, 2012):

- **Paso 1: definir el alcance de la evaluación.** Definir las páginas a las que se va a aplicar la evaluación, el objetivo de la evaluación, y el nivel de conformidad deseado (A, AA o AAA), además de decidir las técnicas que se van a utilizar.
- **Paso 2: explorar el sitio Web objetivo de la evaluación.** Identificar las páginas y funcionalidades comunes en todo el sitio. Identificar los tipos de páginas y las tecnologías empleadas.
- **Paso 3: seleccionar una muestra representativa.** Incluir páginas representativas del sitio Web, como la página principal y las páginas comunes. También se deben incluir las más relevantes y otras al azar.

- **Paso 4: auditar la muestra seleccionada.** Comprobar el funcionamiento para distintos casos de uso. Utilizar las técnicas documentadas en la WCAG 2.0. Archivar las páginas auditadas y registrar las herramientas y métodos utilizados.
- **Paso 5: realizar un informe.** Documentar cada uno de los pasos y proporcionar una declaración de conformidad. Proporcionar una puntuación del funcionamiento y, si es posible, informes legibles por máquinas.

Realizar una declaración de conformidad no es obligatorio, es algo opcional. La conformidad se establece para páginas Web individuales. Pero la declaración de conformidad puede abarcar una página o varias páginas Web relacionadas. Si se decide hacer una declaración de conformidad, entonces esta declaración debe incluir los componentes exigidos y, si se desea, los componentes opcionales que se indican en el documento de la WCAG 2.0. (WCAG-EM, 2012)

Es importante llevar a cabo todos los pasos, aunque se puede variar el orden o realizar algunos pasos en paralelo si se solapan, El orden también puede depender del tipo de sitio Web que se evalúa.

### 2.1.6.9. Tecnologías donde la accesibilidad es aplicable

Los criterios de conformidad de las WCAG 2.0 son declaraciones generales que no especifican ninguna tecnología en concreto, pero que se pueden verificar. En otros documentos independientes se dan instrucciones sobre cómo cumplir los criterios de conformidad específicos de la tecnología utilizada, y consejos sobre la manera de interpretar los criterios de conformidad. (REVILLA MUÑOZ, 2013)

#### (X)HTML

El código HTML se encarga de la estructura y el contenido del documento. Simplemente utilizando las etiquetas HTML correctas y cumpliendo con el estándar ya se está creando contenido accesible. (W3C, 2016)

HTML5 permite usar nuevos elementos como <article>, <section>, <aside>, <header>, <footer>, o <nav>, que le dan más significado semántico a los contenidos. Pero para mejorar la accesibilidad no es suficiente que el lenguaje HTML aporte nuevas etiquetas para estructurar el

contenido semántico. También es necesario que los navegadores y tecnologías de apoyo hagan uso de esta información. (W3C, 2016)

En términos de accesibilidad, hay dos áreas clave que HTML5 tiene el potencial de cumplir: la posibilidad para las personas con discapacidad de usar nuevas características, y comunicación efectiva entre tecnología de apoyo y el navegador Web.

## CSS

Las hojas de estilo en cascada, CSS, se encargan de la apariencia de la página. Todos los navegadores actuales las soportan. Utilizar CSS para el estilo de la página, separando el diseño del contenido, es ya una muestra de accesibilidad. (W3C, 2016)

Algunas ventajas de utilizar CSS son (W3C, 2016):

- Control total del formato.
- Aporta gran flexibilidad al permitir cambiar la apariencia completa de un sitio Web o generar variantes para poder imprimir correctamente una página Web.
- Consistencia entre navegadores, permitiendo que una página Web se vea igual en todos ellos, permitiendo ajustar en el CSS las pequeñas diferencias entre navegadores.
- CSS permite escribir el contenido de forma secuencial, que es la forma en la lo leen los lectores de pantalla. Aun así, se pueden posicionar las cosas en cualquier parte de la pantalla.
- Permite que el texto cambie de tamaño fácilmente, lo que es muy útil para personas con problemas de visión.

## JavaScript

JavaScript es una tecnología del lado del cliente que se utiliza para manipular los elementos de la interfaz a través del DOM (Document Object Model) y para crear aplicaciones dinámicas.

Puede ocurrir que algunos navegadores Web y dispositivos no soporten JavaScript. Las guías WCAG 2.0 recomiendan que las aplicaciones Web funcionen, aunque se deshabilite JavaScript o no este soportado por el navegador. Otro posible problema es que se actualice la interfaz de la aplicación, y que no sea evidente que el cambio se haya producido. Este problema

afecta sobre todo a personas con discapacidades visuales. Los lectores de pantalla leen en forma secuencial, y cuando se produce un cambio en la interfaz, el usuario puede no darse cuenta del contenido nuevo. La solución es ofrecer una alternativa sin JavaScript para la aplicación Web y permitir que siga funcionando, aunque no se soporte JavaScript.

## **PDF**

El formato PDF (Portable Document Format) es un formato multiplataforma que puede ser utilizado en cualquier sistema operativo y es uno de los más extendidos para el intercambio de documentos.

Un documento PDF permite introducir texto, imágenes, fuentes, perfiles de color y elementos multimedia. También dispone de opciones de cifrado, firma digital, índices, formularios, enlaces (dentro del documento o a direcciones de Internet) y búsquedas.

Existen varios estándares PDF, como PDF/A (Archive) y PDF/X (eXchange) y PDF/UA (Universal Accessibility). El formato PDF/UA (ISO 14289-1:2012) es el estándar internacional para los PDF accesibles. La especificación describe los componentes obligatorios y los que no se deben incluir en un archivo PDF para que sea accesible al mayor número de público posible. En cualquier caso, todas las acciones que se deben llevar a cabo para hacer un PDF accesible se deben hacer independientemente del tipo de PDF que sea.

La principal característica de un PDF accesible es que esté bien etiquetado (tagged PDF). Un PDF etiquetado incluye el contenido, la estructura y el orden de lectura. De esta forma, se puede determinar el orden de los contenidos en el documento, independientemente de la apariencia y la distribución. Las aplicaciones pueden extraer el contenido de un documento etiquetado para presentarlo a los usuarios con minusvalías, navegando por la estructura jerárquica del documento y presentando los contenidos de cada nodo. Por esta razón es importante producir archivos PDF con una estructura jerárquica de los datos. Pero tener un documento PDF etiquetado no asegura la accesibilidad, pues el etiquetado, además de existir, debe ser correcto, para ello, se deben tener en cuenta todas las recomendaciones generales de la WCAG 2.0. (W3C, 2016)

La estructura lógica de un documento PDF es guardada aparte del contenido visible. Esta separación permite que el ordenamiento de los elementos lógicos sea completamente independiente del orden y localización de objetos gráficos en el documento.

## **XML/XSL**

El lenguaje XML (eXtensible Markup Language), lenguaje de marcas extensible, es desarrollado por el W3C para el almacenamiento e intercambio de información de forma estructurada. Permite definir gramáticas de nuevos lenguajes para estructurar documentos. El lenguaje XSL (eXtensible Stylesheet Language), lenguaje de hojas de estilo extensibles, permite describir cómo son codificados los archivos XML para ser presentados o transformados. (W3C, 2016)

La W3C dispone de una guía donde se especifican las recomendaciones para la creación de archivos XML accesibles, el documento es el “XML Accessibility Guidelines”.

Existen cuatro principios generales de diseño accesible en XML (W3C, 2016):

1. Asegurar que los creadores pueden ofrecer los contenidos con múltiples alternativas, como, por ejemplo, vídeo en diferentes formatos e incluso un texto alternativo al vídeo.
2. Crear lenguajes ricos semánticamente, mejorando la estructura del XML utilizando elementos y atributos que se correspondan semánticamente con el contenido.
3. Diseñar una interfaz de usuario accesible. Se debe asegurar que el lenguaje permita al usuario tener el control de la presentación en todo momento.
4. Documentar y exportar la semántica, asegurándose de que las personas entienden el diseño y su forma de utilizarlo. Hay que conseguir que los usuarios no tengan que hacer suposiciones sobre el propósito de los elementos

## **Reproducción multimedia**

Existen muchas opciones de reproductores multimedia. Un reproductor accesible es el que permite navegación por teclado, subtítulos y canal de audio secundario para audio-descripción. Cuando se ofrece contenido multimedia, el desarrollador debe elegir si ver el contenido a través de un reproductor embebido en una página Web o a través de un reproductor estándar. Introducir el reproductor en una página Web permite al usuario acceder al contenido

sin necesidad de abrir otra aplicación. Pero los reproductores estándares normalmente disponen de más opciones y mejor soporte de accesibilidad.

Uno de los problemas más importantes es la integración del vídeo. Para las personas con minusvalía auditiva el problema es la no disponibilidad de subtítulos de forma estandarizada para vídeos y audio. Para las personas con minusvalías visuales y motoras el mayor problema es la navegación con teclado para reproducir y controlar el vídeo.

HTML5 proporciona nuevas etiquetas como <video> y <audio>, que pueden crear y etiquetar botones. De esta forma se pueden usar atajos de teclado para acceder a ellos, y los lectores de pantalla dicen al usuario qué botones están disponibles. Esto significa que una persona con problemas de visión o movilidad puede usar el teclado fácilmente para iniciar o detener un vídeo de un sitio Web.

## Texto Plano

Se pueden crear archivos de texto plano accesibles siguiendo las guías de la WCAG 2.0. En este caso, se han establecido reglas para la identificación de párrafos, listas y encabezados (W3C, 2016):

- **Párrafos:** Van precedidos por una línea en blanco y finalizan con una o más líneas en blanco.
- **Listas:** Las listas son una secuencia de ítems. Un ítem de la lista es un párrafo que empieza con una marca, que debe ser igual para todas las listas desordenadas. Para las listas ordenadas se pueden usar letras o números terminados con un punto o cierre de paréntesis, y las marcas tienen que estar en orden ascendente (a, b, c, ... o 1, 2, 3, ...)
- **Encabezados:** El comienzo de un encabezado se indica con dos líneas en blanco y el final con una línea en blanco.

## SMIL

El lenguaje de integración y sincronización de archivos multimedia (SMIL) es un lenguaje de marcas, como HTML o XML. El grupo de trabajo del W3C, encargado de SMIL es el grupo de Actividad sobre Sincronización Multimedia (SYMM). Es una extensión del lenguaje XML que permite integrar en sitios Web presentaciones interactivas, con sonidos, textos e imágenes. (W3C, 2016)

SMIL está formado por un conjunto de etiquetas XML que sirven para describir distintos elementos (W3C, 2016):

- **Fuente de contenidos:** imágenes, gráficos, audio, texto o animaciones.
- **Sincronización:** el contenido puede reproducirse de forma secuencia o en paralelo.
- **Temporización:** permite controlar el inicio o detención de la reproducción.
- **Posición:** se puede ajustar su tamaño y posición en pantalla.
- **Enlaces:** permiten interactuar con el usuario.
- **Animaciones:** pueden cambiar dinámicamente las propiedades de los objetos.

Algunas de sus utilidades son la de crear presentaciones multimedia, subtitulación de vídeos o apoyo a personas con discapacidad.

## ARIA

WAI-ARIA (Web Accessibility Initiative – Accessible Rich Internet Applications) son especificaciones técnicas publicadas por el grupo WAI del W3C. Indican cómo mejorar la accesibilidad de páginas Web que usan contenidos dinámicos e interfaces de usuario avanzadas desarrolladas con Ajax, HTML, JavaScript y otras tecnologías relacionadas. La combinación de estas características permite crear soluciones que funcionen con las tecnologías de apoyo. ARIA proporciona a los desarrolladores la opción de mejorar la accesibilidad, agregando información semántica a las páginas Web y widgets. Widget es una pequeña aplicación que se puede incrustar dentro de una página Web o del navegador para que realice una función determinada. (W3C, 2016)

## Silverlight

Microsoft Silverlight es un complemento disponible para todos los navegadores Web, que permite utilizar contenidos multimedia desarrollados con esta tecnología. Sus funciones son muy parecidas a las ofrecidas con Flash y lo normal es que las aplicaciones desarrolladas con Silverlight se ejecuten dentro de un navegador Web. (W3C, 2016)

Silverlight es una tecnología que puede ser accesible, tiene soporte para tecnologías de apoyo y está disponible para todos los navegadores actuales (excepto para el mismo Microsoft, que discontinuó el soporte a esta tecnología, en la última versión de su navegador llamado

Microsoft Edge). Otro problema es que el soporte para accesibilidad se basa en herramientas de Microsoft que sólo están disponibles con sistemas operativos Windows.

### **2.1.7. DevOps**

DevOps es un proceso de desarrollo y entrega de software que se apoya en la automatización y monitoreo de los procesos de integración de software, testing, despliegue y cambios en la infraestructura. Esto se logra estableciendo un conjunto de prácticas y hábitos, junto con un entorno donde la construcción, testeo y liberación del software ocurren de manera continua, rápida, frecuente y confiable. (KIM y BEHR 2013)

De acuerdo a la literatura, la habilitación de DevOps requiere una serie de etapas (KIM y HUMBLE 2016):

1. Implementar Integración y Testing continuo (Continuous Integration, o CI), mediante la automatización de los procesos de compilación (building) y la automatización de las pruebas (testing automatizado).

2. Implementar Entrega y luego Implantación Continua (Continuous Delivery o CD y Continuous Deployment) mediante la automatización del proceso de instalación de las aplicaciones en los distintos ambientes (QA, Stress y en el caso de la Implantación Continua también en Producción). La Entrega Continua permite asegurar que las aplicaciones operarán adecuadamente en un ambiente productivo mediante un proceso automatizado de pruebas. La Implantación Continua es el paso que sigue, que permite automatizar totalmente la entrega en producción. No todas las organizaciones pueden instituir la Implantación Continua, e incluso en algunos casos puede no ser posible (por ejemplo, debido a marcos regulatorios).

3. Gestionar la configuración de los distintos ambientes, lo cual es necesario para lograr la Implantación Continua, mediante la implementación de tecnología que permita configurar y desplegar infraestructura de manera controlada (conocido también como Infrastructure as Code).

4. Monitorear y obtener retroalimentación a lo largo de toda la cadena de entrega (Continuous Monitoring), mediante herramientas de monitoreo en los ambientes productivos (alertas, análisis dinámico de logs, monitoreo de la plataforma).

La Figura 1.3 ilustra las etapas de la habilitación de las técnicas de DevOps, que muestra los niveles de automatización que se pueden alcanzar usando procesos y herramientas de DevOps.



Figura 1.3. Niveles de automatización en DevOps

El proceso de Integración Continua y su extensión, la Entrega Continua (Fowler), requieren la adopción de una cierta infraestructura que los soporte. Una arquitectura para dicha infraestructura de soporte es tal como lo ilustra la Figura 1.4 (GRUVER y MOUSER, 2015)

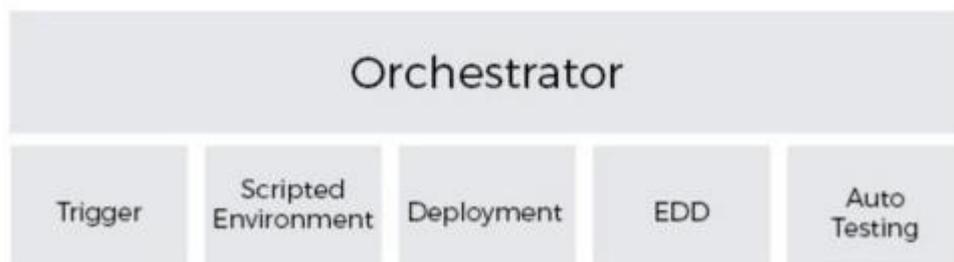


Figura 1.4. Arquitectura para soportar Entrega Continua. (GRUVER y MOUSER, 2015).

En el modelo propuesto por Gruver y Mouser, existe un orquestador (Orchestrator) que corresponde a un sistema que se encarga de controlar todo el proceso automatizado. Todo se dispara desde un agente externo (Trigger o acción externa), el cual corresponde normalmente a

un proceso de entrega de código fuente a través de un gestor de versiones (Por ejemplo: GIT, SVN) conectado al orquestador.

Una vez disparado el evento de entrega, se requiere que el orquestador decida si es necesario crear un nuevo ambiente. La creación de los ambientes está preconfigurada mediante un guion (script) que permite definir cómo instalar y configurar un ambiente donde se ejecutará el software. Una forma de adoptar este mecanismo es utilizar tecnología de contenedores (Containers), por ejemplo, mediante la tecnología Docker, el cual proyecto de código abierto que automatiza el despliegue de aplicaciones dentro de contenedores de software, proporcionando una capa adicional de abstracción y automatización de virtualización de aplicaciones en múltiples sistemas operativos (DOKER, 2022).

Lo importante es que mediante estos scripts es posible habilitar servidores configurados con todo lo necesario para soportar las aplicaciones a implantar (FARCIC, 2016).

Una vez obtenido el ambiente, se produce el despliegue de la aplicación, lo que implica instalar automáticamente las piezas y configurarlas para que empiecen a operar. Generalmente, es necesario hacer cambios en las bases de datos (GRUVER y MOUSER, 2015). Esto se administra mediante mecanismos de gestión de la evolución de la base de datos (AMBLER y SADALAGE 2006), los que corresponden a herramientas que calculan las diferencias que se producen en el modelo de datos y generan scripts con instrucciones para hacer cambio a nivel de estructura de tablas, y scripts que ejecutan la migración de datos correspondientes. Por último, la aplicación es testeada de forma automática, entregando retroalimentación al equipo técnico de operaciones sobre el estado de esta.

Para que esta arquitectura funcione, un concepto esencial es el de “Infraestructura como Código” (Infrastructure as Code), que se ilustra en la Figura 1.5. En la actualidad es posible generar ambientes virtuales que contienen todas las características necesarias para que opere el software. Lo relevante es que se puede describir en un script común todo lo necesario para operarlo, lo que permite simplemente aplicar parámetros de operación según el tipo de ambiente en que se quiere realizar la implantación. De este modo, los distintos ambientes de operación son idénticos (inmutables), lo que permite descartar errores debido a la mala configuración de los ambientes.

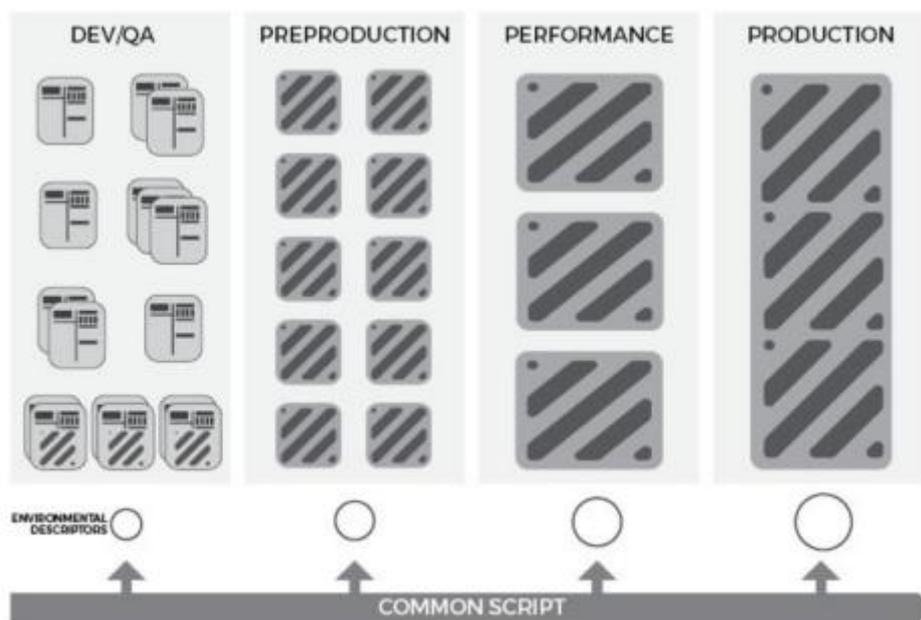


Figura 1.5. Infraestructura como código. (GRUVER y MOUSER, 2015)

## 2.2. Estado del Arte

### 2.2.1 Herramientas para validar la Accesibilidad Web

Para evaluar la accesibilidad existen una serie de herramientas que permiten identificar problemas de accesibilidad. Actualmente, no se puede automatizar totalmente la evaluación de la accesibilidad de un sitio Web y siempre será necesaria la intervención de una persona. (CARRERAS MONTOTO, 2007)

Existen varios tipos de herramientas de validación, algunas son servicios online, otras se basan en el uso del navegador instalándole algún complemento y otras son herramientas de escritorio que se instalan localmente.

Como parte del Estado del Arte se realizó un relevamiento de las herramientas de accesibilidad más importantes:

#### Validador HTML de W3C

Comprueba si el código HTML de la página Web está bien escrito gramaticalmente y cumple los estándares de la W3C. (WC3 MARKUP VALIDATOR, 2022)



Figura 1.6. W3C Markup Validator

### **Validador CSS de W3C**

Permite comprobar si una hoja de estilos cumple las especificaciones de la W3C. (CSS VALIDATOR, 2022)

### **Level Access**

Es una empresa que provee soluciones informáticas, software, consultoría y capacitación para mejorar la accesibilidad en aplicaciones Web para empresas corporativas, gubernamentales y de educación. Estas soluciones aseguran conformidad con las leyes, standars y guidelines de accesibilidad actuales (LEVELACC) (LEVEL ACCESS, 2022).

Level Access ofrece la herramienta AMP (Accessibility Management Platform), es un centro de comando para la organización, que resuelve la complejidad y los desafíos de la administración de toda la empresa para que pueda construir, probar y mantener software accesible de manera rápida y eficiente (LEVELACC). Ofrece reportes muy completos sobre los problemas de accesibilidad encontrados en dicha aplicación. Estos reportes contienen descripción de los problemas presentes y cuáles podrían ser sus posibles soluciones.

Además, Level Access es un servicio pago, ofrece diferentes paquetes y precios dependiendo el tamaño del sistema o empresa.



Figura 1.7. Logo Level Access

### **TAW**

Es una herramienta Web para el análisis e información del grado de accesibilidad que presentan otras Webs. El objetivo de esta Web es difundir la accesibilidad como requisito en el diseño y realización de páginas Web con el fin de permitir el acceso a todas las personas (TAW). Fue creada teniendo como referencia técnica, las pautas de accesibilidad al contenido Web

(WCAG 2.0) del W3C (W3C). Al igual que AMP de LevelAccess, las comprobaciones que realiza TAW (TAW, 2022) en el análisis se dividen en dos categorías:

- Automáticos. Problemas de accesibilidad que la herramienta detecta por si sola y que deben ser solucionados.
- Manuales. La herramienta señala la existencia de un posible problema que el evaluador/a debe confirmar o descartar.



Figura 1.8. TAW

### **OAW-Analizador WEB, Observatorio de Accesibilidad Web**

El Observatorio de Accesibilidad es un instrumento para evaluar la accesibilidad de una muestra de sitios Web de la República de Ecuador de acuerdo con las recomendaciones de las Pautas de Accesibilidad para el Contenido Web 2.0 (WCAG 2.0) (OAWEC).

La validación puede ser mediante código o URL. Al igual que TAW, se pueden obtener un reporte de errores que puede contener un sitio Web. Este validador es usado por el Observatorio de accesibilidad para hacer la monitorización y evaluación periódica de la accesibilidad de los portales del sector público. Solo está disponible online para el personal de la Administración Pública, a través del Servicio de Diagnóstico en línea de la Comunidad Accesibilidad, pero en 2020 se liberó y se puede instalar localmente. (OAW ECUADOR, 2022)



Figura 1.9. OAW Analizador Web

### **AChecker- Web Accessibility Checker**

AChecker es otro ejemplo de plataforma gratuita que ofrece una gran libertad al usuario para escoger el tipo de análisis a realizar, ya sea de una Web, un archivo HTML o un fragmento de código, así como el conjunto de directrices de accesibilidad a aplicar durante el análisis. (ACHECKER, 2022)

En el apartado de resultados se encuentra una forma de presentarlos similar a lo visto en TAW y OAW: una lista de los errores encontrados y problemas potenciales, así como su localización dentro del código fuente de la página. Otra opción interesante es la capacidad de descargar los resultados en varios formatos, y, además, escoger qué parte del análisis exportar.

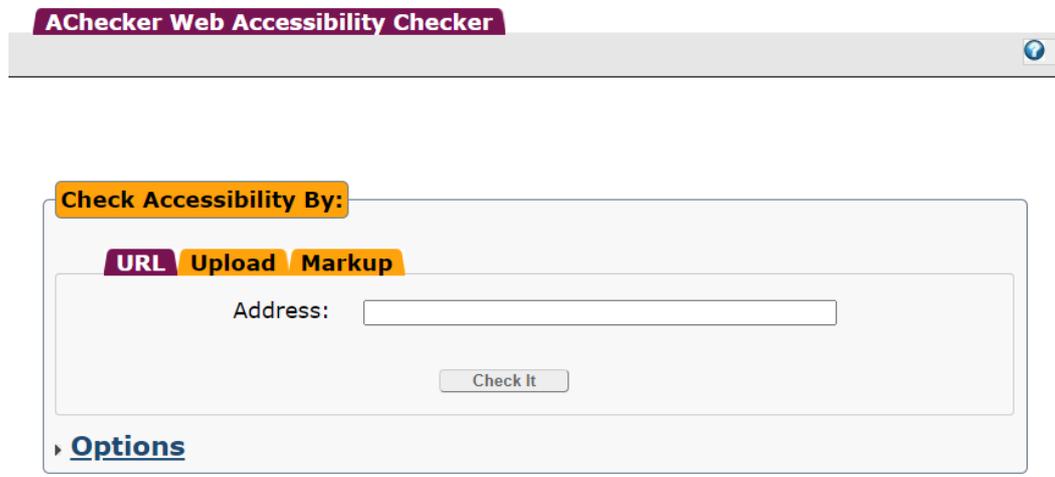


Figura 2. AChecker- Web Accessibility Checker

### Total Validator

Es un software que se instala en el equipo local y está disponible para Windows, Mac OS X y Linux. Existe una versión gratuita y otra versión profesional de pago. La versión gratuita permite validar el código HTML, validar la accesibilidad, comprobar los enlaces rotos y comprobar la ortografía. La versión profesional incluye más opciones, como validar uno o varios sitios Web completos, validación CSS, seleccionar las partes del sitio Web a validar, y más opciones de personalización del análisis. (TOTAL VALIDATOR 2022)



Figura 2.1. Logo Total Validator

### WAVE

La herramienta WAVE (Web Accessibility Tool) permite evaluar la accesibilidad de una página Web, ya sea indicando la URL de la página a analizar o subiendo directamente la página. Al igual que otras herramientas similares, permite elegir el nivel de análisis. Lo más interesante de esta herramienta son las opciones de configuración que posee y la información que proporciona en los informes de análisis. (WAVE 2022)

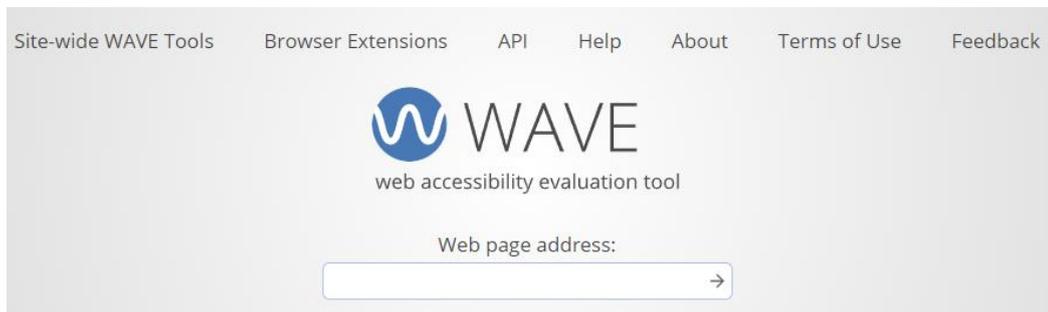


Figura 2.2. WAVE

### WCAG Color Contrast Checker

Esta extensión Firefox o Chrome permite comprobar si se cumplen las especificaciones, en cuanto a nivel de contraste, brillo y en la combinación de colores del texto y el fondo, de las recomendaciones WCAG 1.0, WCAG 2.0 Y WCAG 2.1. (CONSTRUCT CHECKER, 2022)



Figura 2.3. WCAG Color contrast checker

### Firebug

Es una de las herramientas más populares entre los desarrolladores Web. Permite inspeccionar elementos HTML y modificar estilos editando los archivos CSS y ver los resultados en tiempo real. También permite ver el código HTML de forma dinámica conforme se va generando o modificando con JavaScript. (FIREBUG, 2022)



Figura 2.4. Logo Firebug

## HERA

Es una utilidad desarrollada por la Fundación Sidar, basándose en las recomendaciones WCAG 1.0. Permite introducir la URI que se desea validar, realiza el análisis automático de la página e informa si se encuentran errores y qué puntos de verificación deben ser revisados manualmente. Además, indica los niveles de prioridad a los que pertenecen los errores y facilita la revisión manual proporcionando información acerca de los elementos a verificar e instrucciones sobre cómo realizar ese control. (HERA 2022)

Actualmente se encuentra Hera-FFX, la cual es una extensión para FireFox y tienen el mismo objetivo: realizar un análisis automático y dar soporte para realizar un análisis manual de las páginas Web.



Figura 2.5. HERA

### 2.2.2 Conclusiones

Al realizar el análisis de las distintas herramientas se observa que cada una presenta unas características y funcionalidades propias. Ya sea mediante un servicio Web, una aplicación de escritorio o una extensión para el navegador. Todas cumplen con el mismo objetivo de detectar problemas de accesibilidad en los sitios Webs.

Algunas de ellas se destacan por tener disponibles reportes, otras por indicar en que segmento de código está el error, pero con ninguna de estas herramientas es posible amoldarlo a un proceso de integración continua.

Otro dato y no menor que cabe resaltar es que estas herramientas están adecuadas a los niveles de accesibilidad globales, pero ya se ha visto, según la legislación argentina para que una Web sea accesible debe cumplir con el ultimo estándar dispuesto por la ONTI en el decreto 6/2019.

## 2.3. User Research

### 2.3.1 Informe de investigación

Durante diciembre de 2018 a junio de 2019 se realizó un estudio exploratorio realizado en la región del Cono Sur, en la que derivó en un informe llamado “El acceso a los servicios de la información y la comunicación y las personas con discapacidad”, el cual fue publicado en agosto de 2019.

Esta compilación fue realizada en base a una investigación llevada adelante por el equipo del Observatorio de la Discapacidad de la Universidad Nacional de Quilmes: Gabriela A. Toledo, Mirian Aguerrido, Antonella Rapanelli, Silvia G. Zor, Hugo Gallardo, Sergio E. García Canto y Liliana L. Pignataro, con la colaboración de Agueda Fernández y Lelia C. Schewe.

Para esta investigación se establecieron tres niveles de análisis conectados entre sí de manera interdependiente:

1 – El primer análisis contempla normativa internacional referida al derecho de las personas con discapacidad y la accesibilidad de los entornos virtuales, con sus especificaciones regionales y nacionales.

2 – Un segundo análisis abarca la vista retrospectiva del recorrido de la investigación, con la identificación de actores y el trabajo en terreno, que se extendió de diciembre de 2018 a junio de 2019, incluyendo las discusiones y redefiniciones metodológicas técnico y académicas propias de un proceso de investigación.

3 – El tercer nivel de análisis es el más relevante porque involucró la participación de personas, en particular usuarios con discapacidad, interactuando con las tecnologías digitales y su observación situada.

Cabe resaltar, que se estima que más de mil millones de personas viven con algún tipo de discapacidad; es decir, alrededor del 15% de la población mundial. (OMS 2021)

Esta investigación fue realizada en Argentina, Chile y Uruguay. En Argentina, la prevalencia de población con “alguna dificultad” de 6 años y más es 10,2% (del total de la población argentina). En términos absolutos, se corresponde con una estimación de 3.571.983 personas, según datos del INDEC 2018. (INDEC 2018)

En primera instancia para validaciones automáticas (de navegaciones en páginas de inicio y en profundidad) se adoptó TAW20 como herramienta de validación online para verificar el cumplimiento de los criterios del WCAG 2.0 en las 389 páginas WEB propuestas.

Luego siguieron las validaciones por experiencia de usuario, en donde se definió la utilización de un cuestionario de Formularios de Google (Google Forms), para perfil de los usuarios y consentimiento, combinando con planillas Excel con Dimensiones a observar desde los cuatro Principios de accesibilidad Web (perceptible, operable, comprensible y robusto), cuestionario guía semiestructurado para entrevista y planilla de observación del contexto de la validación.

A medida que se avanzó en la ejecución de las validaciones automáticas de las 389 páginas Web seleccionadas, se realizaron lecturas parciales de los datos recogidos, en función de los cuatro Principios de accesibilidad Web publicados por el W3C (World Wide Web Consortium).

Al principio se seleccionó un total de 10 sitios Web en cada país, considerados de posible interés y necesidad para los usuarios con discapacidad, que significaron una primera aproximación a las características de accesibilidad de las páginas Web ofrecidas a los usuarios validadores (personas con discapacidad) como espacios a navegar.

De modo de lograr una visión general de las validaciones automáticas realizadas, se presenta la siguiente tabla, donde se presenta la cantidad de reportes de fallas recibidos:

Tabla 1.2. Evaluaciones automáticas realizadas con TAW sobre 389 páginas Web

	Número total de páginas Web validadas	Número de páginas por país Argentina, Chile, Uruguay	Número de reportes
Primera instancia de ponderación	30	10/10/10	1634
* Páginas globales o generales	2		
Segunda instancia de ponderación	89	26/27/36 más 2 pág.globales	5196
Validaciones automáticas en profundidad	45	13/15/17	
Tercer instancia de ponderación	255	93/97/63	14872
<b>Totales finales</b>	<b>389</b>		<b>20068</b>

La tabla que muestra las cantidades de páginas validadas por país en cada instancia de la investigación y la cantidad de reportes de fallas resultantes.

En la primera instancia de ponderación se analizaron 30 páginas, 10 por cada uno de los tres países, que resultaron en 1634 reportes.

En una segunda instancia de ponderación se analizaron 89 páginas (26 por Argentina, 27 por Chile y 36 por Uruguay, más Google y Youtube como 2 páginas globales), resultando en 5196 reportes. En las validaciones automáticas en profundidad se analizaron 45 páginas (13 por Argentina, 15 por Chile y 17 por Uruguay).

En la tercera instancia de ponderación se analizaron 255 páginas (93 por Argentina, 97 por Chile y 63 por Uruguay) resultando en un total de 14.872 reportes. El total de páginas

relevadas fue de 389, resultado de la suma de tres recortes: 89 (que incluye a las 30 primeras+2) +45+255.

El número total final de reportes fue de 20.068 (que incluye 14872 de la tercer instancia + 5196 de la segunda instancia, que incluye a la primera también).

De los cuatro principios observados, la mayor dificultad se encuentra en la perceptibilidad y la robustez. Se puede observar en los siguientes gráficos por instancia:

**Primera instancia**

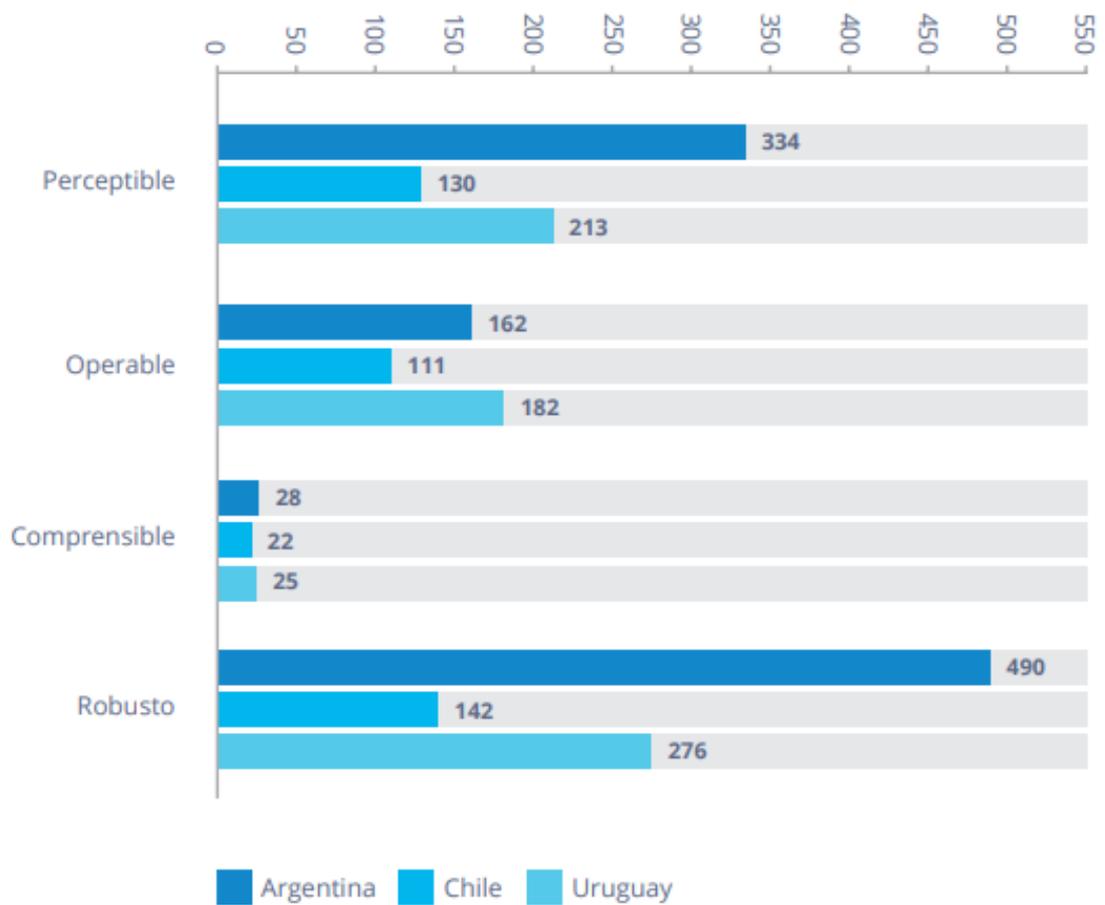


Figura 2.6. Cantidad de problemas por principio (Resultados de evaluaciones automáticas realizadas con TAW en primera instancia).

### Segunda instancia

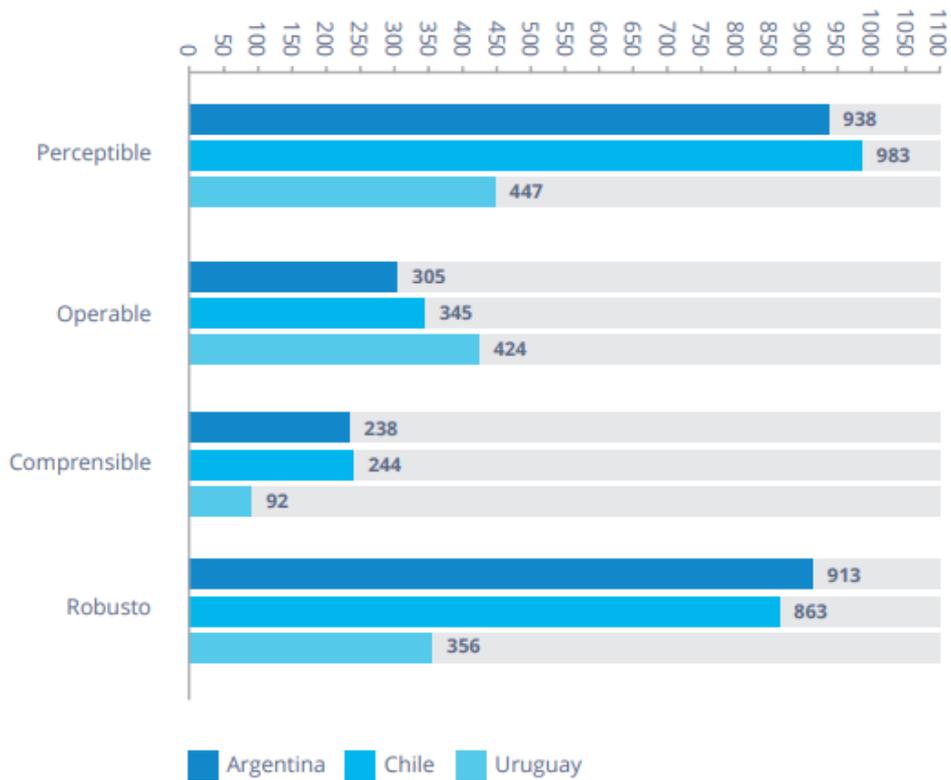


Figura 2.7. Cantidad de problemas por principio (Resultados de evaluaciones automáticas realizadas con TAW en segunda instancia).

### Tercera instancia

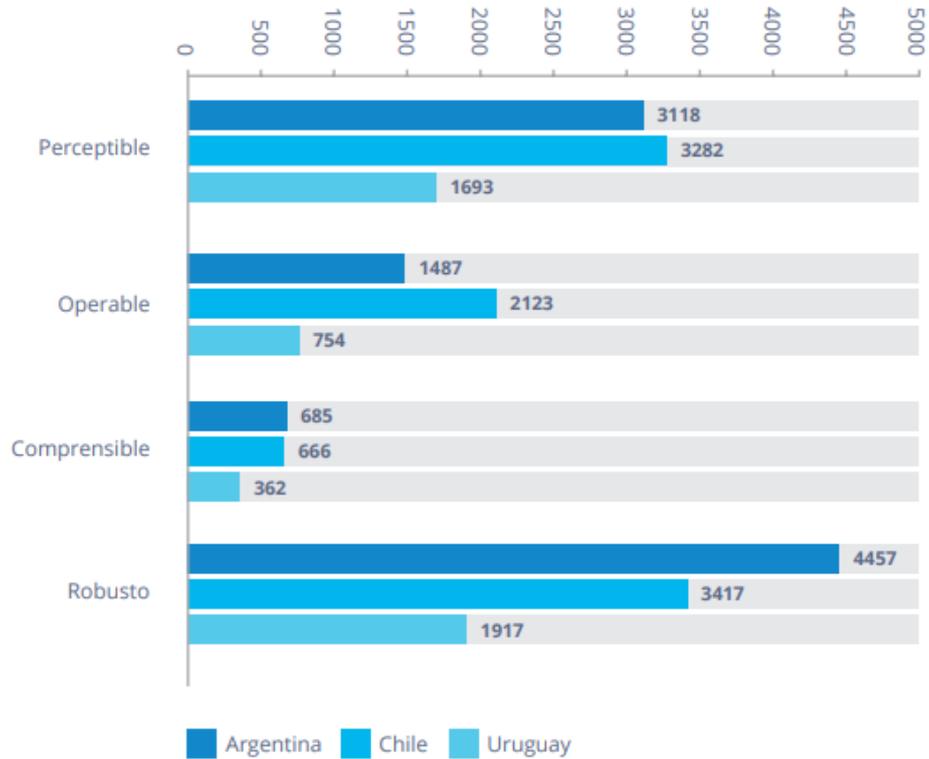


Figura 2.8. Cantidad de problemas por principio (Resultados de evaluaciones automáticas realizadas con TAW en tercera instancia).

Finalmente, el cuadro con cantidad de Fallas o Problemas por Principio del total de 389 páginas validadas.

Tabla 1.3. Resultados obtenidos por país

Principio	Argentina	Chile	Uruguay	Globales
Perceptible	3118	3282	1693	197
Operable	1487	2123	754	58
Comprensible	685	666	362	35
Robusto	4457	3417	1917	1415
<b>TOTAL</b>	<b>9747</b>	<b>9488</b>	<b>4726</b>	<b>1705</b>

En este cuadro se refleja la de problemas por principio del total de 389 páginas validadas. En Argentina se contabilizaron 3118 problemas del principio Perceptible, 1487 problemas del principio Operable, 685 problemas del principio Comprensible y 4457 problemas del principio Robusto, arrojando un total de 9147 problemas para Argentina.

En Chile se contabilizaron 3282 problemas del principio Perceptible, 2123 problemas del principio Operable, 666 problemas del principio Comprensible y 3417 problemas del principio Robusto, arrojando un total de 9488 problemas para Chile.

En Uruguay se contabilizaron 1693 problemas del principio Perceptible, 754 problemas del principio Operable, 362 problemas del principio Comprensible y 1917 problemas del principio Robusto, arrojando un total de 4726 problemas para Uruguay.

Sobre las páginas globales relevadas (Google y Youtube) se registraron 197 errores del principio Perceptible, 58 del principio Operable, 35 del principio Comprensible y 1415 del principio Robusto, arrojando un total de 1705 problemas.

Como conclusión se observa que los informes de los tres países señalan falencias en la implementación de cuestiones de accesibilidad, indicando incumplimientos específicos sobre los derechos de las personas con discapacidad en general. En lo que a Argentina refiere, y sabiendo de la existencia de la Ley de Accesibilidad de la Información en las Páginas Web (Ley 26.653), en el informe se puede ver su incumplimiento a pesar de haber sido reglamentada en el año 2013, al mostrar la falta de accesibilidad en la gran mayoría de los contenidos Web.

## 2.3.2 User Persona

WCARG busca ser un proyecto que comprenda las necesidades tanto del usuario de una aplicación web, como del grupo de desarrollo y para ello es importante ponerse en el lugar de estos actores. Por tal motivo, se desarrollan los User Persona, de un desarrollador DevOps, de un trabajador tecnológico que posee una discapacidad y de una persona con discapacidad.

### 2.3.2.1. Desarrollador DevOps



Figura 2.9. User persona Desarrollador DevOps

El desarrollador de DevOps tiene como objetivo incorporar procesos, herramientas y metodologías para equilibrar las necesidades durante todo el ciclo de vida del desarrollo de software, desde la programación y la implementación hasta el mantenimiento y las actualizaciones.

En muchos casos, el equipo de desarrollo no sabe la totalidad sobre buenas prácticas de desarrollo o vulnerabilidades, por eso utilizan herramientas como Sonar en donde las mismas ayudan con este proceso. Lo mismo ocurre con la accesibilidad web, por eso WCARG es el aliado perfecto para ayudar al grupo de desarrollo a detectar errores de accesibilidad y como resolverlos.

### 2.3.2.2. Trabajador tecnológico que posee una discapacidad



Figura 3. User persona Trabajador tecnológico que posee una discapacidad

El trabajador tecnológico, al igual que un desarrollador, frecuentemente no tiene conocimientos sobre la accesibilidad web y como la misma puede cambiar y mejorar su forma de trabajar. Por ello, cuanto más accesibles sean las herramientas con las cuales trabajan y en el caso de tener alguna discapacidad como es el caso de este User Persona, ya no tendrá el obstáculo que impide o dificulta la realización de una determinada tarea, mejorando radicalmente su desempeño profesional.

### 2.3.2.3. Usuario de páginas web que posee una discapacidad



Figura 3.1. User persona Usuario de páginas web

Al entender que la accesibilidad web permite el acceso de todas las personas al conocimiento y a la información, aumenta la posibilidad de estar en igualdad y equidad de condiciones, sobre todo en las oportunidades de desarrollo personal y profesional.

En este caso estimulará y mejorará la confianza del usuario final al uso de herramientas tecnológicas, aumentando de manera considerable la audiencia en el uso de las mismas.

### **3. Descripción**

Para intentar mitigar los efectos negativos que tienen las páginas web no accesibles, tanto de empresas privadas como de entes del estado, se propone el desarrollo de WCARG, una librería desarrollada en NodeJs, para que la misma sea integrada al marco de trabajo DevOps y conseguir evaluar páginas web en ambientes bajos y poder llegar a producción de acuerdo con las normas vigentes en Argentina dispuestas por la ONTI.

#### **3.1. Requerimientos Funcionales**

Los requerimientos funcionales de un proyecto de software son las descripciones de las funcionalidades que deberá realizar el mismo. Se refiere tanto a manejo de datos como acciones que el usuario puede ejecutar sobre el programa. Estos requerimientos se definen para permitir la implementación de la solución. A continuación, se detallan las historias de usuario de WCARG, explicaciones cortas que siguen un formato definido de las funciones de la librería, desde el punto de vista del usuario final.

##### **3.1.2. Épicas**

Las historias de usuario llamadas épicas son las que, generalmente, se realizan a más alto nivel y no incluyen detalles.

###### **3.1.2.1. Integración continua**

Como usuario desarrollador quiero poder integrar WCARG a mi proceso de integración continua.

###### **3.1.2.2. Validar criterios WCAG 2.0**

Como usuario desarrollador quiero analizar los criterios de aceptación de nivel A y nivel AA alcanzados por la disposición de la ONTI en Argentina.

###### **3.1.2.3. Página accesible**

Como usuario desarrollador quiero analizar si la página cumple con 30 criterios o más según la disposición de la ONTI en Argentina y así determinar si la página es accesible o no.

### **3.1.2.4. Información de criterios**

Como usuario desarrollador quiero saber qué criterios de aceptación tienen errores y bajo qué principio. Además, poder identificar bajo en que segmento de código se encuentra el error y como resolverlo.

### **3.1.2.5. Warnings**

Como usuario desarrollador quiero no solo identificar errores de accesibilidad, sino que también posibles errores que deben validados de forma manual.

### **3.1.2.6. Reporte**

Como usuario desarrollador quiero un informe detallado de todos los errores de accesibilidad encontrado en la página.

### **3.1.2.7. Múltiples URLs**

Como usuario desarrollador quiero poder analizar más de una página por vez.

### **3.1.2.6. Pipeline**

Como usuario desarrollador quiero poder llamar a la librería y configurar que detenga o no el pipeline si la web no es accesible.

## **3.2. Elección de librerías**

Las herramientas de evaluación son aplicaciones de escritorio que se pueden descargar e instalar en la computadora del usuario o aplicaciones web que se pueden acceder y emplear a través de un navegador. (LUJAN MORA 2006)

Hoy en día, estas herramientas permiten realizar un testeado de las aplicaciones una vez ya lanzadas en producción. Existen librerías que permiten realizar pruebas de accesibilidad web como parte del proceso de desarrollo. En caso de utilizar un proceso de integración continua (CI/CD), es posible no solo incluir tests unitarios, sino tests completos de accesibilidad. De esta manera, se puede detectar las vulnerabilidades pasadas por alto durante el proceso de crear y lanzar un sitio web a producción, evitando lanzar código con errores y un sitio no accesible. Además, se podrá utilizar los resultados de estos tests automatizados, para crear informes para el equipo de producto y desarrollo.

De esta manera, se realiza un análisis de tres librerías para generar pruebas automatizadas de accesibilidad e integrarlas a WCARG:

- Axe-core
- Pa1ly
- Lighthouse

Se realiza un análisis de las tres herramientas, evaluando siete ítems que se observan en la tabla:

Tabla 1.4. Comparativa de librerías

ITEMS	PA11Y	AXE CORE	LIGHTHOUSE
FACILIDAD DE CONFIGURACIÓN	EXCELENTE	EXCELENTE	EXCELENTE
USABILIDAD	EXCELENTE	MUY BUENA	MUY BUENA
RECURSOS	EXCELENTE	MUY BUENA	MALA
FEEDBACK	EXCELENTE	EXCELENTE	BUENO
COSTO	EXCELENTE	BUENO	EXCELENTE
LIMITACIONES	POCAS	POCAS	MEDIA
FORTALEZAS	MUCHAS	MUCHAS	MEDIA

### 3.2.1. Facilidad de configuración

Pa1ly requiere el uso de una interfaz de línea de comandos, como una terminal o Windows Command Prompt (CMD) para la instalación, las pruebas y los informes. La instalación de PA11y es relativamente sencilla. Requiere NodeJs para la instalación. Una vez que se instala NodeJs, solo se requiere un comando para instalar Pa1ly y tenerlo disponible para su uso: `npm install -g pa1ly`.

En cuanto a la disponibilidad, Pa1ly puede ser utilizada tanto en un entorno de Mac OS, Linux y Windows usando una instalación de paquete de NodeJs diferente.

Con respecto a axe-core, este requiere tener instalado el paquete de NodeJs y el comando `npm install axe-core`.

En su versión de Axe Cli (Axe-core/cli), mandatoriamente necesita NodeJs 6+. De forma predeterminada, axe-cli ejecuta Chrome en modo headless (autónomo), que requiere Chrome 59 o superior. Luego la realización del comando `npm install @axe-core/cli -g`.

Axe, en su versión como extensión de navegador está disponible para Chrome, Firefox y Edge, tarda menos de un minuto en instalarse. Todo lo que tiene que hacer es ir a la página a probar y en extensión de su navegador presionar el botón “Agregar a...”. Después de esto, puede abrir las herramientas de desarrollo para comenzar una auditoría de accesibilidad.

También se encuentra la versión Axe Accessibility Linter, es sencillo para los usuarios de VS Code. Un solo clic instala el complemento universal desde Visual Studio Marketplace o la pestaña Extensiones en el editor. No hay configuraciones requeridas para comenzar a testear.

Lighthouse está disponible automáticamente en Chrome, sin configuración ni extensiones para instalar, y se puede usar para probar sitios locales y páginas autenticadas. Se puede acceder a Lighthouse abriendo Chrome Developer Tools y eligiendo la pestaña Lighthouse en la parte superior.

Lighthouse también está disponible como una extensión de navegador para Chrome y Firefox. Google desaconseja usar Lighthouse de esta manera ya que la extensión no permite probar sitios locales y páginas autenticadas, que es algo que puede hacer en DevTools. La extensión también ejecuta el conjunto completo de auditorías sin permitir que el usuario elija las que desea. Desafortunadamente, esto limita al usuario a usar Lighthouse solo en Chrome en muchas circunstancias. Además, ejecutar Lighthouse en una aplicación web de varias páginas podría volverse tedioso.

### **3.2.2. Usabilidad**

Usar Pally se puede ejecutar una auditoría predeterminada básica con un comando usando solo la URL del sitio web que se desea auditar, como `pally http://ejemplo.com`. La prueba contra un archivo local se puede realizar usando `pally ./directorio/archivo.html`. Se puede optar por que el informe en la terminal se imprima en la pantalla, o puede usar el indicador `--reporter` para que se genere un archivo .CSV, por ejemplo, `pally --reporter csv http://ejemplo.com`. También soporta otros formatos de archivos como: cli, html, json y tsv.

Los indicadores adicionales para diferentes opciones están bien documentados y disponibles para su uso mediante el indicador `--help` en la línea de comando para obtener una lista de los mismos. Estos parámetros permiten hacer que los informes sean más específicos y convierte a Pa1ly en un conjunto de herramientas más completo, pero también requiere más conocimientos y experiencia para la configuración.

Pa1ly también tiene un proyecto llamado Pa1ly Dashboard que se mantiene activamente y le da a Pa1ly una interfaz de usuario que alguien que no se sienta cómodo con la línea de comandos podría usar. Pero si es necesario que alguien con experiencia en línea de comandos lo configure. También hay una aplicación de escritorio llamada Koa1ly que le da a Pa1ly una interfaz de aplicación de escritorio. Esta es gratuita y de código abierto como Pa1ly.

Para realizar una auditoría con Axe-core por la línea de comando, es tan simple como ejecutar `axe http://ejemplo.com`. Al igual que Pa1ly soporta varias URLs, puede ser deshabilitadas ciertas reglas como por ejemplo con el comando `axe www.deque.com --rules color-contrast,html-has-lang`. En cuanto al informe puede ser impreso en pantalla o que recopile los datos a un archivo formato json.

La extensión del navegador Axe es de uso muy sencillo. Una vez que abra la pestaña Axe en sus herramientas de desarrollador, iniciar una auditoría de accesibilidad es tan simple como presionar el botón "Analizar". Después de que la extensión analice una página de su sitio, verá una lista de problemas y una descripción de cada uno. Cada descripción de problema proporciona un enlace "Más información" que lo lleva al sitio de la Universidad Deque donde puede obtener información sobre cómo solucionar ese problema en particular. También puede notar los diversos enlaces dentro del panel para registrarse en Axe Beta, una herramienta de accesibilidad gratuita que le permite realizar auditorías guiadas. La única desventaja de esto es que requiere su nombre, correo electrónico comercial y nombre de la empresa.

Dado que se trata de una extensión del navegador, una de las desventajas es que debe analizar manualmente cada página individual. Esto puede llevar mucho tiempo si tiene muchas páginas únicas. Una vez que resuelva un problema de accesibilidad, debe volver a esa página en particular donde se arregló y analizar la página nuevamente para verificar que se solucionó el problema.

En lo que respecta a los comentarios y la documentación, la comunidad Axe es extremadamente activa, y si tiene preguntas o comentarios sobre la extensión del navegador Chrome o Firefox Axe, puede crear una cuenta con la comunidad Gitter para deque-labs/axe-core.

Con respecto a Axe linter, tras la instalación, el linter inmediatamente comienza a analizar su código. Los resultados se enumeran en la pestaña "Problemas" del editor:

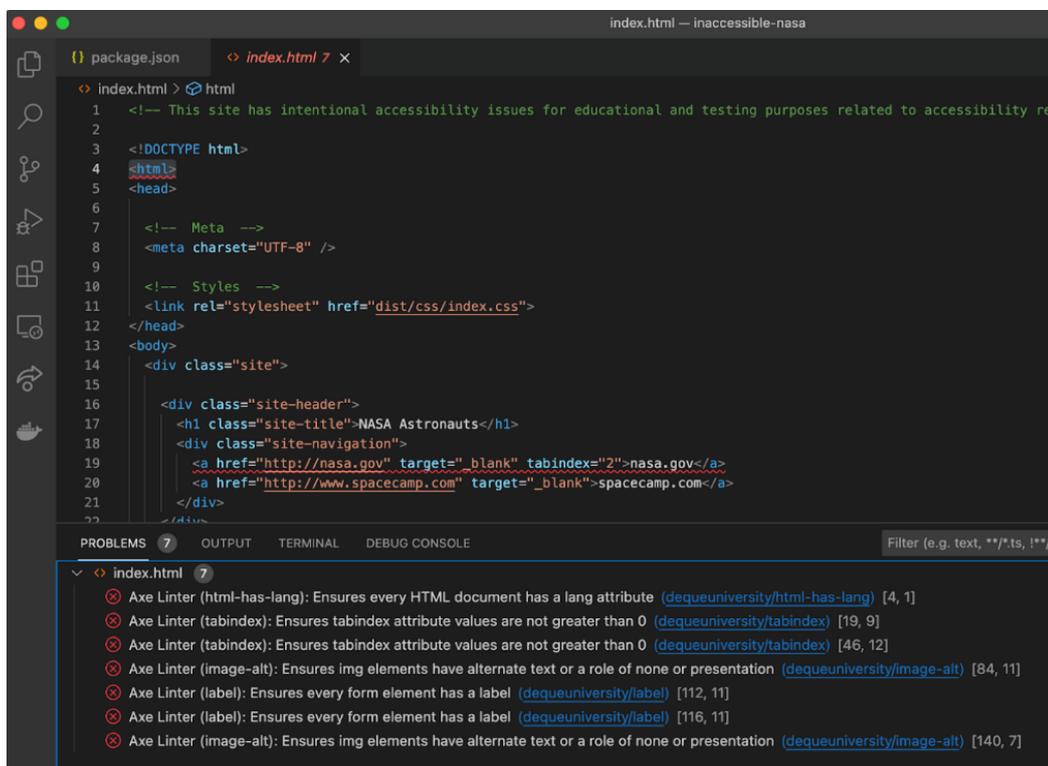


Figura 3.2. Librería Axe Linter

Las reglas con las que comprueba el linter también se pueden configurar en un archivo axe-linter.yml creado por el usuario. Puede habilitar/deshabilitar reglas específicas o grupos completos de WCAG para ajustar las necesidades de su proyecto.

Las reglas se enumeran claramente en la página de este complemento. Cada uno está vinculado al sitio de Deque como referencia. Deque también mantiene una página de informe de problemas y tiene un canal disponible públicamente para obtener comentarios aún más rápidos.

La pestaña Lighthouse en Chrome Devtools tiene una interfaz muy sencilla que es fácil de usar. Le permite elegir ejecutar solo una auditoría de accesibilidad o ejecutar auditorías adicionales, que incluyen Rendimiento, Aplicación web progresiva, Mejores prácticas y SEO. También permite al usuario elegir entre ejecutar la auditoría en el escritorio o en un dispositivo móvil simulado. Una vez que se hayan realizado esas elecciones, al presionar Generar informe se ejecutarán las auditorías seleccionadas directamente en el navegador.

Cuando solo se ejecuta la auditoría de accesibilidad, el informe se genera rápidamente y se muestra directamente en la pestaña Lighthouse en DevTools. El informe incluye una lista de los problemas de accesibilidad que se encontraron y qué elementos fallaron en cada prueba de accesibilidad. Cada problema también incluye una breve descripción del problema y enlaces a más información sobre esos problemas en la documentación de Lighthouse de Google.

Desafortunadamente, la documentación para usar Lighthouse esta desactualizada. En la pestaña "Auditorías" en DevTools, es donde se ubicaban anteriormente los controles de Lighthouse. Ahora, sin embargo, Lighthouse está disponible en su propia pestaña. Eso es algo que un usuario puede descubrir con relativa rapidez, pero un punto negativo es que la documentación no se haya mantenida correctamente.

Lighthouse también tiene algo llamado Lighthouse Viewer, que está destinado a permitir que el usuario comparta la auditoría de accesibilidad, ya sea descargando un archivo JSON de la página de auditoría.

Una desventaja de usar Lighthouse en el navegador como Axe en su extensión, es que solo se puede auditar una página a la vez, lo que hace que el proceso sea engorroso para sitios con una gran cantidad de páginas. Las auditorías también deben volver a ejecutarse manualmente si es necesario volver a verificar una página, como cuando se corrige un error. Aquí es donde la ejecución de Lighthouse mediante programación en lugar de en el navegador tiene ventajas.

### **3.2.3. Recursos**

Pa1ly utiliza fuentes WCAG directamente en sus comentarios. El informe enumerará las pautas de las WCAG para el error en cuestión directamente en el informe. Los usuarios

pueden usar esos comentarios y seguirlos directamente hasta el criterio de éxito de la guía en cuestión dentro de la documentación estándar de las WCAG.

Aunque es importante tener en cuenta que la información puede ser un poco difícil de entender al principio, ya que utiliza puntos por espacios y guiones bajos en lugar de puntos en la cadena de texto. Por ejemplo, si el informe enumerara este "WCAG2AA.Principle1.Guideline1\_1.1\_1\_1.H37", eso significaría que el error de la guía es WCAG 2.0 AA Principio 1, Guía 1.1, Criterio de Conformidad 1.1.1.

```

• Error: Img element missing an alt attribute. Use the alt attribute to specify a short text alternative.
  └─ WCAG2AA.Principle1.Guideline1_1.1_1_1.H37
     └─ #columns > div:nth-child(3) > img
        └─ 
    
```

Figura 3.3. Informe de Pally

La extensión del navegador Axe utiliza una lista de reglas Axe-core que cubre WCAG 2.0 y WCAG 2.1 para A y AA. Cada problema está dentro de la pestaña de "Herramientas de Desarrollo", y se verá las etiquetas WCAG a la derecha del panel que hacen referencia a una guía en particular. Además, cada página de "Más información" tiene una sección de "Recursos" que hace referencia a la documentación de las WCAG del W3C.

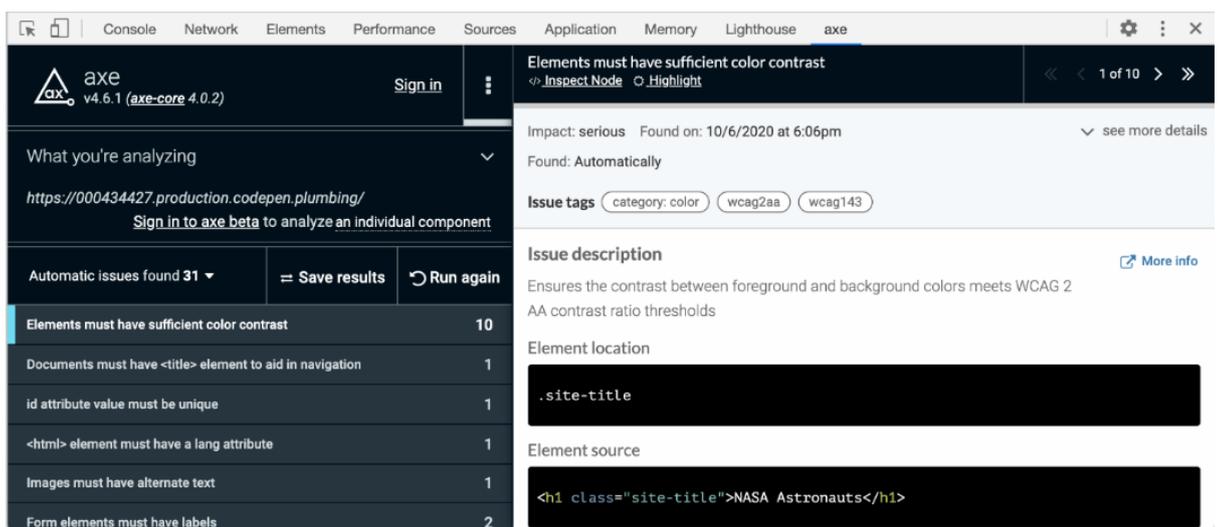


Figura 3.4. Informe de Axe

La característica más destacada del informe de auditoría de accesibilidad de Lighthouse es la puntuación de accesibilidad, que se encuentra en la parte superior del informe. Este puntaje es un promedio ponderado de todas las auditorías de accesibilidad que Lighthouse ejecuta en la página actual y se basa en las evaluaciones de impacto del usuario de Axe de Deque (que Deque usa para su propia extensión de navegador Axe). Estas evaluaciones de impacto del usuario se basan directamente en WCAG 2.0 y WCAG 2.1 para A y AA y son un recurso confiable.

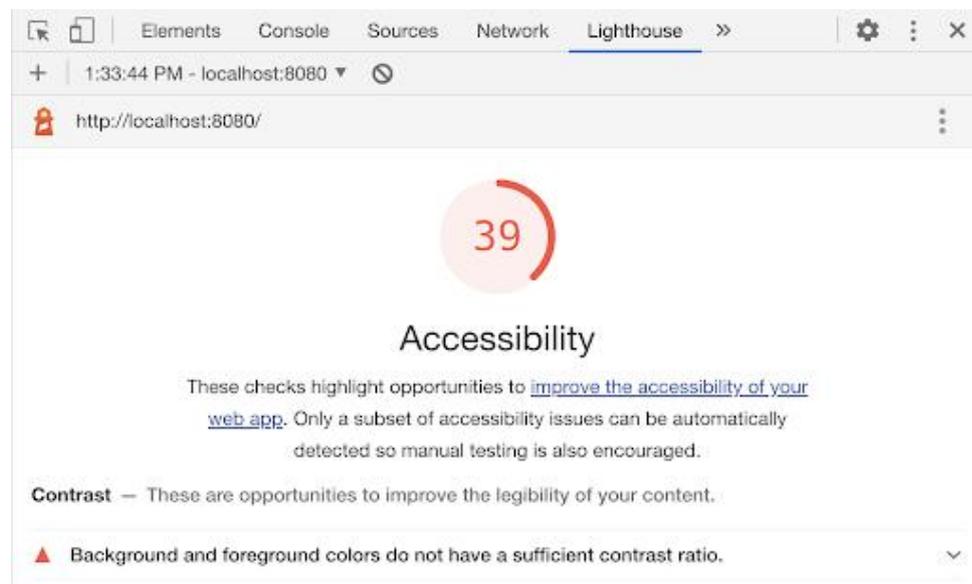


Figura 3.5. Informe de Lighthouse

A pesar de que Lighthouse es basado en la lista de Axe no detecta los problemas que la extensión Axe puede detectar, por lo que la implementación no es consistente. Además, en comparación con Axe, Lighthouse no permite que el usuario sepa qué reglas de las WCAG se están violando.

Además, todos los enlaces en el informe que conducen a más información sobre los problemas de accesibilidad individuales dicen "Más información". Esta es una mala práctica de accesibilidad que va en contra de WCAG 2.4.4 (Nivel A), que dice que el propósito de un enlace debe poder determinarse solo a partir del texto del enlace o de su contexto de enlace determinado programáticamente.

### 3.2.4. Feedback

Pally la configuración predeterminada "lista para usar", provee un exhaustivo análisis de accesibilidad para la WCAGAA, informando los problemas, y los comentarios sobre esos problemas de forma precisa. También ofrece una amplia gama de opciones de configuración que cambian lo que puede auditar, lo que puede eximir de su auditoría y cómo audita (como una sola página o todo el sitio). Se puede usar Pally como runner con otras herramientas como Axe. La configuración de reglas adicionales y recursos de ejecución le permite crear una configuración más completa que puede brindar un informe de comentarios de auditoría más informativo que la herramienta predeterminada.

Pally ofrece referencias de violación de criterios WCAG 2.1 muy exactas y sugiere cambios para esos problemas. Sin embargo, no siempre explica estas correcciones de una manera que un novato en accesibilidad entendería de inmediato. Le proporciona el contexto del elemento en violación, así como una cadena de texto para ubicarse en el elemento, pero no es tan conveniente como un enlace en el que se puede hacer clic directamente al elemento en sí, como ofrece Axe-core.

Al igual que Pally, tanto la extensión del navegador de Axe como la versión por comandos detecta de forma correcta los errores de accesibilidad. Ambas herramientas afirman tener 0 falsos positivos. Y si se llega a encontrar falsos positivos, puede enviarlos como errores en el repositorio de código abierto de Github para Axe-core. Además de tener reglas sobre las WCAG 2.0 y algunas WCAG 2.1, también usan una lista de reglas básicas para las mejores prácticas que ha adoptado la comunidad de accesibilidad.

Además, para la extensión de Axe una vez completada la auditoría, al hacer clic en un problema se mostrará un área de descripción que explica el problema en relación con la guía WCAG y tiene información sobre cómo resolver el problema. Junto con la descripción, cada problema mostrará una calificación de impacto codificada por colores de "menor", "moderado", "grave" o "crítico". Cada descripción de problema también proporciona enlaces "Más información" para obtener más información sobre cómo solucionar cada problema con muchos recursos adicionales.

Con respecto a Axe linter, Deque Systems menciona que su flujo de trabajo previsto es usar el linter con la extensión del navegador axe para detectar la mayoría de los problemas. La salida de linter es descriptiva del problema que ha marcado. Al hacer clic en el problema en la pestaña "Problemas" de VS Code, lo lleva directamente a la línea de código con el error. Además, se incluye un enlace para navegar directamente a una explicación detallada en el sitio de la Universidad Deque.

La auditoría de accesibilidad de Lighthouse genera una lista de todos los problemas de accesibilidad que detecta y una puntuación general de accesibilidad, que clasifica la accesibilidad de la página en una escala de 0 a 100. Lighthouse tiene documentación sobre cómo se calcula el puntaje de accesibilidad, pero esa información no está disponible de en DevTools o en el informe en sí, y sería útil si esa información estuviera más disponible para los usuarios.

Después de la puntuación de accesibilidad hay una lista de los problemas de accesibilidad encontrados por Lighthouse. Cada sección incluye una breve descripción del problema y presenta una lista de elementos que fallan en esa auditoría en particular. Cada sección incluye un enlace a la documentación de Lighthouse que explica con más detalle el problema de accesibilidad y brinda algunas sugerencias sobre cómo resolverlo.

El informe tiene características que facilitan la determinación de qué elementos fallan en la verificación. Al pasar el cursor sobre cualquiera de las etiquetas de elementos enumeradas en un problema, se resalta el elemento en la ventana del navegador para que el usuario pueda identificar visualmente cuál está fallando. Hacer clic en esos elementos lleva al usuario a la línea exacta de código (en el HTML de la pestaña Elementos en DevTools) donde existe el elemento problemático. Esto puede ser útil para solucionar problemas o prepararse para registrar un error de accesibilidad.

En cuanto a la documentación para cada uno de los problemas es inconsistente. Algunos, como "Los colores de fondo y de primer plano no tienen una relación de contraste suficiente", se relacionan directamente con los criterios de éxito específicos de WCAG 2.1 utilizados para evaluar el problema. Otros, como "El documento no tiene un", no hacen referencia a las especificaciones WCAG en absoluto. En su mayor parte, la documentación se basa en educar

al usuario enviándolo a Deque, WebAIM u otros recursos externos en lugar de proporcionar al usuario la información que necesita en una ubicación fácil de encontrar.

Un punto positivo de Lighthouse es que, además de mostrar las auditorías que fallaron y los problemas que las causaron, muestra al usuario qué auditorías pasaron y qué auditorías no se aplicaron a su sitio. Esto puede ayudar al usuario a comprender más acerca de la accesibilidad en su sitio.

Además, posee la sección "Elementos adicionales para verificar manualmente". Las pruebas automatizadas son solo un primer paso para garantizar que un sitio sea accesible. Existen numerosos problemas de accesibilidad que solo un ser humano puede probar con precisión. Esta última funcionalidad también la posee Pa11y, informando al usuario como un "warning" o "notice".

### **3.2.5. Costo**

Pa11y es gratuito, de código abierto y está bien mantenido. No requiere la creación de ninguna cuenta, por lo que no recopila ninguna información del usuario.

La extensión del navegador axe es de uso gratuito tanto para el navegador Chrome como para Firefox. Tanto el plugin linting y la interfaz por comando también lo son. En caso de querer integrar axe en su CI/CD, hay una versión paga llamada axe DevTools que ofrece más funciones.

Lighthouse es de uso gratuito en todas sus formas.

### **3.2.6. Limitaciones y fortalezas**

Las limitaciones de Pa11y radican en el nivel de comodidad del usuario con la línea de comandos y su dedicación para aprender a usar Pa11y de una manera más compleja. Configurarlos para aprovechar todo lo que tiene para ofrecer puede llevar mucho tiempo.

La flexibilidad y escalabilidad de Pa11y son sus mayores fortalezas. Se puede utilizar directamente con un conjunto predeterminado de opciones como un auditor de URL de una página. Pero también puede convertirse en una herramienta mucho más completa que puede

auditar varias páginas bajo demanda mientras realiza pruebas con criterios específicos que usted personaliza.

La documentación proporcionada es una fortaleza. Pally es un conjunto de herramientas de configuración pesada que puede volverse algo complicado si desea crear sus propias especificaciones de auditoría. La documentación anticipa esto y proporciona una gran ayuda. Sin embargo, esto también puede verse como una pequeña debilidad, ya que la cantidad de documentación y la complejidad pueden ser intimidantes hasta que sentirse cómodo con ella. Pero dado que la documentación es bastante útil y completa, es un punto sumamente positivo.

La extensión gratuita del navegador Axe tiene muchas ventajas, ya que es fácil de usar y excelente para realizar una auditoría rápida. Es una herramienta de código abierto que la comunidad y Deque Systems mejoran constantemente. La desventaja de esta herramienta es que, como extensión del navegador, debe analizar manualmente una página cada vez que realiza actualizaciones para verificar que los problemas se solucionen. Deque ofrece una solución paga llamada Axe DevTools para resolver este problema al permitirle realizar pruebas de accesibilidad directamente en su compilación. Con todo, la extensión del navegador Axe es una herramienta de primer nivel para todos los roles en un proyecto para detectar problemas de accesibilidad.

Axe linter se integra a la perfección con un flujo de trabajo de codificación. Con la estrategia de cero falsos positivos del linter, es mucho menos probable que ignore los problemas de accesibilidad. Agregar este tipo de linting al proceso de desarrollo permite encontrar problemas de accesibilidad y que el desarrollador se concentre en menos problemas durante un análisis más completo más adelante en el flujo de trabajo de desarrollo.

Una desventaja es que este linter solo está disponible como complemento de VS Code. Deque está analizando en ampliar su alcance apuntando a algunos IDE más populares.

Si bien Lighthouse en DevTools tiene mucho margen de mejora, tanto en su implementación como en su documentación, la capacidad de ejecutar auditorías en una variedad de métricas en un solo lugar es definitivamente una fortaleza. Si bien implica una configuración más complicada, ejecutar Lighthouse mediante programación resolvería muchos de los desafíos que encuentra un usuario en la versión de DevTools.

### 3.2.7. Conclusiones

La evaluación de accesibilidad es un paso importante para lograr una web accesible. Que una web sea accesible beneficia a todos los usuarios, mejorando el acceso web en general.

Tras comparar las herramientas de análisis de accesibilidad web, se observa, que si bien Pa1ly puede no ser tan conveniente como algunas otras herramientas que se instalan a través de la extensión del navegador, no deja de tener sus propios méritos. Como una opción gratuita, puede ajustarse a cualquier presupuesto y no viene con los problemas de privacidad de acceso a la cuenta de usuario que pueden surgir al considerar las extensiones del navegador. La configuración predeterminada de Pa1ly con auditoría de URL única encuentra una cantidad grande de errores y es sencilla con su informe. Sin embargo, un requisito previo para Pa1ly es el uso de la línea de comandos. Este requisito hace retroceder a esta herramienta en lo que respecta a su facilidad de uso e instalación. Si eso no es una preocupación, Pa1ly tiene muchas configuraciones, funciones y extensiones adicionales disponibles para personalizar. Una de esas extensiones es Pa1ly CI, que le permite configurar Pa1ly para realizar pruebas automatizadas para su sitio. Aunque Pa1ly tiene algunos desafíos con sus requisitos previos y configuración para pruebas más sólidas, su flexibilidad lo convierte en una herramienta muy sólida.

El uso de una herramienta automatizada como parte de su flujo de trabajo de accesibilidad es un primer paso fácil para asegurarse de que está desarrollando aplicaciones web que todos pueden usar. Si bien la extensión gratuita del navegador Axe detecta la mayoría de los problemas, no reemplaza las pruebas manuales para detectar problemas de accesibilidad con elementos como la navegación con el teclado y el texto de enlace no específico. Aún más importante es la prueba de usabilidad para la accesibilidad con personas reales que utilizan sus propias tecnologías de asistencia.

Linters ha sido una herramienta de desarrollo de confianza durante años. Tener un linter funcionando durante el desarrollo puede detener muchos problemas potenciales en su camino. Es como un sistema de alerta temprana para su código. Los linters no pueden atrapar todo, pero reducen la cantidad de problemas que pueden surgir más adelante. Accessibility Linter de Axe es fácil de configurar y usar, se esfuerza por lograr cero falsos positivos y proporciona una gran cantidad de información procesable en el camino.

Lighthouse en DevTools tiene mucho potencial como herramienta de auditoría, pero se siente subdesarrollado en muchos aspectos ya mencionados. Sin embargo, para un usuario que esté interesado en realizar una auditoría de accesibilidad y no necesite entrar en detalles (como las especificaciones WCAG), esta sigue siendo una buena manera de detectar muchos problemas de accesibilidad.

En conclusión, a esta evaluación, se decide integrar Pa11y a la librería WCARG, ya que cuenta con numerosas configuraciones que pueden ser adaptadas y así brindar un informe detallado de la accesibilidad web de las páginas dentro de un marco de un proceso de integración continua.

## 4. Metodología de Desarrollo

El desarrollo del proyecto se realiza completamente dentro del marco de las metodologías ágiles, lo que permite un desarrollo incremental e iterativo. Este enfoque permite que el trabajo sea completamente colaborativo, con el beneficio de poder ofrecer mejoras constantes y reevaluar las decisiones de desarrollo cada vez que sea necesario.

En un marco de trabajo ágil se priorizan los resultados funcionales y la comunicación cara a cara antes de la documentación exhaustiva, lo que ayuda a llevar un mejor entendimiento de la situación actual y las necesidades del proyecto.

### 4.1. Scrum

Scrum es uno de los posibles marcos de trabajo ágil que se pueden utilizar para procesos de desarrollo de software. Es una técnica útil en situaciones de fuerte colaboración, con requerimientos que pueden ser reevaluados y cambiar regularmente.

Se ejecuta en ciclos de tiempo cortos, llamados sprint, que generalmente duran 2 semanas. Al principio de cada sprint se planifican las tareas a cumplir y al finalizar se realizan reuniones de revisión y retrospectiva para identificar oportunidades de mejora a futuro.

El proceso de desarrollo del primer release de WCARG se basó en este framework, tomando algunas técnicas ágiles que ayudaron para mantener un desarrollo iterativo incremental del proyecto. Entre ellas se puede mencionar la priorización de backlog para reconocer las tareas que mayor valor traían en ese momento al proyecto, la utilización de sprint y retrospectivas luego cada sprint para identificar las posibles mejoras del proceso y generar un plan para implementarlas en el siguiente.

### 4.2. Herramientas utilizadas

En esta sección se describen las herramientas utilizadas para asistir el desarrollo del proyecto WCARG.

#### 4.2.1 Visual Studio Code

Visual Studio Code (VS Code) es un editor de código fuente desarrollado por Microsoft. Es software libre y multiplataforma, está disponible para Windows, GNU/Linux y macOS. Es

compatible con una gran variedad de lenguajes de programación, como React, Node.JS y JavaScript, entre otros. Es posible extender sus características mediante la instalación de extensiones ofrecidas dentro de la misma aplicación

### 4.2.2 Jenkins

Jenkins es una herramienta de automatización de código abierto escrita en Java con complementos creados para fines de Continuous Integration (CI) o Continuous Delivery (CD).

Jenkins se utiliza para crear y testear los proyectos de software continuamente, lo que facilita a los desarrolladores la integración de cambios en el proyecto y facilita a los usuarios la obtención de una nueva versión.

Con Jenkins, las organizaciones pueden acelerar el proceso de desarrollo de software a través de la automatización. Integra procesos de ciclo de vida de desarrollo de todo tipo, incluidos compilación, documentación, prueba, empaquetado, etapa, implementación, análisis estático y mucho más.

Jenkins logra la integración continua con la ayuda de complementos. Los complementos permiten la integración de varias etapas de DevOps. Si deseas integrar una herramienta en particular, debe instalar los complementos para esa herramienta como, por ejemplo, Git, proyecto Maven, NodeJs, Amazon EC2, editor HTML, etc.

Trae muchos beneficios, entre los cuales, se destacan:

- Es de código abierto y es fácil de usar, fácil de instalar y no requiere instalaciones o componentes adicionales.
- Fácilmente configurable. Jenkins se puede modificar y extender fácilmente. Implementa código de forma instantánea, genera informes de prueba. Jenkins se puede configurar de acuerdo con los requisitos de integraciones y entrega continuas.
- Plataforma independiente. Jenkins está disponible para todas las plataformas y diferentes sistemas operativos, ya sea OS X, Windows o Linux.
- El amplio conjunto de complementos hace que Jenkins sea flexible y permita construir, implementar y automatizar en varias plataformas.

- Fácil soporte debido a que es de código abierto y ampliamente utilizado, no hay escasez de soporte de grandes comunidades en línea de equipos ágiles.
- El desarrollador escribe las pruebas para detectar los errores de su código lo más rápido posible. De modo que el tiempo del desarrollador se guarda sin desperdiciar integraciones plagadas de errores a gran escala.
- Los problemas se detectan y resuelven casi de inmediato, lo que mantiene el software en un estado en el que se puede liberar en cualquier momento de forma segura.
- La mayor parte del trabajo de integración está automatizado. Por lo tanto, los problemas de integración son menores. Esto ahorra tiempo y dinero durante la vida útil de un proyecto.

### 4.2.3 Trello

Trello es una aplicación basada en el método Kanban y sirve para gestionar tareas permitiendo organizar el trabajo en grupo de forma colaborativa mediante tableros virtuales compuestos de listas de tareas en forma de columnas.

Su función principal es la gestión de proyectos, ya que se pueden representar distintos estados y compartirlas con diferentes personas que formen el mismo. Con ella se intenta mejorar las rutinas de trabajo de un equipo generando prioridades, tiempos, avisos y otras opciones perfectas para organizar un proyecto en el que colaboran varias personas.

A parte de las ya comentada, existen varias características que hacen a esta aplicación muy útil:

- La versión gratuita es muy útil y dispones de las funciones necesarias para un uso amplio de la misma.
- La aplicación online siendo editable y compartida en tiempo real por todos los componentes del proyecto, no precisando de actualizaciones ni guardados.
- La aplicación es muy simple e intuitiva donde no se necesita de formación ni aprendizaje de los usuarios. Además, esto hace que su carga sea muy rápida.

- El sistema de notificaciones te hará saber cuándo se hacen cambios en los tableros que quieras. pudiendo controlar cualquier cambio y desviación del proyecto. Estas notificaciones también pueden recibirse por correo electrónico.
- Función de etiquetado de colores para destacar y organizar visualmente las tareas o marcar algunas destacadas.
- El motor de búsquedas es bueno y rápido. Así poder encontrar la tarea que se precise en el momento que se quiera.
- Se pueden publicar tableros privado o públicos y compartirlo con quien quieras, haciendo que esta aplicación tenga pocos límites.
- Conexión segura y confidencialidad de los datos con copias de seguridad cifradas similares a la que utilizan los bancos.

#### 4.2.4 GitHub

Es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador, y que fue comprada por Microsoft en junio del 2018. La plataforma está creada para que los desarrolladores suban el código de sus aplicaciones y herramientas, y que como usuario no sólo puedas descargar la aplicación, sino también entrar a su perfil para leer sobre ella o colaborar con su desarrollo.

GitHub utiliza el sistema de control de versiones Git y un sistema de gestión de versiones es ese con el que los desarrolladores pueden administrar su proyecto, ordenando el código de cada una de las nuevas versiones que sacan de sus aplicaciones para evitar confusiones. Así, al tener copias de cada una de las versiones de su aplicación, no se perderán los estados anteriores cuando se va a actualizar.

Además, permite comparar el código de un archivo para ver las diferencias entre las versiones, restaurar versiones antiguas si algo sale mal, y fusionar los cambios de distintas versiones. También permite trabajar con distintas ramas de un proyecto, como la de desarrollo para meter nuevas funciones al programa o la de producción para depurar los bugs.

Las principales características de la plataforma es que ofrece las mejores características de este tipo de servicios sin perder la simplicidad, y es una de las más utilizadas del mundo por los desarrolladores. Es multiplataforma, y tiene multitud de interfaces de usuario.

## 4.2.5 CANVAS

Es una herramienta para analizar y crear modelos de negocio de forma simplificada. Se visualiza de manera global en un lienzo dividido en los principales aspectos que involucran al negocio y gira entorno a la propuesta de valor que se ofrece.

El modelo canvas se utiliza para pasar de idea a proyecto y plasmar una idea en un modelo empresarial. Es un modelo “vivo”, es decir, que se va modificando según se va desarrollando, validando clientes, surgen nuevas ideas.

Posee diversos beneficios, en cuales se destacan:

- **Mejora la comprensión:** Utiliza herramientas visuales. Esta metodología fomenta el pensamiento creativo de los trabajadores que crean el lienzo.
- **Amplios puntos de enfoque:** En este modelo se mantiene una constante visión del modelo de negocio desde diferentes perspectivas: comercial, mercado, canales de distribución...
- **Análisis estratégico:** En solo una hoja se pueden visionar todos los elementos del lienzo. Una forma sencilla para sacar el mayor partido a esta herramienta.

## 4.3. Frameworks, Lenguajes y Librerías

En esta sección se describen los frameworks, lenguajes y librerías principales utilizados en el desarrollo del proyecto.

### 4.3.1 NodeJS

Node.JS es un entorno de tiempo de ejecución de JavaScript de capa servidor que permite ejecutar programas escritos en este lenguaje, orientado a eventos asíncronos. De esta forma, permite crear programas del lado del servidor y no del cliente como lo hace Javascript puro.

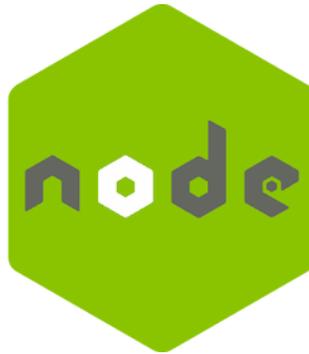


Figura 3.6. Logo NodeJs

### 4.3.2 Pa11y

Pa11y es una librería que permite conocer por medio de cierta configuración y comandos, si una página cumple o no con ciertos estándares de accesibilidad.

Esta fue la librería elegida para ser integrada a WCARG y se explica en la sección anterior su justificación.



Figura 3.7. Logo Pa11y

## 5. Desarrollo

### 5.1. Introducción

Se decide desarrollar una librería que permita mostrar el trabajo de investigación para la resolución de problemas de accesibilidad que tienen las páginas Webs de acuerdo a la regulación argentina.

Para la realización de la librería se utiliza el lenguaje de programación NodeJs, ya que facilita la integración con la antes mencionada librería Pa11y.

El desarrollador podrá integrar nuestra librería en cualquier proceso de integración continua mediante una configuración estándar.

### 5.2. Funcionamiento

El funcionamiento de la librería se basa en la selección de ciertos parámetros de entrada por parte del desarrollador, utilizando el archivo Jenkinsfile. En dicho archivo, el desarrollador agregará la llamada a WCARG de la siguiente manera:

```
stage('WCARG') {
  steps {
    bat './jenkins/scripts/wcarg.bat'
  }
}
```

Figura 3.8. Configuración Jenkins

Una vez agregada la llamada a WCARG al pipeline, el desarrollador crea el archivo wcarg.bat o wcarg.sh según corresponda, que deberá tener la siguiente configuración:

URL=[URL1, URL2]

MAIL=accesibilidad.wcarg@gmail.com

git clone https://github.com/AccesibilidadWCARG/WCARG.git

cd WCARG

call npm install

node wcarg %URL% %STOP\_PIPELINE% %MAIL%

La variable URL contiene las n URLs que del desarrollador desea testear.

La variable MAIL, debe contener el mail al cual se envía el reporte generado por WCARG.

De esta manera, WCARG queda integrada al proceso de integración continua:

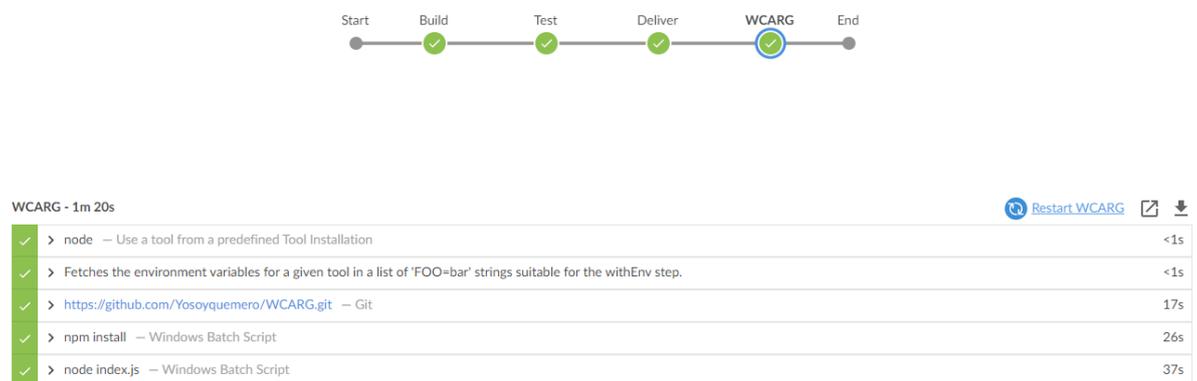


Figura 3.9. Pipeline Jenkins

Una vez terminado el pipeline, WCARG emite el reporte a los desarrolladores. Dicho reporte se emite en formato PDF y contiene el siguiente formato:



Resultados del reporte creado con fecha 11/28/2022 9:41:10 PM

- URL: <https://www.musimundo.com/> | [Accesible](#)

**Errores encontrados**

<b>Error</b>	Principio: 4   Pauta: 4_1   Criterio: 4_1_2   Técnica: H91ANoContent   Nivel: A
<b>Mensaje</b>	Elemento de anclaje encontrado con un atributo href válido, pero no se ha proporcionado contenido de enlace.
<b>Segmento de Código</b>	<a id="cboxClose" href="javascript:void(0);" class="" style="float: left;"></a>
<b>Selector</b>	#cboxClose

<b>Error</b>	Principio: 1   Pauta: 1_1   Criterio: 1_1_1   Técnica: H302   Nivel: A
<b>Mensaje</b>	El elemento img es el único contenido del enlace, pero falta el texto alternativo. El texto alternativo debe describir el propósito del enlace.
<b>Segmento de Código</b>	<a href="/"></a>
<b>Selector</b>	#header > div.nth-child(2) > div > div > div > div.nth-child(1) > div > div > a

<b>Error</b>	Principio: 4   Pauta: 4_1   Criterio: 4_1_2   Técnica: H91InputTextName   Nivel: A
<b>Mensaje</b>	Este elemento de entrada de texto no tiene un nombre disponible para una API de accesibilidad. Los nombres válidos son: elemento de etiqueta, título indefinido, aria-label indefinido, aria-labeledby indefinido.
<b>Segmento de Código</b>	<input class="siteSearchInput left input_SearchBox ui-autocomplete-input" type="text" name="text" value="" maxlength="100" placeholder="Encontrá lo que buscás" data-options="{&quot;autocompleteUrl&quot;: &quot;/search/autocomplete/SearchBox&quot;, &q...
<b>Selector</b>	#header > div.nth-child(2) > div > div > div > div.nth-child(2) > div > div > div > form > div > div > input

1/136

Figura 4. Reporte WCARG

El reporte contiene los errores de accesibilidad encontrados. Informa URL del sitio, si la misma es accesible o no y la siguiente información:

- “Error”, en donde figura el criterio que falló, bajo qué principio, pauta, técnica y nivel.
- “Mensaje”, que informa una descripción del error.
- “Segmento de Código”, que es el segmento de código en donde se ubica el error.
- “Selector”, que es el identificador del elemento dentro de la página Web. Por ejemplo, se puede identificar por ID, XPATH, CSS, entre otros.

### 5.3. Diagrama de Arquitectura

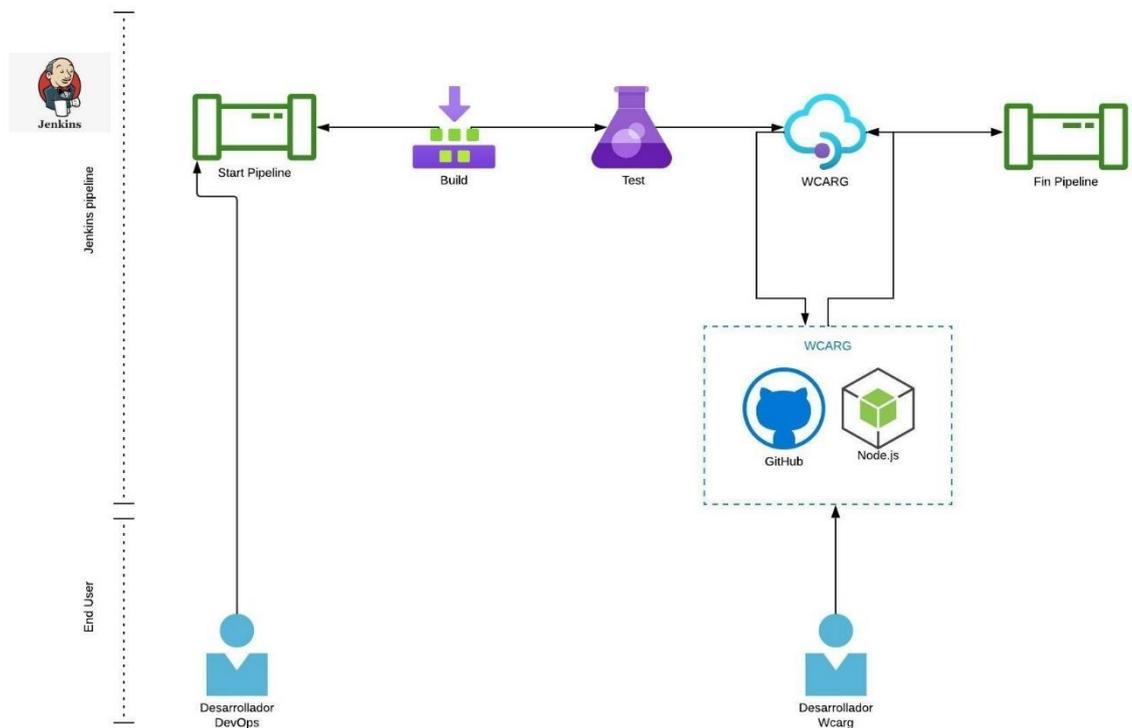


Figura 4.1. Diagrama de Arquitectura

### 5.4. Diagrama de Clases

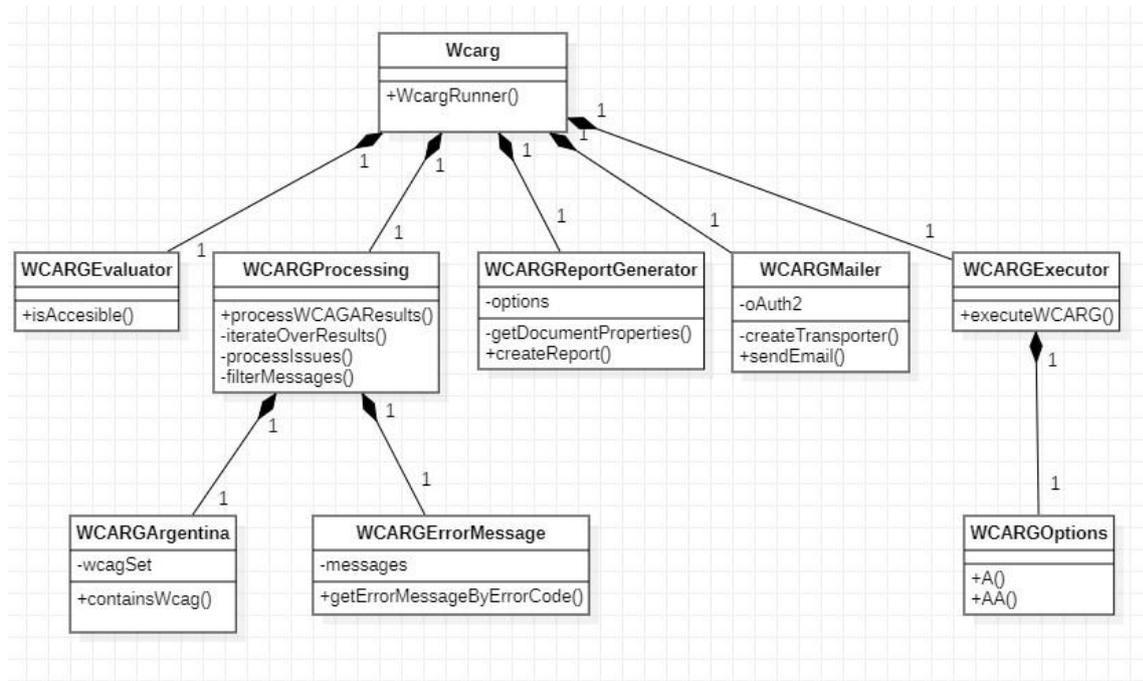


Figura 4.2. Diagrama de Clases

### 5.5. Logo

El logo de WCARG está representado por la combinación de la letra “W” por WCARG, la “A” de accesibilidad y otra “A” de Argentina.

Se utiliza dos colores principales, celeste y blanco, representando a Argentina, y así terminar de representar a la accesibilidad web argentina por medio de los colores.



Figura 4.3. Logo WCARG

## 6. Conclusión

### 6.1. Discusión

El objetivo de la accesibilidad web es que las páginas web sean utilizadas por la mayor cantidad de personas, independientemente de sus conocimientos o capacidades personales y de las características técnicas del equipo utilizado para acceder a la Web.

Cualquier persona que requiere acceder a contenido en la Web puede, bajo distintas circunstancias, depender de que una Web sea accesible para disponer de la información que necesita.

El interés por la accesibilidad web generó una serie de proyectos de investigación en el que se evaluaba la accesibilidad web en distintos segmentos del mercado (universidades, bancos, etc.). Para realizar el análisis de todos los sitios se utilizó como referencia las normas y requisitos de accesibilidad que deben cumplir los sitios web del sector público. Los sitios web del sector público deben respetar las normas y requisitos de accesibilidad recomendados por la ONTI (Oficina Nacional de Tecnología Informática).

La experiencia lograda a lo largo de los distintos proyectos de investigación ponía en evidencia la necesidad de una herramienta que facilite la evaluación de accesibilidad web de acuerdo con la norma vigente en Argentina desde agosto de 2019.

Como respuesta a esta necesidad, se presentó en UADE mediante el INTEC, este proyecto de investigación para desarrollar la librería WCARG, para que sus resultados faciliten la evaluación directa de los que pide la norma argentina.

WCARG busca ser una herramienta que utilicen los desarrolladores como parte de su proceso de integración continua y lograr en un marco de agilidad que las páginas web sean utilizables por el máximo número de personas posibles.

De esta manera, con WCARG se incrementará la cuota de mercado y audiencia de la Web, se mejorará la eficiencia de las páginas Webs y su tiempo de respuesta. Y aún más importante se demuestra responsabilidad social, reforzando positivamente la imagen de las entidades enunciadas en el art 1 de la ley 26.653.

## 6.1. Futuras líneas de investigación

Este trabajo final puede ser continuado en varias líneas de investigación:

- Contemplar en su totalidad los Criterios de Conformidad; es decir, los de Nivel AAA planteados por el W3C en la especificación WCAG versión 2.0.
- Incorporar validaciones de elementos para cumplir con la WCAG versión 3.0, la cual está aún en revisión.

## 7. Anexos

### 7.1. Anexo A: Tabla de criterios de conformidad

A continuación, se describen los criterios de conformidad relacionados con cada una de las pautas. (WCAG 2.0, 2008)

Tabla 1.5. Criterio de Conformidad de la Pauta 1.1. (WCAG, 2008)

<b>Pauta 1.1. Alternativas textuales</b>	
<p><b>1.1.1</b> <b>Contenido no textual.</b> <b>[A]</b></p>	<p>En una página Web se pueden encontrar textos, imágenes, sonidos o vídeos. Para que la página Web sea accesible todos los contenidos no textuales deben tener un equivalente textual que tenga un propósito equivalente. Esto es así siempre, excepto para los siguientes casos:</p> <ul style="list-style-type: none"> <li>• <b>Controles y entrada de datos:</b> si el contenido no textual es un control o acepta datos de entrada del usuario, debe tener un nombre que describa su propósito.</li> <li>• <b>Contenido multimedia dependiente del tiempo:</b> si el contenido no textual es contenido multimedia dependiente del tiempo, entonces el texto alternativo proporciona, al menos, identificación descriptiva del contenido no textual.</li> <li>• <b>Prueba:</b> si el contenido no textual es una prueba o ejercicio que sería inválido si es presentado como texto, entonces el texto alternativo al menos debe proporcionar una identificación descriptiva del contenido no textual.</li> <li>• <b>Sensorial:</b> si el contenido no textual tiene la intención principal de crear una experiencia sensorial específica, entonces el texto alternativo al menos debe proporcionar una identificación descriptiva del contenido no textual.</li> <li>• <b>CAPTCHA:</b> si el propósito del contenido no textual es confirmar que el que está accediendo es una persona y no una computadora, entonces los textos alternativos deben identificar y describir el propósito del contenido no textual, y se deben proporcionar formas alternativas de CAPTCHA usando modelos de salida para diferentes tipos de percepciones sensoriales para acomodarse a diferentes discapacidades.</li> <li>• <b>Decoración, formato, invisible:</b> si un contenido no textual es puramente decorativo, es usado simplemente para dar una apariencia visual, o no es presentado a los usuarios, entonces debe ser implementado de forma que pueda ser ignorado por las tecnologías de apoyo.</li> </ul>

Tabla 1.6. Criterios de Conformidad de la Pauta 1.2. (WCAG, 2008)

<b>Pauta 1.2. Contenido multimedia dependiente del tiempo</b>	
<b>1.2.1 Sólo audio y sólo vídeo (pregrabado) [A]</b>	<p>Para contenidos pregrabados de solo audio y solo vídeo, se proporcionan alternativas textuales equivalentes para que la información pueda ser recibida por cualquier usuario:</p> <ul style="list-style-type: none"> <li>• <b>Solo audio pregrabado:</b> se proporciona una alternativa textual del audio para que las personas que no puedan oírlo puedan leerlo.</li> <li>• <b>Solo vídeo pregrabado:</b> se proporciona un texto alternativo con la descripción del vídeo, para ser leído o una pista de audio con la audiodescripción.</li> </ul>
<b>1.2.2 Subtítulos (pregrabado) [A]</b>	Se ofrecen subtítulos para cualquier audio pregrabado presente en un contenido multimedia.
<b>1.2.3 Audiodescripción o alternativas multimedia (pregrabadas) [A]</b>	Se dispone de una alternativa textual o una audiodescripción para los contenidos multimedia. Las alternativas deben incluir los diálogos, sonidos ambientales, descripción de personajes, etc.
<b>1.2.4 Subtítulos (en directo) [AA]</b>	Se proporcionan subtítulos sincronizados para todo el contenido de audio en directo.
<b>1.2.5 Audiodescripción (pregrabada) [AA]</b>	Se proporciona una audiodescripción para todo el contenido de los vídeos pregrabados.
<b>1.2.6 Lengua de señas (pregrabada) [AAA]</b>	Se ofrece un vídeo con una interpretación en lengua de señas para cualquier contenido de audio pregrabado.
<b>1.2.7 Audiodescripción extendida (pregrabada) [AAA]</b>	Normalmente se utilizan los tiempos de silencio en el vídeo para insertar las descripciones auditivas. Cuando las pausas del audio sean insuficientes para que la audiodescripción transmita correctamente el sentido del vídeo, se proporciona una versión alternativa del vídeo. Se insertan pausas más largas que permitan la correcta descripción del vídeo.
<b>1.2.8 Alternativas multimedia (pregrabada) [AAA]</b>	Se proporciona una alternativa descriptiva para cualquier contenido que incluya vídeo, tenga sonido o no.
<b>1.2.9 Sólo audio (en directo) [AAA]</b>	Se dispone de una transcripción descriptiva que presenta la información equivalente para todo el contenido de audio en directo.

Tabla 1.7. Criterios de Conformidad de la Pauta 1.3. (WCAG, 2008)

<b>Pauta 1.3. Adaptabilidad</b>	
<b>1.3.1 Información y relaciones [A]</b>	La información, la estructura y las relaciones transmitidas a través de la presentación visual se conservan al cambiar el formato de la presentación.
<b>1.3.2 Secuencia significativa [A]</b>	Cuando la secuencia en la que se presenta un contenido afecta a su significado, la secuencia correcta de lectura estará determinada en el código.
<b>1.3.3 Características sensoriales [A]</b>	Las instrucciones que se proporcionan para interactuar con un contenido no dependen de la forma, el tamaño, ubicación visual, orientación o sonido.

Tabla 1.8. Criterios de Conformidad de la Pauta 1.4. (WCAG, 2008)

<b>Pauta 1.4. Distinguible</b>	
<b>1.4.1 Uso del color [A]</b>	El color no se usa como el único método visual para transmitir la información, indicar una acción o distinguir elementos.
<b>1.4.2 Control de audio [A]</b>	Se proporciona un mecanismo para pausar, detener o controlar el volumen de cualquier audio que se reproduzca automáticamente durante más de tres segundos. Se recomienda que sea el usuario el que elija iniciar el audio y no su detención.
<b>1.4.3 Contraste (mínimo) [AA]</b>	La presentación de texto o imágenes de texto tienen una relación de contraste de al menos 4:5:1 entre el texto y el fondo. Excepto para textos grandes donde la relación de contraste mínima es de 3:1. En caso de textos o imágenes de texto decorativos no hay requisitos mínimos, al igual que los textos que pueden aparecer en logotipos.
<b>1.4.4 Variar el tamaño del texto [AA]</b>	Excepto para los subtítulos y las imágenes de texto, se puede variar el tamaño del texto, sin necesidad de utilizar tecnologías de apoyo, hasta el doble de su tamaño sin perder contenido ni funcionalidad.
<b>1.4.5 Imágenes de texto [AA]</b>	Para transmitir la información se usa preferentemente solo texto antes que imágenes de texto. También se pueden utilizar imágenes de texto si pueden ser personalizadas a las necesidades de los usuarios.

<p><b>1.4.6 Contraste (mejorado) [AAA]</b></p>	<p>La presentación de textos o imágenes de texto tiene una relación de contraste de al menos 7:1 entre el texto y el fondo; excepto para textos grandes donde la relación de contraste mínima es de 4:5:1.</p> <p>En caso de textos o imágenes de texto decorativos no hay requisitos mínimos, al igual que los textos que puedan aparecer en logotipos.</p>
<p><b>1.4.7 Sonidos de fondo de audio bajo o inexistente [AAA]</b></p>	<p>No existen sonidos de fondo en el audio o el sonido de fondo se puede apagar. Si existen, los sonidos de fondo permiten entender claramente la locución principal. El sonido de fondo debe ser al menos de 20 dB más bajo que la locución principal, excepto para sonidos ocasionales de menos de dos segundos.</p>
<p><b>1.4.8 Presentación visual [AAA]</b></p>	<p>Se permite que el usuario pueda configurar la presentación visual de bloques de texto según sus necesidades.</p> <p>Para cumplir este criterio de conformidad se deben verificar las siguientes características:</p> <ul style="list-style-type: none"> <li>• El usuario puede seleccionar los colores del texto y del fondo.</li> <li>• El ancho de línea no excede los 80 caracteres.</li> <li>• El texto no está justificado.</li> <li>• El espacio entre líneas es por los menos un espacio y medio en el interior de los párrafos. Y entre párrafos al menos 1.5 veces más amplio que el interlineado.</li> <li>• El texto puede escalar sin el uso de tecnologías de apoyo hasta el doble de su tamaño de manera que no aparezca una barra de desplazamiento horizontal.</li> </ul>
<p><b>1.4.9 Imágenes de texto (sin excepción) [AAA]</b></p>	<p>Las imágenes de texto solo se usan como elementos decorativos o en casos donde la presentación del texto es esencial para la información que se transmite.</p>

Tabla 1.9: Criterios de Conformidad de la Pauta 2.1. (WCAG, 2008)

<p><b>Pauta 2.1 Accesible a través del teclado</b></p>	
<p><b>2.1.1 Teclado [A]</b></p>	<p>Todas las funciones del contenido están disponibles a través de un teclado, excepto las que no sea posible de ninguna manera, como introducir texto a mano alzada.</p>
<p><b>2.1.2 Sin trampa de teclado [A]</b></p>	<p>El foco del teclado no está fijado en un elemento. Si el foco puede moverse a un componente de la página mediante el teclado, entonces el foco tiene que poder salir de ese componente, utilizando</p>

	también el teclado. Si para eso se necesita algo más que pulsar la tecla de [tabulación], se informa al usuario de cómo salir del componente.
<b>2.1.3 Teclado (sin excepción) [AAA]</b>	Se puede utilizar toda la funcionalidad de un contenido a través del teclado sin requisitos de tiempo específico para pulsar las teclas.

Tabla 2. Criterios de Conformidad de la Pauta 2.2. (WCAG, 2008)

<b>Pauta 2.2. Tiempo suficiente</b>	
<b>2.2.1 Límite de tiempo ajustable [A]</b>	Si existe un límite de tiempo para interactuar con el contenido se debe proporcionar la opción de desactivar, ajustar o extender ese límite de tiempo siempre que sea posible. Excepcionalmente no es necesario cumplir este requisito para eventos en tiempo real donde el límite sea estrictamente necesario o cuando el límite de tiempo supera las 20 horas.
<b>2.2.2 Pausar, detener, ocultar [A]</b>	El usuario puede pausar, detener u ocultar cualquier elemento que se mueva, parpadee, se desplace o se actualice automáticamente, si se inicia de forma automática, dura más de cinco segundos y se presenta paralelamente a otro contenido.
<b>2.2.3 Sin tiempo [AAA]</b>	El contenido y la funcionalidad no dependen del tiempo, excepto para el contenido multimedia no interactivo y para los eventos en tiempo real.
<b>2.2.4 Interrupciones [AAA]</b>	El usuario puede proponer o desactivar las interrupciones, como pueden ser las actualizaciones de la página o alertas. Existen excepciones cuando las interrupciones son provocadas por una emergencia, como la existencia de algún riesgo de seguridad o pérdida de datos.
<b>2.2.5 Reautenticación [AAA]</b>	Si una sesión autenticada expira, el usuario puede reautenticar su sesión y continuar la actividad sin perder sus datos.

Tabla 2.1: Criterios de Conformidad de la Pauta 2.3. (WCAG, 2008)

<b>Pauta 2.3. Ataques</b>	
<b>2.3.1 Tres destellos o por debajo del umbral [A]</b>	No existen contenidos que destellen más de tres veces por segundo, o el parpadeo es suficientemente leve, de bajo contraste y no contiene mucho color rojo.
<b>2.3.2 Tres destellos [AAA]</b>	No existen contenidos que destellen más de tres veces por segundo.

Tabla 2.2. Criterios de Conformidad de la Pauta 2.4. (WCAG, 2008)

<b>Pauta 2.4. Navegable</b>	
<b>2.4.1 Saltar bloque [A]</b>	Se proporciona un mecanismo que permite saltar bloques de contenido que se repiten en varias páginas Web, como los elementos de navegación.
<b>2.4.2 Página titulada [A]</b>	Las páginas Web tienen títulos descriptivos que identifican su tema o propósito.
<b>2.4.3 Orden de foco [A]</b>	Si una página Web puede navegarse secuencialmente y el orden de navegación afecta su significado u operatividad, los componentes que pueden recibir el foco lo hacen de forma coherente y ordenada para que la información conserve su significado.
<b>2.4.4 Propósito de un vínculo (en su contexto) [A]</b>	El propósito de cada enlace puede determinarse únicamente con el texto del enlace sin su contexto. Los enlaces con el mismo destino deben tener la misma descripción, y los enlaces con destinos y propósitos diferentes deben tener descripciones diferentes.
<b>2.4.5 Múltiples medios [AA]</b>	Existe más de una forma para localizar una página Web dentro de un sitio Web para que cada usuario elija la que mejor se adapte a sus necesidades. Esto se debe cumplir excepto cuando la página Web es el resultado, o uno de los pasos, de un proceso.
<b>2.4.6 Encabezados y etiquetas [AA]</b>	Los encabezados y las etiquetas son descriptivas y ayudan a encontrar la información e identificar los componentes de la página Web y sus funciones.
<b>2.4.7 Foco visible [AA]</b>	Si la interfaz de usuario se maneja a través del teclado se proporciona un indicador visible que permite saber en qué elemento está el foco del teclado en todo momento.

<p><b>2.4.8 Ubicación [AAA]</b></p>	<p>Se proporciona al usuario información de su ubicación dentro de una estructura de páginas Web.</p>
<p><b>2.4.9 Propósito de un vínculo (vínculo solo) [AAA]</b></p>	<p>Se puede identificar el propósito de cada enlace únicamente con el texto del propio enlace, excepto cuando el propósito del enlace puede resultar ambiguo para los usuarios en general.</p>
<p><b>2.4.10 Encabezados de sección [AAA]</b></p>	<p>Se emplean encabezados de sección para organizar el contenido. Ayudan a organizar el contenido y facilitar su comprensión.</p>

Tabla 2.3. Criterios de Conformidad de la Pauta 3.1. (WCAG, 2008)

<p><b>Pauta 3.1. Legible</b></p>	
<p><b>3.1.1 Idioma de la página [A]</b></p>	<p>El idioma por defecto de cada página Web está identificado en el código, para que las tecnologías de apoyo muestren la información con más precisión.</p>
<p><b>3.1.2 Idioma de partes [AA]</b></p>	<p>Si existen partes de la página en otro idioma distinto del principal se debe especificar para que los agentes de usuario y tecnologías de apoyo presenten la información correctamente. Existen excepciones para nombres propios, palabras técnicas o de un idioma indeterminado.</p>
<p><b>3.1.3 Palabras inusuales [AAA]</b></p>	<p>Se proporcionan mecanismos para identificar definiciones muy específicas, palabras empleadas de forma inusual o ambigua, incluyendo modismos y jerga.</p>
<p><b>3.1.4 Abreviaturas [AAA]</b></p>	<p>Se proporciona una manera de identificar las formas expandidas o el significado de las abreviaturas utilizadas.</p>
<p><b>3.1.5 Nivel de lectura [AAA]</b></p>	<p>El texto está escrito de forma sencilla y comprensible. Si se requiere una habilidad de lectura más avanzada, se proporciona contenido complementario que facilite su comprensión, o una versión que sea más fácil de entender.</p>
<p><b>3.1.6 Pronunciación [AAA]</b></p>	<p>Se proporciona una forma de identificar la pronunciación específica de palabras donde el significado pueda ser ambiguo, en su contexto, al escribirse igual, pero tener distinto significado, dependiendo de su pronunciación.</p>

Tabla 2.4. Criterios de Conformidad de la Pauta 3.2. (WCAG, 2008)

<b>Pauta 3.2. Predecible</b>	
<b>3.2.1 Con foco [A]</b>	Cuando cualquier componente recibe el foco no se provoca ningún cambio de contexto en la página.
<b>3.2.2 Con entrada de datos [A]</b>	Si se cambia la configuración de un componente de la interfaz de usuario no se produce ningún cambio de contexto a menos que el usuario haya sido advertido del comportamiento antes de utilizar el componente.
<b>3.2.3 Navegación consistente [AA]</b>	Los elementos de navegación comunes, que se repiten en varias páginas dentro del sitio Web, aparecen en la misma posición y en el mismo orden al navegar por el sitio, a menos que el usuario decida cambiarlos.
<b>3.2.4 Identificación consistente [AA]</b>	Los elementos que tienen la misma funcionalidad dentro de un sitio Web se identifican de forma coherente.  Si el elemento tiene exactamente la misma funcionalidad en todas las páginas, deberá etiquetarse de la misma forma en todas.
<b>3.2.5 Cambio a petición [AAA]</b>	Los cambios de contexto se inician únicamente a petición del usuario, o existe un mecanismo para detener los cambios.

Tabla 2.5. Criterios de Conformidad de la Pauta 3.3. (WCAG, 2008)

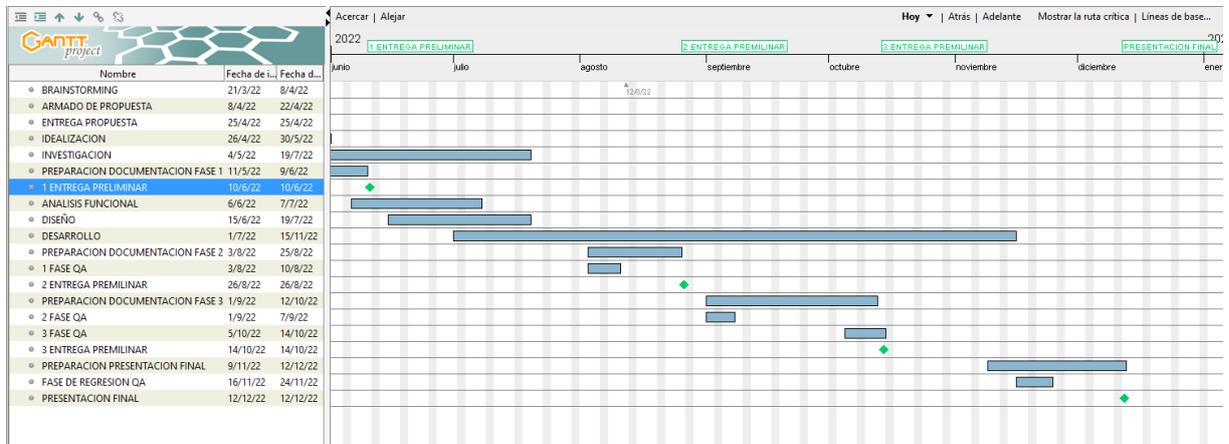
<b>Pauta 3.3. Ayuda a la entrada de datos</b>	
<b>3.3.1 Identificación de errores [A]</b>	Si se detecta automáticamente un error en los datos de entrada, se identifica y se le describe al usuario el error con un mensaje de texto muy específico.
<b>3.3.2 Instrucciones o etiquetas [A]</b>	Se proporcionan instrucciones o etiquetas claras y sencillas cuando se requiere que el usuario introduzca datos.
<b>3.3.3 Sugerencias ante errores [AA]</b>	Si se detecta automáticamente un error de entrada de datos y se pueden determinar las sugerencias apropiadas para la corrección, entonces se proporcionan las sugerencias al usuario, a menos que esto pudiese poner en riesgo la seguridad o el propósito del contenido.
<b>3.3.4 Prevención de errores (legales,</b>	Las páginas Web que permiten a los usuarios enviar, modificar o borrar datos que supongan para el usuario compromisos

<b>financieros, de datos) [AA]</b>	legales o transacciones económicas, deben permitir que estas acciones sean reversibles, comprobadas o confirmadas.
<b>3.3.5 Ayuda [AAA]</b>	Se proporciona ayuda contextual, para que el usuario disponga de una guía de cómo realizar una tarea.
<b>3.3.6 Prevención de errores (todos) [AAA]</b>	Las páginas Web que requieran que el usuario envíe información deben permitir que el envío sea reversible, comprobado o confirmado.

Tabla 2.6. Criterios de Conformidad de la Pauta 4.1. (WCAG, 2008)

<b>Pauta 4.1. Compatible</b>	
<b>4.1.1 Interpretación [A]</b>	El contenido se ha implementado utilizando un lenguaje de marcas bien formadas, respetando sus especificaciones y sin errores sintácticos. Los elementos cuentan con etiquetas de apertura y cierre, están anidados correctamente, no contienen atributos duplicados y cualquier identificador es único.
<b>4.1.2 Nombre, rol, valor [A]</b>	Para todos los componentes de la interfaz de usuario como elementos de formulario, enlaces y componentes generados por medio de scripts, el nombre y el rol pueden ser determinados por software. Los estados, propiedades y valores que pueden ser establecidos por el usuario pueden ser establecidos por software. Y los cambios en estos elementos se deben notificar a los agentes de usuario, incluidas las tecnologías de apoyo.

## 7.1. Anexo B: Cronograma



Se llega al 100% de desarrollo y de alcance de la PFI. Durante la etapa 2 se realizó la presentación de dos papers, las cuales fueron a los congresos CACIC y CoNaIISI. Ambos papers durante la fase 3 fueron aprobados.

Cabe mencionar que el paper a CACIC fue defendido exitosamente y ya publicado. Con respecto a CoNaIISI, el paper va a ser defendido durante el transcurso de noviembre.

## 9. Glosario

**Agente de usuario (Navegador).** Aplicación informática que funciona como cliente en un protocolo de red. El nombre aplica generalmente para referirse a aquellas aplicaciones que acceden a la Web (navegadores, arañas de buscadores, teléfonos móviles, lectores de pantalla y otros).

**ATAG.** Authoring Tool Accessibility Guidelines. Pautas de accesibilidad del consorcio W3 para herramientas de diseño.

**Barrera (a la accesibilidad).** Elemento en una página que impide a las personas con discapacidad acceder al recurso.

**Captcha.** Completely Automated Public Turing test to tell Computers and Humans Apart. Prueba desafío-respuesta utilizada en computación para determinar cuándo el usuario es o no humano. Consiste en que el usuario introduzca un conjunto de caracteres que se muestran en una imagen distorsionada que aparece en pantalla. Se supone que una máquina no es capaz de comprender e introducir la secuencia de forma correcta por lo que solamente el humano podría hacerlo.

**Clic/Click.** En informática, se denomina clic, hacer clic, "clicar", "clicquear" o pinchar a la acción de pulsar cualquiera de los botones de un ratón de computadora. Como resultado de esta operación, el sistema aplica alguna función o proceso al objeto señalado por el cursor o el puntero en el momento de realizarla.

**CSS.** Cascading Style Sheets. Hojas de estilo en cascada. Es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

**DTD.** Document Type Definition. Definición de Tipo de Documento. Es una descripción de estructura y sintaxis de un documento XML o SGML. Su función básica es la descripción de la estructura de datos, para usar una estructura común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma, los documentos pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae

consigo cada documento, y pueden además compartir la misma descripción y forma de validación dentro de un grupo de trabajo que usa el mismo tipo de información.

**DOM.** Document Object Model. Modelo de Objetos del Documento. Es esencialmente una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse los objetos, y una interfaz estándar para acceder a ellos y manipularlos. A través del DOM, los programas pueden acceder y modificar el contenido, estructura y estilo de los documentos HTML y XML, que es para lo que se diseñó principalmente.

**Examinator.** Analizador web que califica el nivel de accesibilidad basado en la normativa WCAG 2.0 y en pruebas acorde a las discapacidades que puede tener una persona.

**Flash. Adobe Flash Player.** Aplicación en forma de reproductor multimedia creado inicialmente por Macromedia y actualmente distribuido por Adobe Systems. Permite reproducir archivos SWF que pueden ser creados con la herramienta de autoría Adobe Flash, con Adobe Flex o con otras herramientas de Adobe y de terceros. Estos archivos se reproducen en un entorno determinado. En un sistema operativo tiene el formato de aplicación del sistema, mientras que, si el entorno es un navegador, su formato es el de un Plug-In u objeto ActiveX.

**Guideline (Pauta, referido a la accesibilidad).** Conjunto de recomendaciones detalladas y organizadas con criterios de conformidad. Cada pauta contiene sugerencias y ejemplos a seguir para lograr la conformidad con cada uno de sus criterios.

**HTML.** HyperText Markup Language. Lenguaje de marcado de hipertexto. Es el lenguaje de marcado predominante para la elaboración de páginas Web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de etiquetas, rodeadas por corchetes angulares (<,>).

**HTTP.** HyperText Transfer Protocol. Protocolo de Transferencia de Hipertexto. Es el protocolo usado en cada transacción de la WWW. Fue desarrollado por el W3C y la IETF (Internet Engineering Task Force), colaboración que culminó en 1999 con la publicación de una serie de RFC, el más importante de ellos es el RFC 2616 que especifica la versión 1.1. Es un

protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor.

**JavaScript.** Es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipificado y dinámico. Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador Web permitiendo mejoras en la interfaz de usuario y páginas Web dinámicas, en bases de datos locales al navegador.

**JSON.** Formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript, aunque, debido a su amplia adopción como alternativa a XML, se considera un formato independiente del lenguaje.

**PDF.** Portable Document Format. Formato de documento portátil. Es un formato de almacenamiento de documentos, desarrollado por la empresa Adobe Systems. Este formato es de tipo compuesto (imagen vectorial, mapa de bits y texto).

**Script.** En informática un guion, archivo de órdenes o archivo de procesamiento por lotes, vulgarmente referidos con el barbarismo script, es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano. Los guiones son casi siempre interpretados, pero no todo programa interpretado es considerado un guion. El uso habitual de los guiones es realizar diversas tareas como combinar componentes, interactuar con el sistema operativo o con el usuario. Por este uso es frecuente que los shells sean a la vez intérpretes de este tipo de programas.

**Sistema Operativo.** Es un programa o conjunto de programas que en un sistema informático gestiona los recursos de hardware y provee servicios a los programas de aplicación, y corre en modo privilegiado respecto de los restantes.

**Software.** Se conoce como software al equipamiento o soporte lógicos de un sistema informático; comprende el conjunto de los componentes lógicos necesarios que hacen posible la realización de tareas específicas, en contraposición a los componentes físicos, que son llamados hardware.

**TAW.** Test Accessibility Web. Validador de accesibilidad web basado en la normativa WCAG.

**TIC.** Tecnologías de Información y Comunicación. Agrupan los elementos y las técnicas usadas en el tratamiento y la transmisión de las informaciones, principalmente de informática, Internet y telecomunicaciones.

**UAAG.** User Agent Accessibility Guidelines. Pautas del consorcio W3 que explican cómo hacer agentes de usuario accesibles para personas con discapacidad, y cómo incrementar la accesibilidad al contenido Web.

**UML.** Unified Modeling Language. Lenguaje Unificado de Modelado. Es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad; está respaldado por el OMG (Object Management Group). Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema.

**URL.** Uniform Resource Locator. Localizador de recursos uniforme. Es una secuencia de caracteres, de acuerdo a un formato modélico y estándar, que se usa para nombrar recursos en Internet para su localización o identificación, como por ejemplo documentos textuales, imágenes, videos, presentaciones digitales, etc.

**W3C.** World Wide Web Consortium. Consorcio Web Mundial. Abreviado W3C, es un consorcio internacional que produce recomendaciones para la World Wide Web. Está dirigida por Tim Berners-Lee, el creador original de URL, HTTP y HTML que son las principales tecnologías sobre las que se basa la Web.

**WAI.** Web Accessibility Initiative. Grupo de interés del Consorcio W3 sobre asuntos de accesibilidad en la Web.

**WCAG.** Web Content Accesibility Guidelines. Son un conjunto de pautas de accesibilidad Web publicados por el grupo WAI (Web Accesibility Initiative) perteneciente a el W3C. Estos consisten en un conjunto de pautas para desarrollar contenido accesible en las páginas Web, principalmente para los usuarios con discapacidad, pero también para los agentes de usuario o componentes de navegación, incluyendo dispositivos altamente limitados, como los teléfonos móviles. La versión actual es 2.0.

**WWW. World Wide Web.** Red Informática Mundial. Es un sistema de distribución de información basado en hipertexto o híper-medios enlazados y accesibles a través de Internet. Con un navegador Web, un usuario visualiza sitios Web compuestos de páginas Web que pueden contener texto, imágenes, videos u otros contenidos multimedia, y navega a través de ellas usando hiperenlaces.

**XHTML.** Extensible HyperText Markup Language. Es básicamente HTML expresado como XML válido. Es más estricto a nivel técnico, pero esto permite que posteriormente sea más fácil al hacer cambios o buscar errores entre otros.

**XML.** Extensible Markup Language. Lenguaje de Marcas Extensible. Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C).

## 10. Bibliografía

- ACHECKER, 2022. Inclusive Design Research Centre. *Inclusive Design Research Centre* [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://idrc.ocadu.ca/>.
- ALONSO, A.J. y CAMACHO, J.G., 2010. La baja interacción del espectador de vídeos en Internet: caso Youtube España. *Revista Latina de Comunicación Social*, no. 65, pp. 421-435. ISSN , 1138-5820.
- AMBLER, S. y SADALAGE, P.J., 2006. *Refactoring Databases*. S.l.: s.n.
- AQUINO ZÚÑIGA, S.P., GARCÍA MARTÍNEZ, V. y IZQUIERDO, J., 2012. La inclusión educativa de ciegos y baja visión en el nivel superior: Un estudio de caso. *Sinéctica*, no. 39, pp. 01-21. ISSN 1665-109X.
- CADENAS, S., 2010. La tipografía y su complejidad creativa. , no. 3, pp. 5.
- CARRERAS MONTOTO, O.C., 2007. Usable y accesible. Olga Carreras. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://olgacarreras.blogspot.com/>.
- CONSTRUCT CHECKER, 2022. WCAG Contrast checker – Extensión para Firefox (es). [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://addons.mozilla.org/es/firefox/addon/wcag-contrast-checker/>.
- CORIA, S.A., 2010. Accesibilidad e inclusión social en las currículas de diseño. *Actas de Diseño N°8*, vol. Año IV, Vol. 8, Marzo 2010, Buenos Aires, Argentina., pp. 94-95.
- CSS VALIDATOR, 2022. W3C CSS Validation Service. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <http://www.css-validator.org/>.
- DICCIONARIO DE LA LENGUA ESPAÑOLA, 2022. Diccionario de la lengua española | Edición del Tricentenario. «Diccionario de la lengua española» - Edición del Tricentenario [en línea]. [Consulta: 5 junio 2022]. Disponible en: <https://dle.rae.es/>.
- DOKER, 2022. Docker site. [en línea]. [Consulta: 10 junio 2022]. Disponible en: <https://www.docker.com/>.
- FARCIC, V., 2016. *The DevOps 2.0 Toolkit: Automating the Continuous Deployment Pipeline with Containerized Micro Services*. S.l.: s.n.
- FERNANDEZ, J.P., 2015. *Aplicación de técnicas de usabilidad y accesibilidad en el entorno cliente. IFCD0210*. S.l.: IC Editorial. ISBN 978-84-16433-05-6.
- FIREBUG, 2022. Firebug. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://getfirebug.com/>.

- GARROTE ROJAS, D. y PALOMARES RUIZ, A., 2014. *Una mirada a la realidad: propuestas innovadoras para favorecer la inclusión educativa*. S.l.: Universidad de Castilla La Mancha. ISBN 978-84-617-0349-4.
- GONZÁLEZ ORELLANA, C.A., 2012. Guía demostrativa de técnicas de impresión análogas no industrializadas, aplicadas en un sistema de papelería de branding. En: Accepted: 2013-04-09T21:30:18Z [en línea], [Consulta: 5 junio 2022]. Disponible en: <http://dspace.ucuenca.edu.ec/handle/123456789/355>.
- GRUVER, G. y MOUSER, T., 2015. *Leading The Transformation*. Kindle Edition, 2015. S.l.: s.n.
- HERA, 2022. HERA FFX. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <http://www.sidar.org/recur/aplica/heraffx.php>.
- HONORABLE CONGRESO DE LA NACIÓN ARGENTINA, 2010. LEY DE ACCESIBILIDAD DE LA INFORMACION EN LAS PAGINAS WEB. [en línea]. [Consulta: 5 junio 2022]. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/175000-179999/175694/norma.htm>.
- INDEC, 2018. *el Estudio Nacional sobre el Perfil de las Personas con Discapacidad*. 2018. S.l.: s.n.
- KIM, G. y BEHR, K., 2013. The Phoenix Project. [en línea]. [Consulta: 6 junio 2022]. Disponible en: [https://www.goodreads.com/work/best\\_book/23848838-the-phoenix-project-a-novel-about-it-devops-and-helping-your-business](https://www.goodreads.com/work/best_book/23848838-the-phoenix-project-a-novel-about-it-devops-and-helping-your-business).
- KIM, G. y HUMBLE, J., 2016. *The DevOps Handbook* [en línea]. S.l.: s.n. [Consulta: 6 junio 2022]. Disponible en: [https://books.google.com/books/about/The\\_DevOps\\_Handbook.html?hl=es&id=8kRDEAAQBAJ](https://books.google.com/books/about/The_DevOps_Handbook.html?hl=es&id=8kRDEAAQBAJ).
- LEVEL ACCESS, 2022. Level Access: Web Accessibility Compliance Solutions | ADA, 508, WCAG, VPAT. *Level Access* [en línea]. [Consulta: 15 octubre 2022]. Disponible en: <https://www.levelaccess.com/>.
- LUJAN MORA, S., 2006. Accesibilidad en la Web. [en línea]. [Consulta: 5 junio 2022]. Disponible en: <https://accesibilidadenlaweb.blogspot.com/>.
- LUJAN MORA, S., 2022. Accesibilidad Web: Beneficios. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <http://accesibilidadweb.dlsi.ua.es/?menu=beneficios>.
- MARCOS ALBA, C.L. y OLIVAREZ RUIZ, J., 2014. *Percepción y Pensamiento Gráfico. Estrategias gestálticas en la docencia de E.G.A. y el Diseño Gráfico* [en línea]. S.l.: Universidad de Alicante. Vicerrectorado de Estudios, Formación y Calidad. [Consulta: 5 junio 2022]. ISBN 978-84-697-0709-8. Disponible en: <http://rua.ua.es/dspace/handle/10045/41945>.

- OAW ECUADOR, 2022. Analizador Online - Observatorio de Accesibilidad Web Ecuador. [en línea]. [Consulta: 15 octubre 2022]. Disponible en: <http://45.235.140.51/owa/analizador.jsf>.
- OBSERVATORIO DE LA ACCESIBILIDAD, 2022. Accesibilidad Universal. [en línea]. [Consulta: 5 junio 2022]. Disponible en: <https://observatoriodelaaccesibilidad.es/accesibilidad-2>.
- OMS, 2021. Discapacidad y salud. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://www.who.int/es/news-room/fact-sheets/detail/disability-and-health>.
- PODER EJECUTIVO NACIONAL, 2003. ACCESO A LA INFORMACION PUBLICA. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/90000-94999/90763/norma.htm>.
- PODER EJECUTIVO NACIONAL, 2005. PLAN NACIONAL DE GOBIERNO ELECTRONICO Y PLANES SECTORIALES DE GOBIERNO ELECTRONICO Decreto 378/2005 Apruébanse los Lineamientos Estratégicos para la puesta en marcha de los mencionados Planes. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/105000-109999/105829/norma.htm>.
- PODER EJECUTIVO NACIONAL, 2019. Disposición 6/2019 - Ministerio de Justicia y Derechos Humanos - Argentina. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <http://servicios.infoleg.gob.ar/infolegInternet/anexos/325000-329999/328722/norma.htm>.
- REVILLA MUÑOZ, O., 2013. Ebook WCAG 2.0 DE FORMA SENCILLA EBOOK de OLGA REVILLA MUÑOZ | Casa del Libro. *casadellibro* [en línea]. [Consulta: 5 junio 2022]. Disponible en: <https://www.casadellibro.com/ebook-wcag-20-de-forma-sencilla-ebook/9788461464111/2119087>.
- RODRÍGUEZ, J.M., 2011. La (con)ciencia del Trabajo Social en la discapacidad: hacia un modelo de intervención social basado en derechos. *Documentos de trabajo social: Revista de trabajo y acción social*, no. 49, pp. 9-33. ISSN 1133-6552.
- SHAWN LAWTON, 2014. Easy Checks – A First Review of Web Accessibility. *Web Accessibility Initiative (WAI)* [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://www.w3.org/WAI/test-evaluate/preliminary/>.
- SOTIL, W. y CALVO, N., 2015. Tests de visión cromática asistidos por computadora. *VI Congreso Argentino de Informática y Salud (CAIS) - JAIIO 44 (Rosario, 2015)* [en línea]. S.l.: s.n., [Consulta: 5 junio 2022]. Disponible en: <http://sedici.unlp.edu.ar/handle/10915/54837>.
- STOCKMAN, A. y BRAINARD, D.H., 2010. COLOR VISION MECHANISMS. , pp. 104.

- SUBIELA HERNANDEZ, B.J., 2010. El diseño de revistas como lenguaje: fundamentos teóricos. En: Accepted: 2011-10-03T08:38:21Z [en línea], [Consulta: 5 junio 2022]. Disponible en: <http://repositorio.ucam.edu/handle/10952/80>.
- TAW, 2022. TAW - Servicios de accesibilidad y movilidad web. [en línea]. [Consulta: 14 octubre 2022]. Disponible en: <http://tawdis.fundacionctic.org/index.html?lang=es>.
- TOTAL VALIDATOR, T., 2022. Web Accessibility Testing | Total Validator. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://www.totalvalidator.com/>.
- TUCH, A.N., PRESSLABER, E., STOECKLIN, M., OPWIS, K. y BARGAS-AVILA, J., 2012. The role of visual complexity and prototypicality regarding first impression of websites: Working towards understanding aesthetic judgments. *International Journal of Human-Computer Studies*, vol. 70(11), pp. 794-811.
- TUTUY, A.J.E., MARTÍNEZ, S.E., VILA, V.G. y BARRIOS, C.E., 2012. LA ATENCIÓN ODONTOLÓGICA DEL PACIENTE CON DEFICIENCIA AUDITIVA. *Acta Odontológica Colombiana*, vol. 2, no. 2, pp. 211-216. ISSN 2027-7822.
- VEGA PÉREZ, M.A., 2016. Evolución de diseños basados en la naturaleza diseño de un juguete didáctico para entender a las arañas. En: Accepted: 2016-08-16T20:26:59Z [en línea], [Consulta: 5 junio 2022]. Disponible en: <http://repositorio.usfq.edu.ec/handle/23000/5152>.
- VELASCO SANTOS, P., SÁNCHEZ GUERRERO, L., LAUREANO CRUCES, A.L. y MORA TORRES, M., 2011. *Un diseño de interfaz: tomando en cuenta los estilos de aprendizaje* [en línea]. S.l.: s.n. [Consulta: 5 junio 2022]. ISBN 978-607-7854-36-4. Disponible en: <https://repository.udca.edu.co/handle/11158/400>.
- VELÁSQUEZ HENAO, J.D., 2012. Principios básicos de diseño gráfico aplicados a la preparación de ayudas visuales para presentaciones científicas y de negocios. *Estudios Gerenciales*, vol. 28, no. 123, pp. 167-189. ISSN 0123-5923,.
- VILLOSLADA SÁNCHEZ, A., 2011. *La Enseñanza de español como lengua extranjera a estudiantes con deficiencia visual o cieguera: Propuesta didáctica de adaptación de actividades para el DELE A1* [en línea]. 2011. S.l.: s.n. Disponible en: <https://www.educacionyfp.gob.es/dam/jcr:1a751afe-1e4c-4bc9-a81f-d88fb3208fda/2013-bv-14-03ana-villoslada-pdf.pdf>.
- WAI, 1998. Initiative (WAI) -W3C Web Accessibility. *Web Accessibility Initiative (WAI)* [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://www.w3.org/WAI/>.
- WAVE, 2022. WAVE Web Accessibility Evaluation Tool. [en línea]. [Consulta: 14 octubre 2022]. Disponible en: <https://wave.webaim.org/>.
- WC3 MARKUP VALIDATOR, 2022. The W3C Markup Validation Service. [en línea]. [Consulta: 6 junio 2022]. Disponible en: <https://validator.w3.org/>.

WCAG 2.0, 2008. Web Content Accessibility Guidelines (WCAG) 2.0. [en línea]. [Consulta: 5 junio 2022]. Disponible en: <https://www.w3.org/TR/WCAG20/>.