



SiMor

Analizador intensivo de Accesibilidad Web basado en reglas

Agradecimientos

Magalí

Principalmente quiero agradecer a mi hijo Derek por entender mis ausencias, por su paciencia y por ser mi principal motivación durante todos estos años.

A Francisco, por su compañía y por ser mi sostén durante este tiempo.

A mi familia, por apoyarme y acompañarme.

A mis amigos y compañeros, especialmente a Emilia, por hacer más fácil este camino. A Diego por darme su confianza y apoyo para emprender esta carrera.

Finalmente a Gaspar por su amistad, confianza y dedicación con la que trabajó durante el desarrollo de esta tesina.

Gaspar

A mis padres, por el inmenso apoyo que me dieron durante toda la carrera, y por haberme dado la posibilidad de llevarla a cabo.

A Einar, por haber sido el puntapié para la realización de esta tesina.

A mis amigos, especialmente a Juampi, por haber sido una necesaria y leal compañía durante toda mi carrera.

A Magalí, por la gran amistad y por el entusiasmo puesto en este trabajo.

A la Facultad, por toda la excelente formación que tuve y sigo teniendo, y por todos los amigos que me dio.

Agradecemos a nuestras directoras, Claudia e Ivana, por habernos guiado, acompañado y aconsejado durante todo el desarrollo de esta tesina.

“El poder de la Web reside en su universalidad. El acceso para todo el mundo, a pesar de la discapacidad, es un aspecto esencial.”

- Tim Berners-Lee

Índice

Introducción	6
Objetivos	8
Capítulo 1: El acceso a la información	9
1.1. Problemática en el acceso a la información en la Web	10
1.1.1. Barreras de Accesibilidad	10
Barreras de hardware	10
Barreras de software	10
Las discapacidades y el acceso a la Web	11
Visual	11
Auditiva	12
Motora	12
Cognitiva y neuronal	13
Discapacidades relacionadas con la edad	14
Otras barreras	14
1.2. Conclusión	14
Capítulo 2: Accesibilidad Web	15
2.1. Introducción	15
2.2. Beneficios que ofrece la accesibilidad web	16
2.3. Construcción de sitios web accesibles	20
2.4. Estándares vigentes	21
2.4.1. World Wide Web Consortium (W3C)	21
2.4.2. Pautas WCAG 1.0	23
Estructura de las pautas WCAG 1.0	24
Principios fundamentales	24
2.4.3. Pautas WCAG 2.0	27
Criterios de conformidad	28
Principios fundamentales	28
2.4.4. Pautas UAAG	32
2.4.5. Pautas ATAG	32
2.4.6. Estándares de la Sección 508	33
2.4.7. WAI-ARIA	33
2.4.8. Estándar ISO/IEC 40500:2012	34
2.4.9. Metodología Unificada de Evaluación Web (UWEM):2006	34
2.5. Marco legal y normativas vigentes	35
2.5.1. Ley Nacional de Accesibilidad 26.653	35
2.5.2. Normas internacionales	36
AENOR	36
Norma UNE 139803:2004	36
Norma UNE 139803:2012	36

Ley Española 56/2007	37
Sección 508 - Estados Unidos de América	37
Real Decreto 1494/2007	38
Norma Técnica Colombiana 5854	38
Norma CWA 15554:2006	39
2.6 Conclusión	39
Capítulo 3: Validadores de accesibilidad	40
3.1. Introducción	40
3.2. Clasificación de validadores	40
3.3. Características relevadas a profesionales del área	43
3.4. Evaluación y revisión	45
3.4.1. Características de los validadores evaluados	46
AChecker - Web Accessibility Checker	46
FAE - Functional Accessibility Evaluator 1.1	48
WAVE - Web Accessibility Evaluation Tool	49
TAW - Test de Accesibilidad Web	52
3.4.2. Comparativa de características	55
3.4.3. Comparación de los resultados de validación emitidos	55
3.4.4. Correcciones que requieren intervención humana	57
3.5. Conclusión	58
Capítulo 4: SiMor	59
4.1. ¿Qué es SiMor?	59
4.2. Definición del proceso de desarrollo	61
4.3. Decisiones de diseño	61
4.4. Implementación de SiMor	62
4.4.1. Java como plataforma	62
4.4.2. Tecnologías utilizadas	63
4.4.3. El modelo de reglas	68
4.5. Frontend HTML responsive y accesible	71
4.6. Licencia y disponibilidad	76
4.7. Comparación de SiMor	76
4.8. Conclusión	77
Capítulo 5: Uso de SiMor	79
5.1. Validación de código HTML	80
5.2. Validación de sitio web	85
5.3. Auto evaluación de SiMor	88
5.4. Evaluación de SiMor con WAVE	91
5.5. Evaluación de la accesibilidad de los demás validadores con SiMor	92
5.6. Conclusión	96
Capítulo 6: Conclusiones	97
Anexo	99
Referencias	102

Introducción

En los últimos años ha habido una importante concientización social respecto de la integración de personas con capacidades funcionales disminuidas y el acceso a la información y servicios. Esto, sumado al constante avance de la tecnología y al surgimiento de nuevos dispositivos, hace que cada vez se trabaje más en el desarrollo de una Web inclusiva, cuyo objetivo es permitir que cualquier persona pueda acceder a sus contenidos independientemente de sus capacidades sensoriales y/o cognitivas, y del medio que utilice para tal fin.

Por lo mencionado anteriormente y a raíz de la aprobación de la Ley Argentina 26.653 de accesibilidad de la información en las páginas Web, aumenta la necesidad de los desarrolladores de sitios web, de contar con herramientas que faciliten y garanticen el desarrollo de interfaces accesibles, a través del cumplimiento de las pautas WCAG 1.0 y WCAG 2.0 publicadas por la W3C.

Todos estos aspectos sirven de motivación para la realización de esta tesina, que consiste en el desarrollo de un analizador intensivo de accesibilidad web basado en reglas, cuyas características abarcan todos los aspectos considerados importantes, y de esta manera contribuir a la formación de una sociedad más inclusiva e igualitaria, con el aporte de una herramienta sin restricciones de uso (Licencia GPL Versión 3.0 [Free Software Foundation Inc., 2007]) a la comunidad de desarrolladores.

A lo largo de este informe, se tratan todos los aspectos considerados para el desarrollo de SiMor, comenzando por una introducción y la especificación detallada de los objetivos.

En el Capítulo 1 se realiza un análisis de la problemática a la que se enfrentan los individuos con discapacidades en su intento de acceder a la información disponible en la Web.

A continuación, en el Capítulo 2, se realiza una introducción al término *accesibilidad web*, donde se explican sus beneficios y se presenta el marco legal inherente a este proyecto, determinado a través de la Ley Nacional de Accesibilidad 26.653 mencionada anteriormente, ligado a un conjunto de leyes y normas internacionales referidas a la accesibilidad web.

Adicionalmente, se presenta el Capítulo 3, conformado por el análisis de un grupo de características predefinidas de los validadores de accesibilidad disponibles en la actualidad, y de las necesidades detectadas durante la realización de una encuesta a profesionales de área, que sirven de base al desarrollo de SiMor, pretendiendo incorporar y mejorar todas aquellas características que se cree deberían facilitar las tareas de un desarrollador web.

Luego, en el Capítulo 4, se presenta SiMor: su especificación, las decisiones de diseño tomadas durante su planificación, las etapas de desarrollo, las metodologías ágiles empleadas para un mejor control del proceso y las tecnologías informáticas utilizadas.

También se expone el diseño de la interfaz de usuario y su adaptabilidad en diferentes dispositivos.

En el Capítulo 5 se detalla el funcionamiento de SiMor, los resultados de su utilización para validarse a sí mismo con el fin de comprobar su condición de accesibilidad y, además, se realiza una comparación con los validadores analizados en el Capítulo 3.

Por último, en el Capítulo 6 se citan las conclusiones resultantes del proceso de análisis, planificación y desarrollo de SiMor.

Objetivos

El desarrollo de SiMor nace con la intención de proporcionar una herramienta de código libre, capaz de analizar sitios web completos con el fin de verificar el cumplimiento de las pautas de accesibilidad establecidas por la W3C¹, e informar al desarrollador los errores encontrados y las posibles soluciones a dichas fallas, con el fin de facilitar su labor a la hora de crear contenido accesible para una Web inclusiva.

Con esta tesina también se pretenden disminuir las barreras de acceso a la información con las que deben lidiar a diario personas con capacidades funcionales diferentes, y de esta manera fomentar aún más el uso de las tecnologías digitales como medio de información y de acceso a productos y servicios.

Al ser una herramienta accesible en sí misma, el objetivo no es solamente contribuir a que los contenidos generados sean accesibles sino también que pueda ser utilizada por cualquier usuario, independientemente de sus habilidades motoras, sensoriales y cognitivas.

¹ El Consorcio World Wide Web (W3C), creado en octubre de 1994, es una comunidad internacional que se dedica al desarrollo de estándares web, dirigido por Tim Berners-Lee.

Capítulo 1: El acceso a la información

En los últimos años Internet se ha convertido en una herramienta muy importante para acceder a diferentes recursos y servicios útiles para la población. A través de ella pueden realizarse numerosas tareas, desde la búsqueda de información general hasta la realización de trámites, manejo de cuentas bancarias, educación a distancia, entre otras.

Es a través de un proceso de comunicación *usuario-sistema* que las personas pueden acceder a la funcionalidad provista por la Web, y es en este punto donde aparece la interfaz gráfica de usuario como medio a través del cual las personas pueden llevar a cabo dicho proceso.

El término “Interfaz gráfica de usuario” (GUI)², aparece en 1970 en el centro de investigación Xerox PARC³, como un método de interacción amigable con las computadoras. [Wilbert O., 1997]

La etimología de la palabra interfaz está compuesta por dos vocablos: *inter*, que proviene del latín *inter*, y significa “entre o en medio”, y *faz* proviene del latín *facies*, y significa “superficie, vista o lado de una cosa”. Por lo tanto una traducción literal del concepto de la palabra interfaz atendiendo a su etimología, podría ser superficie, vista, o lado mediador.

A través de los años la GUI fue transitando por un proceso de madurez donde, en un principio, se definieron sus elementos básicos para terminar convirtiéndose en un producto más bien estético, donde la interfaz, más allá de ser un medio de interacción óptimo, se transformó en un objeto inteligente abierto a los procesos de configuración por parte del usuario.

La GUI exige que los usuarios posean una serie de condiciones fisiológicas determinadas, además del uso de dispositivos que permitan al usuario lograr un contacto directo con el sistema tecnológico. Estos dispositivos, denominados “Dispositivos de interacción hombre-máquina” como el mouse o el teclado, permiten a través de las posibilidades fisiológicas del sujeto, producir parte de la interacción con la interfaz.

Además, en esta interacción, son necesarias ciertas cualidades cognitivas básicas por parte del sujeto, como percibir, decodificar, memorizar, decidir y navegar a través de la GUI, por lo cual la interfaz sólo tiene sentido cuando el sujeto es capaz de comprender el significado y el proceso de interacción, y sus facultades cognitivas capaces de interpretar adecuadamente los signos que se producen sobre la interfaz y usarlas adecuadamente [Luzardo Alliey, 2009].

En la actualidad, los sitios web emplean, en su mayoría, representaciones exclusivamente visuales y multimediales de la información, como animaciones, videos y sonidos que hacen que las páginas y el acceso a su información resulte aún más difícil de interpretar por la diversidad de usuarios y de tecnologías de acceso.

Por lo tanto, la creación de interfaces accesibles ayuda a cumplir con el objetivo para el que fueron

² Del inglés “Graphic User Interface”

³ Xerox PARC (Palo Alto Research Center, ‘centro de investigación de Palo Alto’) es una empresa de investigación y desarrollo, propiedad de Xerox Corporation.

creadas de la GUI's, a través de la reducción de barreras que afectan a las personas con limitaciones sensoriales, motrices o cognitivas en su interacción con los sistemas tecnológicos.

1.1. Problemática en el acceso a la información en la Web

En la sección anterior se realizó una introducción a las interfaces gráficas de usuario y su importante rol en la interacción hombre-máquina. A continuación, y durante el desarrollo de esta unidad, se explican los diversos inconvenientes a los que se enfrentan los diferentes grupos de usuarios en su intento por acceder a los contenidos disponibles en internet.

1.1.1. Barreras de Accesibilidad

Si bien contar con una herramienta como la Web, que permite realizar una gran diversidad de tareas es de gran ayuda para muchas personas, varios grupos quedan excluidos de este beneficio a causa de una variedad de factores que dificultan el acceso a dichos recursos.

Estos factores son denominados *Barreras de Accesibilidad* [WebPosible, 2007], término que hace referencia a cualquier obstáculo presente en un sitio web que impida la percepción de su contenido en la forma en que fue concebido por sus autores [Naftali, 2010].

Barreras de hardware

Las barreras de accesibilidad en el hardware tienen que ver con las características físicas del dispositivo utilizado para acceder a la Web: forma, tamaño, peso, distribución de botones, elementos deslizantes, etc.

Por ejemplo, un tamaño o una disposición poco clara de las características físicas puede confundir a personas mayores. La falta de etiquetas o identificación clara provoca que una persona con problemas visuales no distinga un control del resto; la forma del dispositivo puede ser poco ergonómica⁴ o imponer una postura de manos incompatible con algunos perfiles de discapacidad.

Barreras de software

Las barreras de accesibilidad en el software afectan a la interfaz de control (los canales de comunicación entre dispositivo y usuario) y el método de uso del dispositivo.

Por ejemplo, una persona con problemas visuales se verá afectada por una pantalla en donde se presenta la información con un tamaño pequeño de fuentes, o por el contraste inadecuado de colores que puede dificultar la distinción de textos o elementos.

Las personas con problemas de audición no podrán acceder a los eventos acústicos de los dispositivos si éstos no cuentan con las alternativas de texto correspondientes.

⁴ Relacionado con el estudio de las condiciones de adaptación de una persona o cosa.

Las discapacidades y el acceso a la Web

Cada tipo de discapacidad es susceptible a determinadas barreras, que generalmente están asociadas a la presentación y al estilo del sitio web.

Para comprender la complejidad del problema, es necesario el estudio de la forma en la que cada grupo percibe la Web según el tipo de discapacidad que presenta.

Visual

Dentro de las afecciones visuales, se pueden distinguir diferentes tipos como ceguera, visión débil (pérdida del campo central de visión, visión tipo túnel, visión borrosa), daltonismo (incapacidad de distinguir entre el rojo-verde, amarillo-azul, o incluso incapacidad de distinguir los colores), atrofia óptica, visión reducida, etc.

Según la Organización Mundial de la Salud (OMS) [OMS, 2012], los problemas a los que se pueden enfrentar las personas con este tipo de discapacidad incluyen:

- Imágenes sin un texto alternativo que describa su contenido.
- Imágenes complejas, como gráficos de barras o estadísticas, sin una descripción detallada.
- Elementos multimedia (vídeos, animaciones...) sin descripción textual o sonora.
- Tablas cuyo contenido resulta incomprensible cuando se leen de forma secuencial (celda a celda según el orden en que aparecen en el código HTML).
- Falta de Independencia de dispositivo que no permite usar correctamente la Web con dispositivos de entrada diferentes del mouse (por ejemplo, el teclado). El mouse es un dispositivo de apuntamiento imposible de usar por las personas que no pueden ver dónde está el cursor.
- Formatos no accesibles de documentos que pueden ser problemáticos para un lector de pantalla.
- Tamaño de letra con medidas absolutas que no permiten redefinirlo.
- Diseño de páginas que, al modificar el tamaño de fuente, estropea la maquetación y hace difícil la navegación.
- Poco contraste en las imágenes o texto.
- Texto añadido mediante imágenes que dificulta aumentar el tamaño para facilitar su lectura.
- Uso del color para resaltar texto sin usar otro elemento de formato adicional (como cursiva, negrita o subrayado).
- Poco contraste en las imágenes 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.

Algunos de estos problemas pueden solucionarse con ayudas técnicas (como por ejemplo, magnificadores de pantalla⁵, terminales Braille⁶, navegadores de voz, teclados como dispositivos de entrada, pantallas grandes, etc.) pero otras requieren un diseño web que contemple la forma de navegar tanto de los usuarios, como de las tecnologías asistivas que se utilicen para tal fin.

⁵ Software que simula el efecto de una lupa sobre la pantalla de la computadora.

⁶ Dispositivo periférico que envía la información contenida en la pantalla hasta el usuario utilizando caracteres Braille .

Auditiva

Día a día aumenta aún más el empleo de elementos audiovisuales en la Web. El hecho de que el contenido multimedia no tenga una alternativa textual (por ejemplo subtítulos) para poder ser leído, ya sea por una persona con sordera total o parcial, o un usuario que no tiene altavoces, provoca dificultades, o la total imposibilidad, de obtener la información en los sitios que no tienen en cuenta a este tipo de usuarios.

Según la OMS, los problemas a los que se pueden enfrentar las personas con este tipo de discapacidad incluyen:

- Falta de subtítulos o de transcripciones de los contenidos sonoros.
- Falta de imágenes que ayuden la comprensión del contenido de las páginas. Las páginas con demasiado texto y sin imágenes pueden entorpecer la comprensión a personas cuyo idioma principal es una lengua de signos en lugar del lenguaje escrito o hablado.
- Necesidad de entrada de voz en algunos sitios web.

Este tipo de discapacidad puede mitigarse proporcionando transcripciones o subtítulos y/o alternativas textuales a la entrada de voz.

Motora

Los usuarios afectados por este tipo de discapacidad pueden tener problemas para el manejo del mouse y/o teclado. Las barreras que se encuentran bajo esta categoría están relacionadas al uso de eventos de Javascript que no son universales, como por ejemplo hacer click con el mouse sobre una imagen, navegación imposible a través del tabulador, uso de Flash⁷ para la navegación, etc.

Según la OMS, los problemas a los que se pueden enfrentar las personas con este tipo de discapacidad incluyen:

- Íconos, 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 dispositivo que impide usar correctamente la Web con el teclado en vez de con el mouse.
- Tiempos de respuesta limitados para interactuar con la página.

Existen diversas “tecnologías asistivas o de apoyo”⁸ que pueden beneficiar a estos usuarios, que van desde dispositivos apuntadores (como un mouse de cabeza, o una palanca manejada por la boca) y software de reconocimiento de voz, hasta teclados adaptados al rango de movimiento del usuario, y también sistemas de seguimiento de ojos.

El conocimiento de estas herramientas ha sido de gran relevancia en el desarrollo de las Pautas de Accesibilidad al Contenido en la Web (WCAG), porque su funcionamiento está estrechamente

⁷ Plataforma multimedia y de software creada por Adobe.

⁸ Cualquier producto (incluyendo dispositivos, equipos, instrumentos, tecnología y software) que facilitan la ejecución de actividades o la interacción con el entorno físico y social de las personas en situación de discapacidad.

ligado a una buena estructuración del código del contenido web.

Cognitiva y neuronal

Ejemplos de estas discapacidades son la dislexia⁹, discalculia¹⁰ y el síndrome de Down. Son usuarios que tienen dificultades para aprender, memorizar o mantener fija su atención. Incluso también están en esta categoría las personas bajo influencia de medicación o alguna otra sustancia que pueda potenciar alguno de los síntomas nombrados.

Los problemas al utilizar un sitio web van desde una navegación compleja (por ejemplo, no distinguir claramente la sección del menú de navegación con la de contenido), el uso de términos complicados aunque sean de uso común (vocabulario administrativo, judicial, bancario, etc.), no usar los convenios establecidos (como el color y subrayado de los vínculos), no ofrecer información de contexto en el sitio web que impida una navegación fluida, y también el uso y abuso de animaciones, pantallas emergentes, sonidos de fondo, entre otros, que pueden provocar distracciones al usuario.

Según la OMS, los problemas a los que se pueden enfrentar las personas con este tipo de discapacidad incluyen:

- Elementos visuales o sonoros que no se puedan desactivar cuando se desee y que puedan distraer a las personas con déficit de atención.
- Falta de una organización clara y coherente de la información que ayude a las personas con problemas de memoria o con escasa capacidad cognitiva.
- Lenguaje complejo.
- Ausencia de gráficos en los sitios web que complementen la información textual.
- Tamaño de letra fijo que no se pueda modificar.
- Destellos o parpadeos con altas frecuencias que pueden provocar ataques de epilepsia.

Las soluciones a estos problemas pueden encontrarse a través de las siguientes tecnologías asistivas:

- Lector de pantalla para facilitar la comprensión a las personas con dificultades para la lectura.
- Subtítulos que faciliten la comprensión de un contenido sonoro a las personas con dificultades para procesar la información auditiva.
- Desactivación de los elementos multimedia (animaciones, gráficos y sonido) para concentrarse en el contenido.
- Desactivación de animaciones, sonidos y todos los objetos que puedan provocar ataques.

También es posible disminuir este tipo de barrera mediante una configuración adecuada de los navegadores (desactivación de ventanas emergentes, animaciones, sonidos), pero principalmente mediante el uso de un buen diseño, como crear un sitio web con una estructura y navegación clara, simple e intuitiva (información de navegación y sesión, visualización de rangos de entrada

⁹ Trastorno de desarrollo de la lectura.

¹⁰ Dificultad de aprendizaje específica en matemáticas.

admisibles, ejemplos, formatos), o la utilización de estándares.

Por norma general, este tipo de usuarios no usan productos de apoyo para acceder a la información, sino que dependen sobre todo del diseño del sitio web.

Discapacidades relacionadas con la edad

Cada vez más, las personas de edad avanzada hacen uso de la Web, por lo que es un factor que debe tenerse en cuenta a la hora de desarrollar contenido para la misma.

Estas personas tienen una mayor predisposición a padecer limitaciones visuales, auditivas, u otras, y por tanto cualquier esfuerzo enfocado en la accesibilidad va a beneficiar a este amplio grupo.

Otras barreras

A todos los tipos de barreras mencionadas anteriormente hay que añadir los problemas de acceso que puede tener una persona por motivos de idioma, conocimientos y/o experiencia, la utilización de navegadores discontinuados, conexión a internet de baja velocidad, y otras dificultades en la conectividad.

1.2. Conclusión

Luego del análisis de los diversos factores que implican restricciones en el acceso al contenido de la Web, se entiende que existe una cantidad considerable de personas que, ya sea por aspectos humanos, técnicos o socioeconómicos, quedan excluidas no sólo del acceso a la información, sino también de la realización de trámites, la utilización de servicios web de comunicación y, en general, toda la amplia gama de funciones que ofrece Internet en la actualidad; siempre y cuando dicha diversidad no sea considerada por parte del desarrollador.

En la era de la sociedad de la información, independientemente de las obligaciones legales en materia de accesibilidad que deban respetarse, cumplir con las normas de accesibilidad web supone una obligación moral para todas aquellas personas, empresas u organizaciones que aportan contenido a la Web.

Capítulo 2: Accesibilidad Web

Como se explicó anteriormente existe una gran diversidad de factores que dificultan el acceso a los contenidos y funcionalidad proporcionados por la Web, lo que implica para un grupo numeroso de personas un obstáculo que les impide el acceso a las nuevas tecnologías en la etapa que actualmente atraviesa la sociedad.

En este capítulo se presenta el término *accesibilidad*, y los beneficios que produce su aplicación en el desarrollo de sitios web, tanto a nivel social como empresarial.

Además se presenta el marco legal nacional e internacional, y los estándares citados a continuación:

- Pautas de Accesibilidad para el Contenido Web (WCAG) 1.0, recomendado en la Guía de Accesibilidad 1.0 para Sitios Web del Sector Público Nacional, aprobada por Disposición 4/2011 de la Oficina Nacional de Tecnologías de Información (ONTI). [Disposición 4, 2011]
- Pautas de Accesibilidad para el Contenido Web (WCAG) 2.0 aprobado como estándar internacional por ISO/IEC 40500:2012.
- Pautas de Accesibilidad para Agentes de Usuario (UAAG) 2.0.
- Herramientas de edición de contenidos (ATAG) 2.0.
- Estándares definidos por la Sección 508.
- Estándar WAI-ARIA (Aplicaciones de Internet Enriquecidas).

2.1. Introducción

Según la W3C, el término accesibilidad web “significa que personas con algún tipo de discapacidad puedan hacer uso de la Web. En concreto, al hablar de accesibilidad Web se está haciendo referencia a un diseño que permita que estas personas puedan percibir, entender, navegar e interactuar con la Web, aportando a su vez contenidos.” [Shawn Lawton Henry et al. - W3C. 2005]

La accesibilidad web beneficia a varios grupos de personas. Entre ellos podemos destacar a los de personas de edad avanzada que tienen sus habilidades disminuidas, e incluso a organizaciones que difunden sus productos, servicios e información a través de la misma, permitiendo su consumo por una mayor cantidad de personas. Por otro lado aparecen los usuarios con discapacidades, que utilizan las computadoras a través de tecnologías asistivas.

En un principio, las páginas web eran muy simples y su contenido estaba conformado en su mayoría por textos planos. A medida que la Web fue creciendo, el lenguaje HTML evolucionó hacia una Web más gráfica y tanto la estructura como la presentación de los documentos se volvieron más complejos, dificultando el trabajo de los sistemas de apoyo.

Esencialmente, la accesibilidad web tiene como principio la satisfacción de las necesidades particulares de todas aquellas personas que utilizan la Web.

En la actualidad, la mayoría de los sitios web presentan barreras que dificultan o imposibilitan su uso a una gran cantidad de personas, principalmente aquellas que conforman los grupos anteriormente mencionados; efecto que puede disminuirse con el desarrollo de software y sitios web accesibles.

2.2. Beneficios que ofrece la accesibilidad web

Hoy en día la accesibilidad web es un punto relevante y necesario para el futuro de la Web, tanto a nivel servicio como negocio.

Thatcher J. et al., en su libro *Web Accessibility - Web standards and regulatory*, explica los diversos beneficios que aporta el desarrollo de un sitio web accesible, tanto a nivel social como empresarial. A continuación, se explican y analizan dichos beneficios.

Disminución de las barreras de acceso

En las figuras 2.2.1 y 2.2.2 se estima, según el Censo Argentino 2010, la cantidad de personas a quienes se les dificulta la utilización de la Web si su contenido no es diseñado cumpliendo con los estándares recomendados en este aspecto. [INDEC, 2012]

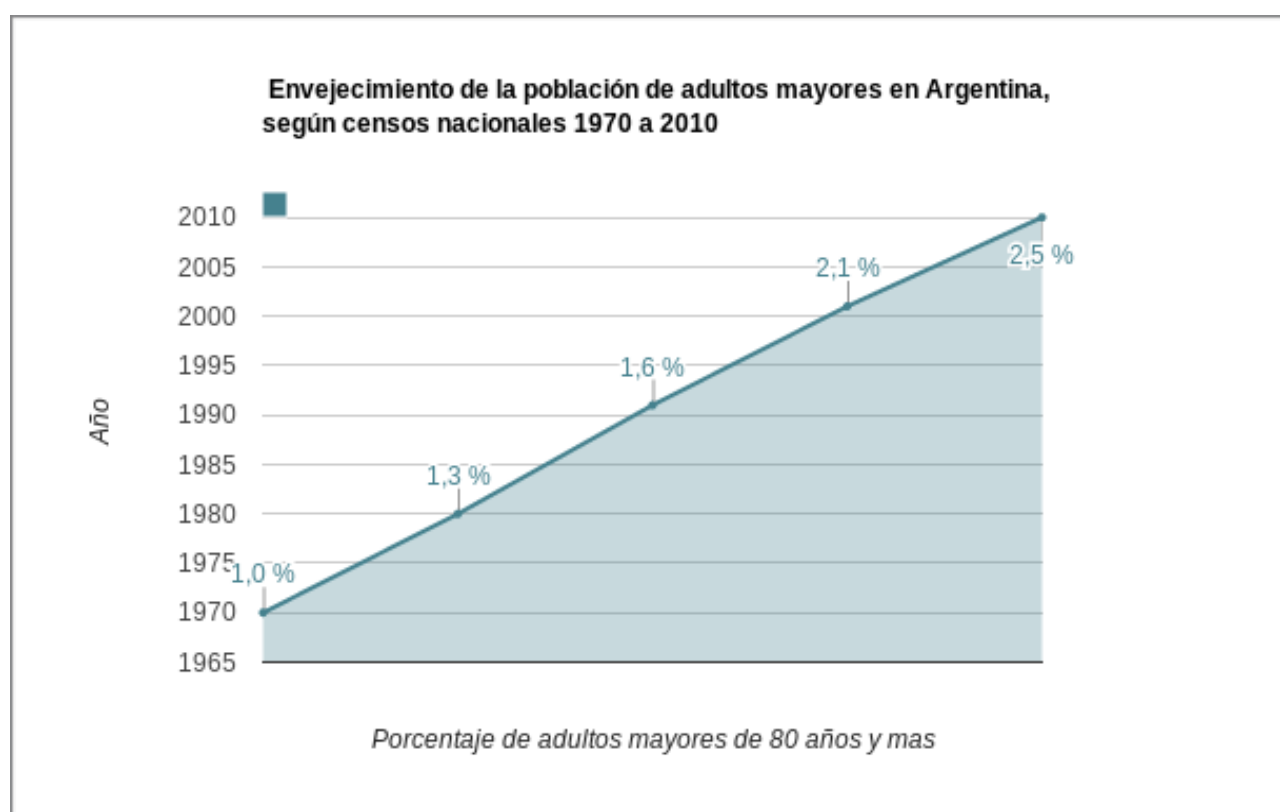


Figura 2.2.1: Envejecimiento de la población de adultos mayores en Argentina, según censos nacionales 1970 a 2010.

El envejecimiento de la población es uno de los fenómenos de mayor impacto del siglo y se trata de un proceso que se da a nivel mundial, siendo Argentina uno de los países con más alta tasa de

envejecimiento de Latinoamérica. Según el Censo 2010, como se muestra en la figura 2.2.1, se observa que, a pesar de que en la última década la población creció en casi cuatro (4) millones de personas, hay 25.378 chicos menos que en 2001. En cambio, hay 517.028 mayores de 65 años más. Además, ya son casi 3.500 los habitantes mayores de 100 años. En 2001 eran 1.855.

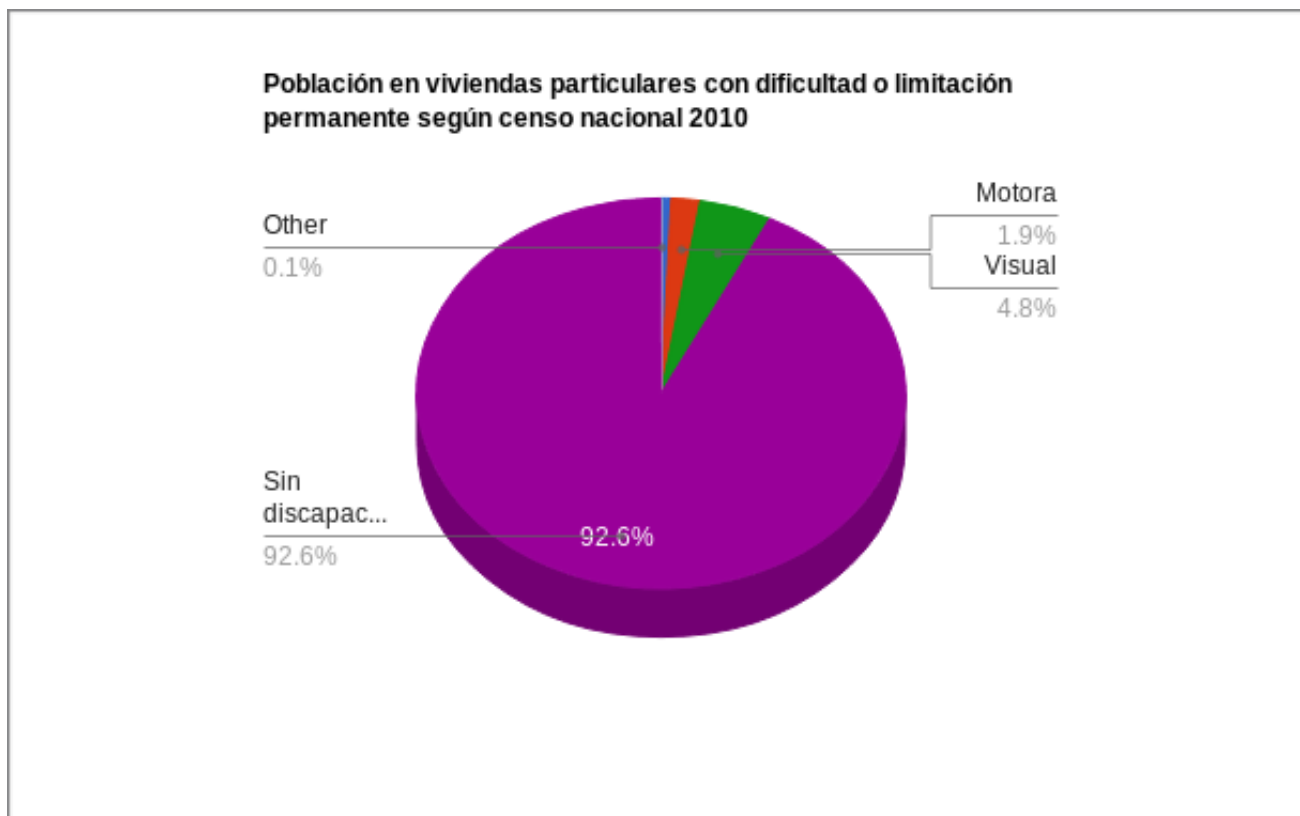


Figura 2.2.2: Población en viviendas particulares con dificultad o limitación permanente según censo nacional 2010.

La figura 2.2.2 expone la situación de las discapacidades presentes en la población total, en la que se puede ver que hay casi un 8% de personas con alguna limitación que dificulte el acceso a la información.

Además de las discapacidades innatas que una persona pueda tener, a lo largo de la vida se pueden sufrir episodios de discapacidad temporal, y es muy probable que se padezca alguna durante la vejez por lo que existe un alto grado de correlación entre la edad y la discapacidad. En consecuencia, la discapacidad debe ser vista como un hecho que afectará a la mayoría de las personas en el tiempo. Por lo tanto, los esfuerzos en accesibilidad que se hagan hoy resultarán en beneficios para todas las personas, incluidas las que no son discapacitadas actualmente.

Disminuye los costos de desarrollo y mantenimiento

El desarrollo de sitios web accesibles, y la incorporación de la accesibilidad como parte transversal en todas las etapas metodológicas, supone un costo extra pero una vez que se poseen los conocimientos, el costo de mantener y desarrollar un sitio accesible se ven disminuidos ya que ciertas condiciones y requisitos técnicos que hacen a la accesibilidad de un sitio web contribuyen a mejorar los procesos de desarrollo. Conceptos como la separación de contenido y presentación, y

la utilización de estándares recomendados, facilitan el desarrollo, el mantenimiento, y la actualización de las aplicaciones.

Aumenta la compatibilidad y mejora la interoperabilidad de la página web

Implica indirectamente que la página podrá ser visualizada desde cualquier navegador, y desde cualquier dispositivo independientemente de la resolución y/o tamaño de pantalla, tipo de dispositivo, y otras cualidades que hacen a la visualización.

Muchas organizaciones están incrementando su interés en que sus sitios sean interoperables e independientes del dispositivo utilizado para el acceso; los sitios web que adopten los principios de la independencia de dispositivo estarán bien posicionados para añadir semántica, permitiendo el filtrado y selección de contenido relevante para dispositivos con menor ancho de banda o aquellos que requieren características de navegación diferentes.

Preparación para tecnologías avanzadas

La accesibilidad web puede ayudar a las organizaciones a aprovechar las tecnologías web avanzadas y estar preparados para tecnologías futuras. Por ejemplo a través de la utilización de metadatos y su representación mediante RDF¹¹, lo que permite la reutilización de contenido.

Como se explica en el artículo *Beneficios Auxiliares del Diseño Web Accesible* [Andrew Arch, 2002], la metainformación es la primera implementación de información estructurada sobre recursos en la Web. Cuando se apliquen conceptos de metainformación, a través de RDF harán posible la recuperación y cotejo automático de la información.

Además, separar la estructura de la presentación añade definición a las páginas en un formato reconocible por máquinas, como son las etiquetas HTML, y permiten la migración desde la "World Wide Web" hacia la "Web Semántica"; donde los agentes de usuario podrán encontrar información del sitio de manera automática y precisa.

Optimización de motores de búsqueda (SEO)

Construir un sitio web accesible puede mejorar significativamente el SEO, aspecto muy importante para algunas organizaciones.

El contenido que no está basado en texto no está disponible para los motores de búsqueda u otras aplicaciones automáticas de explotación de datos (data-mining). El contenido más importante de un sitio puede encontrarse en formatos que no puedan ser indizados. Al utilizar texto para describir dicho contenido, se incrementa significativamente el número de personas que, en busca de un contenido en particular, puedan encontrarlo.

Las siguientes pautas, contempladas dentro de las WCAG, permiten a los buscadores encontrar, indexar y clasificar los sitios:

- Asegurarse que el tag HTML *title* y los atributos *alt* sean descriptivos.
- Comprobar que el código HTML sea correcto.
- Hacer un sitio con vínculos (links) claros.
- Incluir tags *H1* y/o *H2* en cada página.

¹¹ Resource Description Framework diseñado por la W3C como un modelo de datos para metadatos.

Reduce el tiempo de carga de las páginas web y la carga del servidor web

Al separar el contenido de la presentación de un sitio web mediante CSS¹² se logra reducir el tamaño de las páginas web y, por tanto, se reduce su tiempo de carga. Además, se reduce la descarga de imágenes y archivos multimedia de gran tamaño, ya que al incluir un texto alternativo para estos elementos se permite que, por ejemplo, usuarios con un bajo ancho de banda naveguen el sitio con el navegador configurado para que no se descarguen automáticamente las imágenes, y mediante el texto que se les presenta decidan si efectuar la descarga o no [Shawn Lawton Henry - W3C, 2012].

Esto es muy importante para las personas que acceden a Internet a través de dispositivos móviles o poseen conexiones muy lentas.

En las figuras 2.2.3 y 2.2.4 se estiman los porcentajes de individuos con estas características.

Como puede observarse en la figura 2.2.3, la conexión a Internet argentina calificó como una de las más lentas del mundo. Con 4,20 Mbps de velocidad promedio en las descargas, la banda ancha local se ubicó en el puesto número 109 de un ranking mundial creado por la empresa norteamericana Ookla.

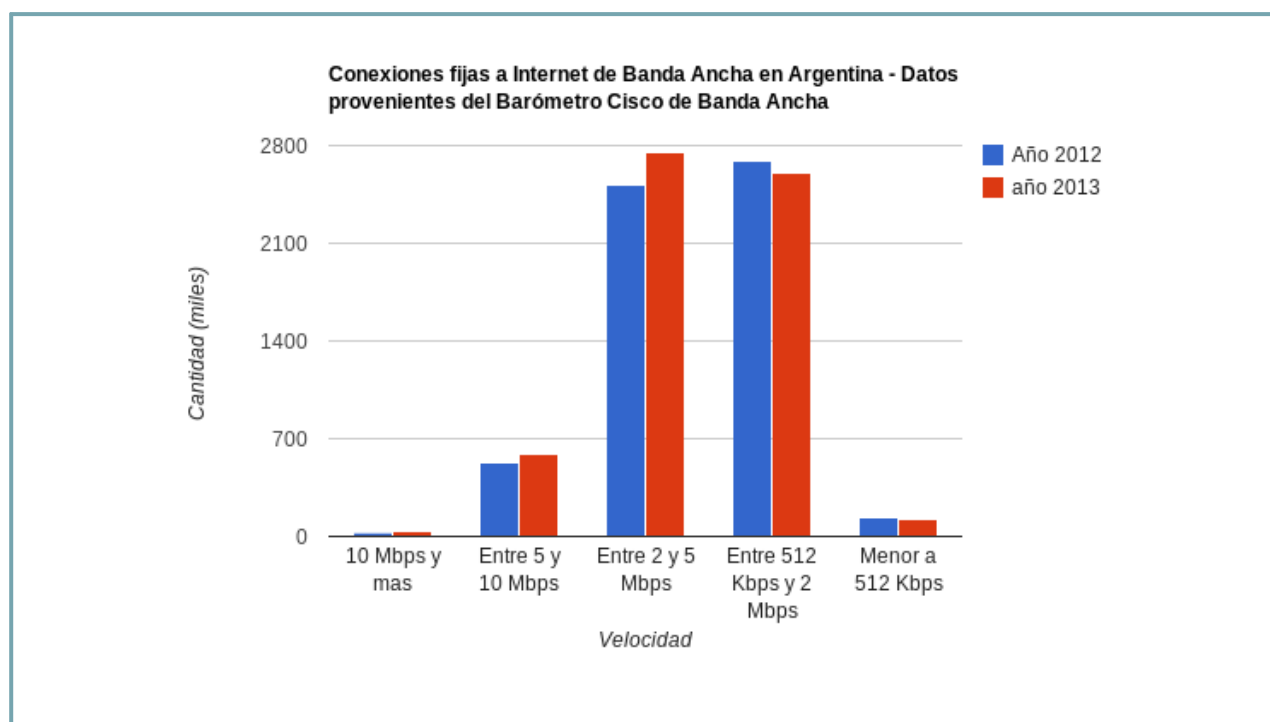


Figura 2.2.3: Conexiones a internet de banda ancha en Argentina - Datos provenientes del Barómetro Cisco de Banda Ancha. [Info News, 2013]

Como muestra la figura 2.2.4, el uso de los dispositivos móviles ha aumentado en los últimos años ganando terreno como instrumento de acceso a Internet gracias al bajo costo en comparación con las computadora, además de su tamaño compacto y portátil.

¹² Cascading Style Sheets (Hoja de estilo en cascada) es un lenguaje usado para definir la presentación de un documento estructurado escrito en HTML o XML.

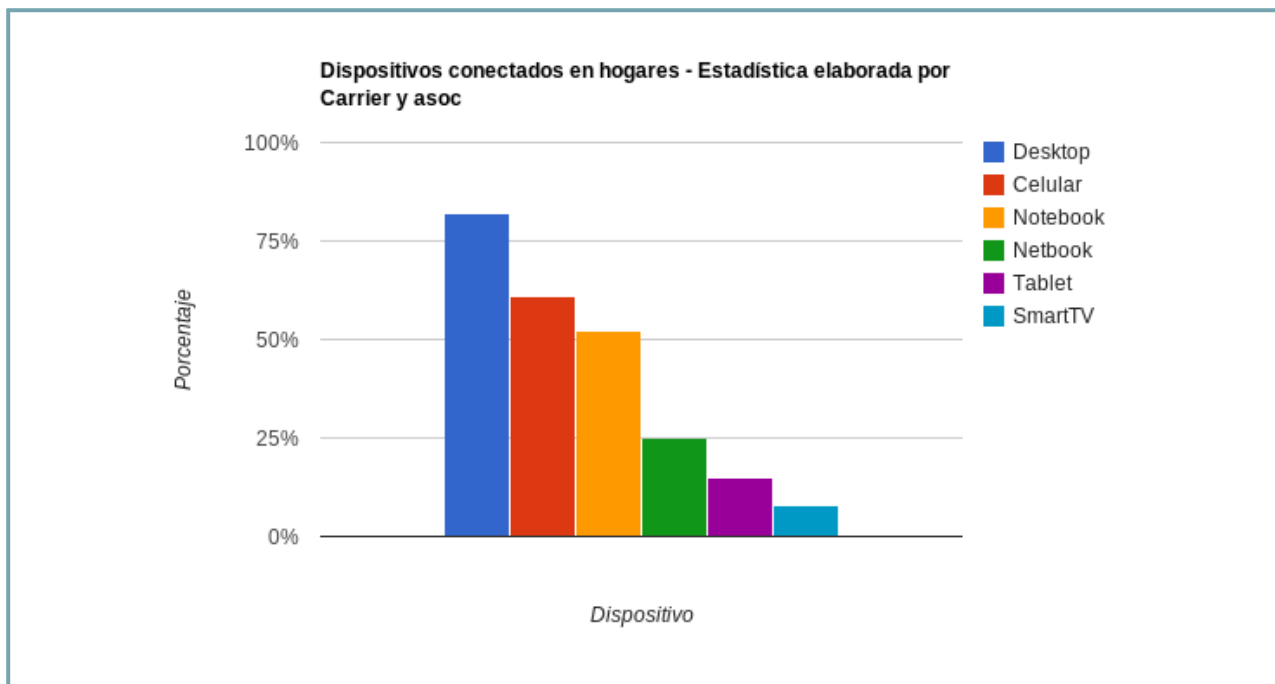


Figura 2.2.4: *Dispositivos conectados en hogares. Estadística elaborada por Carrier y asoc. [FrecuenciaLatinoamérica, 2013]*

A partir de las figuras anteriores, nuevamente se concluye que en Argentina existe un gran número de personas para las cuales el contenido web es de difícil acceso, por lo que la construcción de sitios accesibles es de vital importancia para este grupo de usuarios.

2.3. Construcción de sitios web accesibles

“La construcción de sitios web accesibles puede resultar sencilla o compleja dependiendo de muchos factores, entre los cuales pueden destacarse el tipo de contenido, el tamaño y la complejidad del sitio, y/o las herramientas de desarrollo y el entorno que se utilicen para tal fin” [Shawn Lawton Henry et al. - W3C, 2005]

A partir de esto se puede deducir que es primordial que los diferentes aspectos y componentes del desarrollo web sean abordados conjuntamente para que la Web sea accesible para la mayor cantidad de personas posible.

Estos componentes incluyen:

- Contenido/información presente en una página web o en una aplicación web, incluyendo:
 - Texto, imágenes y sonidos.
 - Código o etiquetas que definen estructura, presentación, etc.
- Navegadores web, reproductores multimedia y otros "agentes de usuario".
- Tecnologías asistivas (lectores de pantalla, teclados alternativos, software de escaneo, etc.).
- Conocimiento de los usuarios, experiencias y, en ocasiones, estrategias de adaptación para la utilización de la Web.

- Desarrolladores, diseñadores, codificadores, autores, etc., incluyendo desarrolladores que tienen alguna discapacidad y usuarios que proporcionan contenido.
- Herramientas de autor (software para la creación de sitios web) .
- Herramientas de evaluación, validadores de accesibilidad web, validadores de HTML, validadores de CSS, etc.

Muchas de las características accesibles de un sitio se implementan de forma sencilla si se planean desde el principio del desarrollo del sitio web o al comienzo de su rediseño. La modificación de sitios web inaccesibles puede requerir un gran esfuerzo, sobre todo aquellos que no fueron desarrollados cumpliendo correctamente con los estándares.

Para facilitar esta tarea existen, por ejemplo, las WCAG que proporcionan información detallada de los pasos básicos para introducir la accesibilidad en un proyecto web por parte de los desarrolladores. [Shawn Lawton Henry et al. - W3C, 2012].

2.4. Estándares vigentes

En la actualidad existen diversos estándares referentes a accesibilidad web. A continuación se explican algunas de las más importantes.

2.4.1. World Wide Web Consortium (W3C)

El Consorcio World Wide Web (W3C), creado en octubre de 1994, es una comunidad internacional que se dedica al desarrollo de estándares web, dirigido por Tim Berners-Lee, mentor de la Web en el año 1989, quien fue además el creador de las principales tecnologías sobre las que se basa la Web.

Dichas tecnologías abarcan desde URL (Uniform Resource Locator; Localizador Uniforme de Recursos), HTTP (Hypertext Transfer Protocol; Protocolo de Transferencia de Hipertexto) y HTML (Hypertext Markup Language; Lenguaje de Marcado de Hipertexto), hasta tecnologías de semántica, como OWL, y de seguridad como APPEL.

A continuación, en la figura 2.4.1.1, se detalla el abanico completo de las tecnologías que conforman la pila tecnológica de la W3C, para las cuales el organismo propone y define los estándares necesarios para evitar la fragmentación del mercado y así lograr construir una Web única; en donde los formatos y protocolos deben ser compatibles unos con otros y a su vez permitir que cualquier hardware y software que se utilice como medio para acceder a la Web funcione conjuntamente.

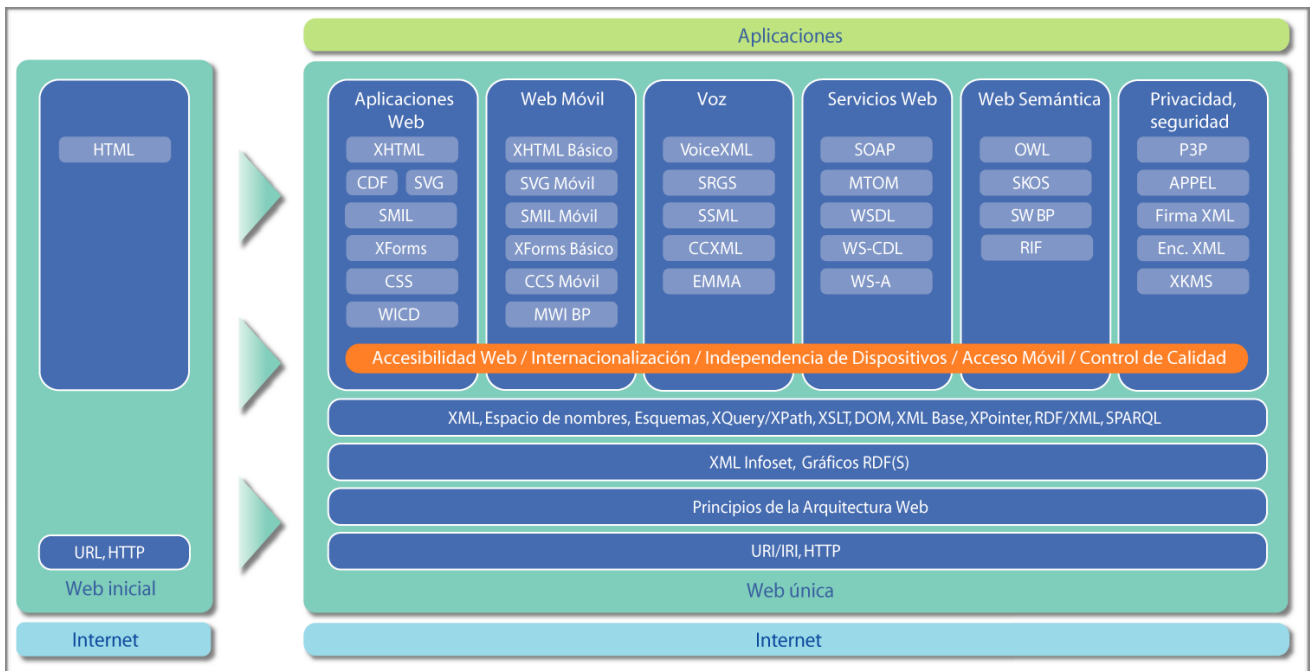


Figura 2.4.1.1: Pila tecnológica de la W3C.

Fuente W3C. <http://www.w3c.es/Consortio/tecnologia.html>

La W3C impulsa la Iniciativa de Accesibilidad Web (WAI) surgida en el año 1997, que pretende, mediante una labor educativa y de concientización, facilitar el acceso de personas con discapacidad a la Web, impulsando el desarrollo de pautas y la mejora de herramientas de evaluación y corrección de accesibilidad. [Shawn Lawton Henry and Judy Brewer - W3C, 2005]

Para hacer esto posible, la WAI ha emitido las denominadas “Pautas de Accesibilidad del Contenido en la Web (WCAG)” donde se especifican las condiciones que deben cumplirse al desarrollar un sitio web para hacer accesibles sus contenidos. Estas pautas, específicamente las WCAG 2.0 que se explican en el inciso 2.4.3, son las bases fundamentales del análisis realizado por SiMor. Cada pauta contiene sugerencias y ejemplos a seguir para lograr la conformidad con cada uno de sus puntos de control. [Ben Caldwell et al. - W3C, 2008]

Existen tres componentes diferenciados, como se observa en la figura 2.4.1.2, sobre los cuales la WAI genera pautas de accesibilidad:

- **Herramientas de autor y agentes de usuario:** Navegadores web, reproductores multimedia, tecnologías de apoyo, y otros "agentes de usuario" para llegar e interactuar con el contenido web.
- **Tecnologías asistivas:** Herramientas de creación y evaluación para crear contenido web utilizadas por desarrolladores web.
- **Contenido web:** Información en una página web o una aplicación web, incluyendo texto, imágenes, formas, sonidos, etc.

Además se muestra la relación entre los distintos componentes a los que se aplica accesibilidad.

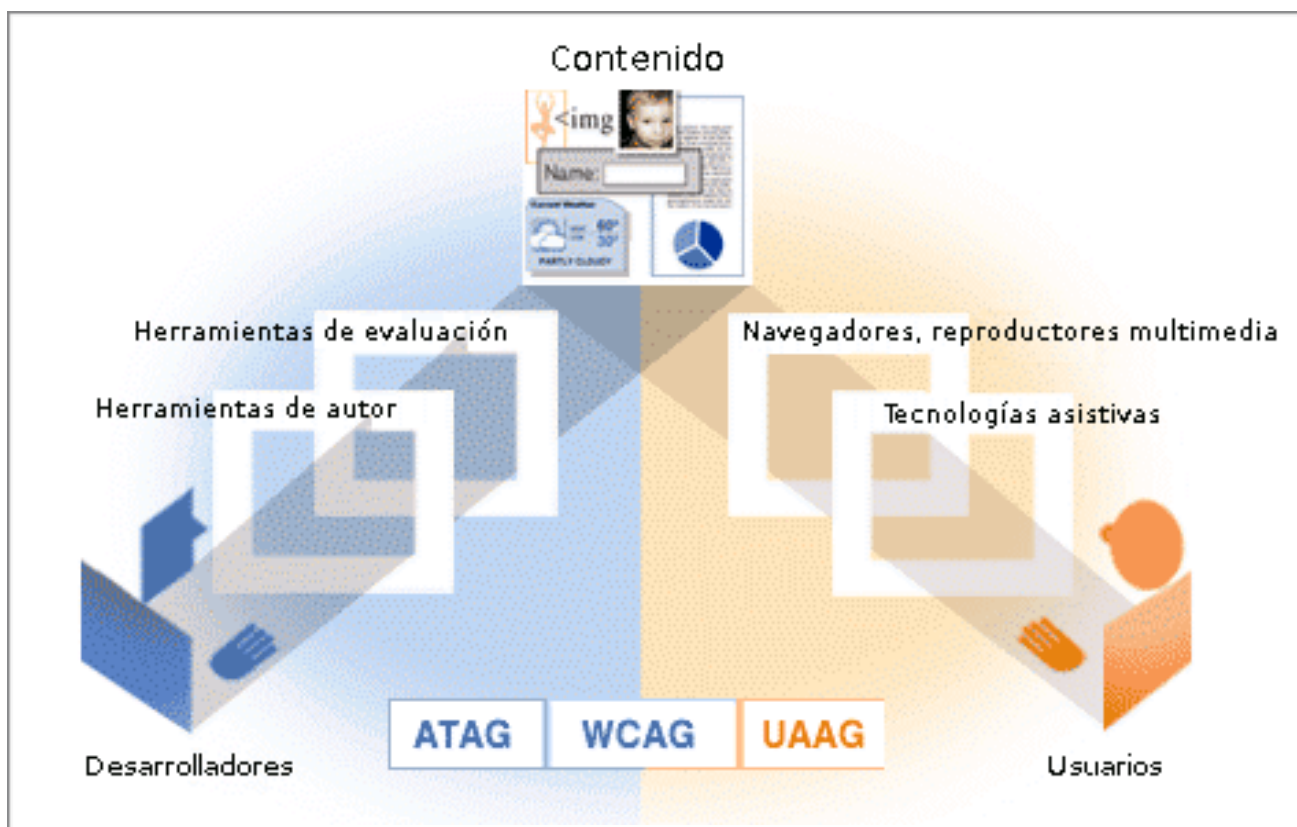


Figura 2.4.1.2: Componentes de Accesibilidad Web - W3C.
Fuente W3C. <http://www.w3.org/WAI/intro/components.php>

Estos componentes son tenidos en cuenta por la WAI ya que los desarrolladores web suelen utilizar herramientas de autor y herramientas de evaluación para crear contenido web, y las personas (usuarios) utilizan los navegadores web, reproductores multimedia, tecnologías de apoyo u otros "agentes de usuario" para obtener e interactuar con el contenido.

2.4.2. Pautas WCAG 1.0

Las Pautas de Accesibilidad al Contenido en la Web 1.0 (WCAG 1.0) [Wendy Chisholm et al., 1999] fueron publicadas el 5 de mayo de 1999 y están compuestas por 14 pautas que constituyen los principios generales para el diseño accesible.

En la actualidad las WCAG 1.0 han sido sustituidas por las WCAG 2.0, a causa de sus limitaciones referentes a las tecnologías en las cuales pueden ser aplicadas, su ambigüedad de interpretación y rigidez por su falta de actualización.

A pesar de lo mencionado anteriormente, las WCAG 1.0 son el punto de partida de las sucesivas recomendaciones y siguen siendo un referente válido para solucionar las principales barreras de accesibilidad.

Muchos países han adoptado las WCAG 1.0 o una variación de éstas, junto con reglas particulares

para esa jurisdicción. Otros países como Australia o Dinamarca, en cambio, han adoptados los estándares web de la sección 508 de Estados Unidos, detallada en la sección 2.4.6.

Es importante conocer y entender la importancia de la aplicación de estas pautas ya que su uso y aplicación permite, además de los beneficios mencionados anteriormente, el cumplimiento de la legislación vigente en Argentina [Ley 26.653, 2010].

Estructura de las pautas WCAG 1.0

Las WCAG 1.0 están conformadas por catorce (14) pautas o principios generales de diseño accesible, y cada una de ellas incluye:

- Número de pauta.
- Declaración del principio.
- Fundamento de la pauta y algunos grupos de usuarios que resultan beneficiados por su cumplimiento.
- Lista de definiciones de los puntos de verificación: Estas definiciones explican cómo se aplica la pauta en situaciones típicas de desarrollo de contenido. Cada definición de punto de control incluye:
 - El número de punto de verificación.
 - La declaración del punto de verificación.
 - Su prioridad
 - Notas informativas, ejemplos de aplicación y referencias cruzadas a las directrices o puntos de verificación relacionados.

Principios fundamentales

A continuación se explican los principios fundamentales definidos por las WCAG 1.0.

Pauta 1: Proveer alternativas equivalentes para el contenido visual y auditivo

El contenido presentado al usuario debe ofrecer una alternativa equivalente para ser entendido tanto de forma visual como auditiva, ya que algunos usuarios con deficiencia visual pueden usar interpretadores que transforman texto en locución.

Esta pauta enfatiza el uso de texto equivalente (o alternativo) para contenido no textual.

El contenido en el formato de texto puede ser interpretado por sintetizadores de voz, tabletas Braille o en distintos tipos de pantallas.

Pauta 2: No basarse sólo en el color

Esta pauta hace referencia a que no deben identificarse determinadas acciones sólo con colores, sino que debe proveerse información adicional como texto, gráficos u otros elementos visuales para indicar su significado, ya que algunas personas pueden no distinguir ciertos colores.

Por ejemplo, utilizar solamente el color rojo para indicar un texto de alerta, puede no ser comprendido en sistemas no visuales.

Pauta 3: Utilizar marcadores HTML y hojas de estilo apropiadamente

Los documentos HTML deben usar los marcadores disponibles adecuadamente respetando su especificación.

Se deben utilizar hojas de estilo para especificar la presentación del HTML, en lugar de atributos del tipo "style".

Pauta 4: Identificar el idioma usado

Esta pauta implica usar marcadores que faciliten la pronunciación o interpretación de texto abreviado o extranjero. Se debe indicar el idioma predominante en cada página y marcar aquellas expresiones que se encuentren en otra lengua. De esta forma, los sintetizadores de voz son capaces de cambiar su pronunciación en función del idioma.

Pauta 5: Crear tablas que se transformen correctamente

Las tablas sólo se utilizan para marcar información tabular (tablas de datos). El uso de tablas con otros fines crea dificultades para los usuarios que usan lectores de pantalla. De igual forma, las tablas mal estructuradas (por ejemplo, sin encabezados <th>) dificultan la lectura a usuarios que no pueden visualizar la información de forma global: personas ciegas con lectores de pantalla y/o dispositivos Braille, personas con deficiencias visuales que utilizan magnificadores de pantalla o, simplemente, usuarios con dispositivos de pantalla pequeña.

Pauta 6: Asegurar que las páginas que incorporen nuevas tecnologías se transformen correctamente

Una página web basada en tecnologías modernas tiene que ser accesible igualmente si el navegador no tiene habilitadas dichas tecnologías (como puede ser Javascript), o al ser visualizada con navegadores antiguos. El usuario puede desconectar las tecnologías más modernas para ganar en rapidez de descarga. Sin embargo, los contenidos deben permanecer accesibles.

Pauta 7: Asegurar al usuario el control sobre los cambios de los contenidos dependientes del tiempo

El movimiento de los objetos o páginas, su parpadeo o actualización automática deben poder ser controlados por el usuario; las personas con discapacidades cognitivas o visuales no pueden leer textos en movimiento. De forma similar, algunos discapacitados físicos no pueden interactuar con objetos móviles (limitaciones motrices).

Pauta 8: Asegurar la accesibilidad directa de las interfaces incrustadas

Cuando un sitio contiene objetos incrustados (Adobe Flash®, Java Applet, etc.) con interfaz propia, ésta también debe ser accesible. Si la interfaz del objeto incrustado no puede hacerse accesible, debe proporcionarse una solución alternativa accesible.

Pauta 9: Diseñar para la independencia del dispositivo

Esta pauta hace referencia a que el usuario debe poder interactuar con la aplicación en su dispositivo de entrada/salida preferido (mouse, teclado, voz, monitor, tablet, etc.). Si, por ejemplo, un campo de un formulario sólo puede ser activado con el mouse u otro dispositivo de apuntamiento, alguien que use la página con entrada de voz, con teclado u otro dispositivo de entrada, no será capaz de utilizar el formulario.

Pauta 10: Utilizar soluciones provisionales para que las ayudas técnicas y los antiguos navegadores funcionen de forma correcta

Esta pauta indica que, mientras los navegadores antiguos y ayudas técnicas no operen correctamente, es imprescindible la utilización de alternativas accesibles.

Pauta 11: Utilizar las tecnologías y pautas W3C

Cuando no se puedan usar tecnologías desarrolladas por la W3C, o al usarlas no se obtengan contenidos accesibles, deben proporcionarse versiones alternativas. Se recomiendan las tecnologías W3C por incluir características accesibles incorporadas, y estar desarrolladas en un proceso abierto consensuado y por ser utilizadas como base para la creación de contenidos accesibles.

Pauta 12: Proveer información de contexto y orientación

Esta información es de ayuda al usuario para comprender páginas o elementos complejos. Se deben agrupar los elementos y ofrecer información sobre la relación entre ellos. Esta acción es fundamental para personas con discapacidades cognitivas y visuales.

Pauta 13: Proveer mecanismos claros de navegación

La utilización de mecanismos claros de navegación facilita a los usuarios hallar la información que necesitan, lo que es fundamental para personas con discapacidades cognitivas y visuales. Algunas maneras de brindar esta información es mediante la inclusión de un mapa de sitio, ayudas, barras de navegación, etc.

Pauta 14: Asegurar que los documentos sean claros y simples

La utilización de lenguaje claro y simple facilita la comunicación de la información. El acceso a la información escrita puede ser difícil para personas con discapacidades cognitivas o con dificultad de aprendizaje, y para personas sordas o que hablan en una lengua extranjera. La comprensión de un documento también depende de la maquetación de la página y de los gráficos (que deben llevar un texto alternativo).

Cada una de estas pautas tiene asociados puntos de verificación (65 en total) que explican cómo se aplica la pauta. A su vez, cada punto de verificación tiene asignada una prioridad (1, 2, 3) que permite calcular el nivel de adecuación.

Prioridad 1: Son aquellos puntos de verificación que deben ser necesariamente aplicados en el desarrollo web. En caso contrario, seguramente el sitio web no sea accesible.

Prioridad 2: Estos puntos de verificación son recomendaciones con una importancia teórica menor que los de Prioridad 1. En la práctica, su aplicación es necesaria si se quiere obtener un sitio web accesible, puesto que conllevan un conjunto de buenas prácticas muy recomendables.

Prioridad 3: Son las recomendaciones de prioridad menor, y los desarrolladores pueden aplicarlas para mejorar la accesibilidad del sitio web.

A partir de estas prioridades se calcula el nivel de adecuación de accesibilidad (nivel de conformidad), estableciendo tres (3) categorías:

Simple - A (A): cuando se cumplen todos los puntos de verificación de prioridad 1.

Doble - A (AA): cuando se cumplen todos los puntos de verificación de prioridad 1 y 2.

Triple - A (AAA): cuando se cumplen todos los puntos de verificación de prioridad 1, 2 y 3.

2.4.3. Pautas WCAG 2.0

En diciembre de 2008, la W3C (WAI) publicó las Pautas WCAG 2.0 [Ben Caldwell et al. - W3C, 2008], que fueron desarrolladas con el fin de adaptarse a los cambios tecnológicos producidos en los últimos años. A diferencia de las WCAG 1.0, desarrolladas exclusivamente para tecnologías HTML y CSS, las nuevas pautas son de aplicación en cualquier tecnología.

La W3C recomienda que los contenidos nuevos y actualizados apliquen las pautas 2.0 y que las políticas de accesibilidad hagan referencia a estas últimas, por los motivos mencionados en el párrafo anterior.

Las WCAG 1.0 y las WCAG 2.0 están organizadas y estructuradas de manera diferente tal como indica la Tabla 2.4.3.1.

WCAG	1.0 (1999)	2.0 (2008)
Aplicación	Exclusivamente tecnologías HTML y CSS	Diferentes tipos de tecnologías. Tecnologías avanzadas.
Organización	14 pautas con 65 puntos de verificación.	4 principios básicos con 12 pautas que contienen un total de 61 criterios de conformidad.
Base de evaluación	Puntos de verificación	Criterios de conformidad
Criterio de conformidad	- Prioridad 1, 2, 3 - Nivel A, AA, AAA	Nivel A, AA, AAA
Material de soporte	Breve descripción de cada pauta	i. Documento principal contiene las pautas y criterios de éxito independientes de la tecnología, sin descripción adicional. ii. Documentos adicionales: - Entendiendo WCAG 2.0 - Técnicas para WCAG 2.0

Tabla 2.4.3.1: Comparación de la estructura de las WCAG 1.0 y WCAG 2.0

Las pautas WCAG 2.0 están organizadas en cuatro (4) principios fundamentales para la accesibilidad del contenido. A su vez, estos principios tienen doce (12) pautas asociadas en total: los dos primeros tienen cuatro (4) pautas, el tercero tiene tres (3) y el último tiene una (1), las cuales no son evaluables en sí mismas, sino que proporcionan las metas básicas para hacer el contenido accesible, y sirven para comprender los criterios de conformidad y poder implementarlos.

Criterios de conformidad

Cada una de estas pautas, además, tiene asociados criterios de conformidad (61 en total) que sí son mensurables, deben cumplirse y tienen una prioridad asociada dependiendo de su impacto en la accesibilidad.

Para cumplir con WCAG 2.0, uno de los siguientes niveles de conformidad se cumple en su totalidad:

- **Nivel A:** Para la conformidad con el nivel A (el mínimo nivel de conformidad), la página web debe satisfacer todos los criterios de conformidad de nivel A.
- **Nivel AA:** Para la conformidad con el nivel AA, la página web debe satisfacer todos los criterios de conformidad de nivel A y nivel AA.
- **Nivel AAA:** Para la conformidad con el nivel AAA, la página web debe satisfacer todo el nivel A, AA y los criterios de conformidad del nivel AAA.

Como se puede observar, WCAG 2.0 elimina el concepto de Prioridad utilizado por WCAG 1.0, dejando cada conjunto de criterios de conformidad detallados directamente por nivel.

Principios fundamentales

A continuación se describen los cuatro (4) principios fundamentales y el conjunto de pautas asociado a cada uno de ellos.

i. Perceptibilidad

‘La información y los componentes de la interfaz de usuario deben ser presentados a los usuarios de modo que ellos puedan percibirlos’.

Las pautas incluidas en este principio especifican que deben proporcionarse diferentes alternativas a contenidos no textuales, manteniendo la información y estructura del contenido. Esto significa que la información debe presentarse de modo que pueda ser apreciada por cualquiera de los sentidos, es decir que si una persona carece de visión, por ejemplo, exista audio o texto alternativo que pueda ser interpretado por un lector de pantallas.

Pauta 1.1: Alternativas textuales

Proporcionar alternativas textuales para todo contenido no textual de modo que se pueda convertir a otros formatos que las personas necesiten, tales como textos ampliados, braille, voz,

símbolos o en un lenguaje más simple.

Pauta 1.2: Medios tempo-dependientes

Proporcionar alternativas sincronizadas para los medios tempo-dependientes.

Pauta 1.3: Adaptable

Crear contenido que pueda presentarse de diferentes formas (por ejemplo, con una disposición más simple) sin perder información o estructura.

Pauta 1.4: Distinguible

Facilitar a los usuarios ver y oír el contenido, incluyendo la separación entre el primer plano y el fondo.

ii. Operabilidad

‘Los componentes de la interfaz de usuario y la navegación deben ser operables’.

Estas pautas definen que toda la información disponible debe ser accesible mediante el teclado, y que la misma debe permanecer visible el tiempo necesario como para poder ser leída y/o utilizada por los usuarios.

Además especifican que deben proporcionarse medios que ayuden a los usuarios a navegar el sitio, localizar el contenido deseado y determinar el posicionamiento dentro del mismo; no se debe requerir un dispositivo en particular para la interacción con el usuario.

Pauta 2.1: Accesible por teclado

Proporcionar acceso a toda la funcionalidad mediante el teclado.

Pauta 2.2: Tiempo suficiente

Proporcionar a los usuarios el tiempo suficiente para leer y usar el contenido.

Pauta 2.3: Convulsiones

No diseñar contenido de un modo que se sepa podría provocar ataques, espasmos o convulsiones.

Pauta 2.4: Navegable

Proporcionar medios para ayudar a los usuarios a navegar, encontrar contenido y determinar dónde se encuentran.

iii. Comprensibilidad

‘La información y el manejo de la interfaz de usuario deben ser comprensibles’.

Las pautas pertenecientes a este principio establecen principalmente que el contenido de un sitio debe ser legible y comprensible, y que su apariencia y operabilidad deben ser predecibles para no confundir a los usuarios. Esto significa que el usuario debe entender la información presentada, como así también el manejo de la interfaz (el contenido o la operación no pueden estar más allá del entendimiento del usuario). Además se deben proporcionar ayudas para evitar que el usuario cometa errores o, en caso de hacerlo, poder revertirlo.

Pauta 3.1: Legible

Hacer que los contenidos textuales resulten legibles y comprensibles.

Pauta 3.2: Predecible

Hacer que las páginas web aparezcan y operen de manera predecible.

Pauta 3.3: Entrada de datos asistida

Ayudar a los usuarios a evitar y corregir los errores.

iv. Robustez

‘El contenido debe ser suficientemente robusto como para ser interpretado de forma fiable por una amplia variedad de aplicaciones de usuario, incluyendo tecnologías asistivas¹³’.

En este principio, se establece que los sitios deben ser compatibles al máximo con los navegadores y tecnologías actuales, y además deben seguir siéndolo a medida que estas evolucionen.

Pauta 4.1: Compatible

Maximizar la compatibilidad con las aplicaciones de usuario actuales y futuras, incluyendo las ayudas técnicas.

A continuación, en la figura 2.4.3.2, se muestra la estructura en que se organizan las pautas WCAG 2.0, junto con sus criterios de éxito.

¹³ Según la Norma UNE-EN ISO 9999:2007: "Cualquier producto, instrumento, equipo o sistema técnico utilizado por una persona minusválida, fabricado especialmente o disponible en el mercado para prevenir, compensar, mitigar o neutralizar la deficiencia, incapacidad o discapacidad"

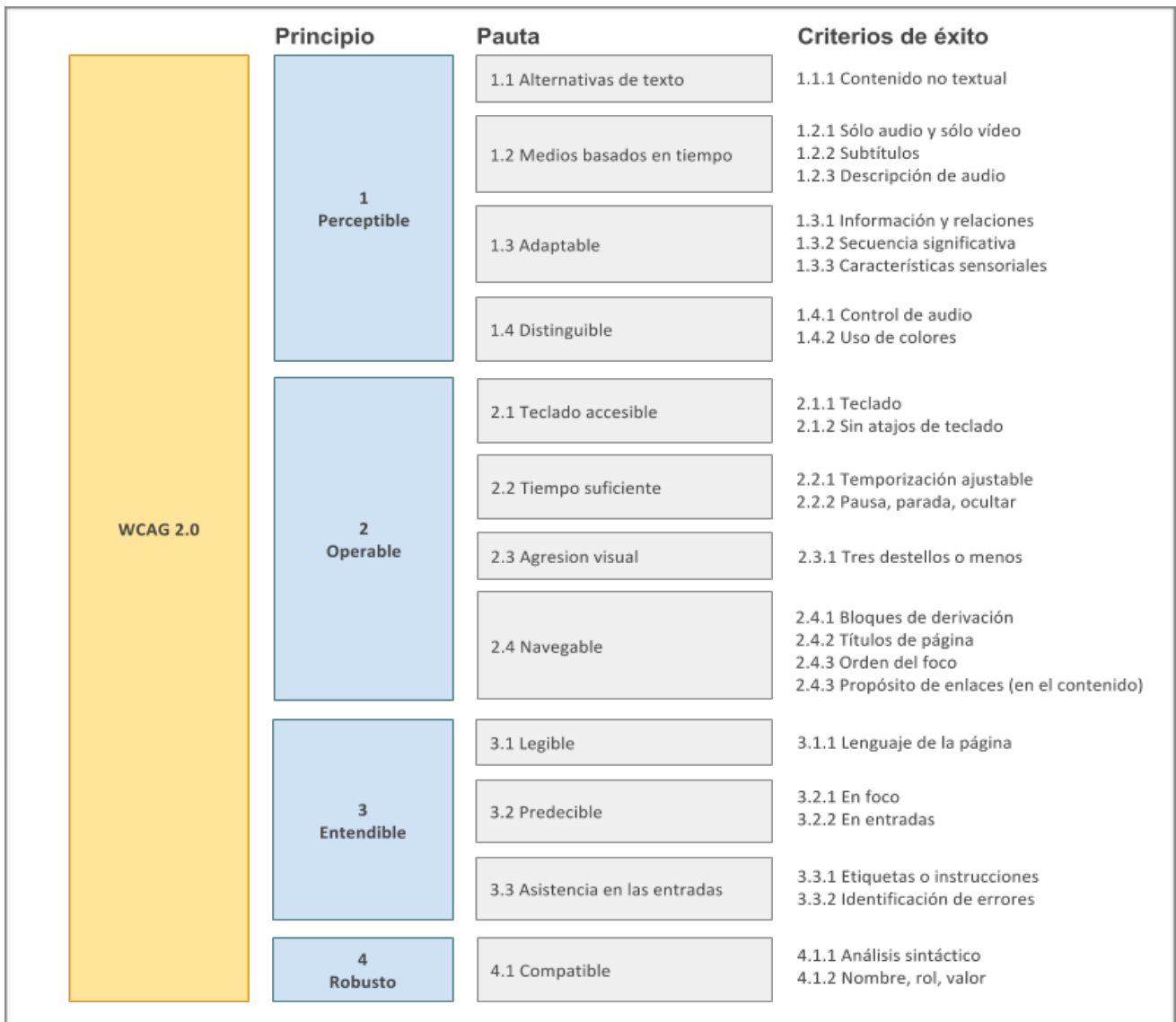


Figura 2.4.3.2: Estructura de las pautas WCAG 2.0.

Para poder alcanzar el cumplimiento de las WCAG 2.0 de una manera más guiada, la W3C expone una serie de técnicas [Michael Cooper et al., 2015] que facilitan la implementación de un sitio accesible. Entre ellas, se encuentran las siguientes:

- Técnicas generales.
- Técnicas para HTML y XHTML.
- Técnicas para CSS.
- Técnicas para scripting del lado cliente.
- Técnicas para scripting del lado del servidor.
- Técnicas para SMIL.
- Técnicas para texto plano.
- Técnicas para ARIA.
- Técnicas para el uso de Adobe Flash.

- Técnicas para el uso de Microsoft Silverlight.
- Técnicas para PDF.
- Técnicas para errores comunes.

Cable aclarar que estas técnicas no son obligatorias, y son meramente informativas. Las bases para el cumplimiento de las WCAG 2.0 están especificadas en el propio estándar.

2.4.4. Pautas UAAG

Las Pautas de Accesibilidad para Agentes de usuario (UAAG) explican cómo hacer accesible los agentes de usuario a las personas con discapacidad, principalmente para aumentar la accesibilidad del contenido web.

La UAAG 1.0 fue aprobada en diciembre de 2002 y es la versión estable y de referencia, y contiene un amplio conjunto de puntos de control entre los que se incluyen:

- Acceso a todo el contenido, incluido el contenido vinculado a eventos generados por el mouse o el teclado.
- Control del usuario sobre cómo es presentado el contenido.
- Control del usuario sobre la interfaz de usuario, con la documentación de las funciones de accesibilidad.
- Interfaces de programación estándar, para permitir la interacción con las ayudas técnicas.

Las pautas UAAG 2.0, que están en desarrollo actualmente, proveen las guías para el diseño de aplicaciones de usuario que reduzcan las barreras de accesibilidad web.

Entre estas aplicaciones se encuentran navegadores, reproductores multimedia y todas aquellas aplicaciones que recuperan y muestran el contenido web.

Los agentes de usuario que forman estas guías promueven la accesibilidad a través de su propia interfaz y su capacidad para comunicarse con otras tecnologías (especialmente las tecnologías de asistencia).

Además de ayudar a los desarrolladores de navegadores y reproductores multimedia, las UAAG 2.0 benefician a los desarrolladores de tecnologías de asistencia, ya que explican qué tipo de información y control puede esperar una tecnología de asistencia de un agente de usuario.

Las UAAG 2.0 son parte de una serie de pautas de accesibilidad publicadas por la Iniciativa de Accesibilidad Web de la W3C (WAI). [Shawn Lawton Henry et al., 2005]

2.4.5. Pautas ATAG

Las pautas ATAG (Authoring Tool Accessibility Guidelines) definen cómo las herramientas de autor deben ayudar a los desarrolladores web a producir contenido web que sea accesible, conforme a las pautas WCAG. Las ATAG también definen la forma de hacer accesibles las herramientas de autor, de forma que puedan ser usadas por personas con discapacidades. [Shawn Lawton Henry et al., 2013]

Las ATAG 1.0 fueron aprobadas en febrero de 2000 y es la versión estable y de referencia.

Las ATAG 2.0 están siendo desarrolladas para ser compatibles con las WCAG 2.0.

Estas pautas están dirigidas principalmente a desarrolladores de herramientas de autor, incluidos los siguientes tipos:

- Herramientas del tipo what-you-see-is-what-you-get (WYSIWYG)¹⁴ como editores de HTML.
- Software para creación de sitios web, por ejemplo, sistemas de gestión de contenidos (CMS), herramientas de cursos, etc.
- Herramientas de creación multimedia.
- Sitios web que permiten a los usuarios añadir contenidos, como blogs, wikis, sitios para compartir fotos y sitios de redes sociales, entre otras.

2.4.6. Estándares de la Sección 508

Poco después de que fue promulgada la ley 508, detallada en la sección 2.5.2, la Comisión de Acceso¹⁵ creó un comité asesor para preparar recomendaciones sobre los estándares que serían desarrollados.

Estos estándares definen los tipos de tecnología comprendidos y establecen provisiones que determinan un nivel mínimo de accesibilidad. En la sección *Aplicación* de esta ley (Sección 1194.2) se esboza el alcance y la extensión de los mismos. Estos estándares cubren un rango completo de tecnologías electrónicas e informáticas del sector Federal, incluidas aquellas usadas para comunicación, duplicación, cómputos, almacenamiento, presentación, control, transporte y producción. Esto incluye computadoras, software, redes, periféricos y otros tipos de equipos electrónicos de oficina.

De acuerdo con esta ley en la Sección 1194.3, los estándares eximen a los sistemas usados para mando militar, armamentos, inteligencia y actividades de encriptación (pero no sistemas comerciales y administrativos de rutina, usados con otros propósitos de defensa o por agencias o personal de defensa). Los estándares también eximen al equipamiento "back office", usado solo por personal de servicio para el mantenimiento, reparación o propósitos similares. [United States Access Board, 2000]

2.4.7. WAI-ARIA

La ARIA (Accessible Rich Internet Applications) es una especificación de la W3C que surge a partir de la aparición de tecnologías útiles para crear sitios web dinámicos y controles avanzados de interfaz desarrollados con AJAX¹⁶, HTML, Javascript y tecnologías relacionadas, que en algunos casos no incluyen información semántica sobre su comportamiento. [Shawn Lawton Henry et al., 2014]

¹⁴ Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final.

¹⁵ Agencia federal independiente dedicada a la accesibilidad de personas con discapacidades.

¹⁶ Asynchronous Javascript And XML, es una técnica para crear aplicaciones interactivas o RIA (Rich Internet Applications) que se ejecutan en el navegador mientras se mantiene la comunicación asíncrona con el servidor.

La utilización de este tipo de tecnologías, ofrece como resultado aplicaciones web más ricas e interactivas pero menos accesibles, ya que su propio funcionamiento puede interferir en el de los productos de apoyo.

El principal objetivo de ARIA es aportar semántica y otros metadatos al contenido HTML, proporcionando métodos para describir roles, estados y propiedades para que éstos sean reconocibles y usables por los usuarios de tecnologías asistivas. Por ejemplo, con WAI-ARIA es posible identificar a una lista de enlaces como un menú de navegación y definir cuándo está expandido o colapsado. Otro ejemplo es el atributo **tabindex**, que es utilizado para reemplazar la navegación mediante un teclado, y hace posible navegar por los distintos menús y/o acciones mediante la tecla tab.

Para cumplir su objetivo, ARIA proporciona estructuras más semánticas para zonas funcionales, mejoras en la navegación mediante el teclado, controles complejos (widgets) más accesibles, y accesibilidad para contenidos actualizados dinámicamente.

Por lo tanto, se puede concluir que ARIA no funciona como una tecnología restrictiva o exclusiva, sino que se trata de una herramienta complementaria que permite hacer accesibles las aplicaciones web enriquecidas.

2.4.8. Estándar ISO/IEC 40500:2012

El estándar ISO/IEC 40500 [ISO/IEC 40500, 2012] convirtió a las WCAG 2.0 en un estándar internacional, lo que aumenta las posibilidades de adopción de la tecnología y las directrices de la W3C, principalmente por aquellos países en los cuales las políticas adoptadas a nivel nacional exigen que las normas técnicas deban ser desarrolladas por el ISO/IEC.

2.4.9. Metodología Unificada de Evaluación Web (UWEM):2006

La UWEM surge en el año 2006 como resultado de la unión de tres proyectos europeos en un grupo llamado WAB Cluster. Veintitrés (23) organizaciones europeas participan en estos tres proyectos cuyos nombres son EIAO (para la creación de un observatorio de la Web), Support EAM (Supporting the creation of an eAccessibility Mark, un proyecto de certificación) y BenToWeb (para el desarrollo de herramientas de evaluación).

Esta metodología fue desarrollada con el fin de garantizar que las herramientas y los métodos de evaluación definidos para la evaluación global de la Web, o la evaluación por separado de cada sitio, sean compatibles y coherentes entre sí y con la WAI. Su meta es incrementar el valor de las evaluaciones basándose en una interpretación común de las WCAG 1.0. La última versión es la UWEM 2.0 que consiste en la adaptación de las UWEM 1.0 a las WCAG 2.0. [Web Accessibility Benchmarking Cluster, 2012]

2.5. Marco legal y normativas vigentes

Como se mencionó anteriormente, la accesibilidad indica la facilidad con la que algo puede ser usado, visitado o accedido en general por todas las personas, especialmente por aquellas que poseen algún tipo de discapacidad. En otras palabras, se trata de una condición que deben cumplir los entornos, productos y servicios para que sean comprensibles, utilizables y accesibles por todas las personas.

En el ámbito de Internet, gracias a las normas de accesibilidad de los contenidos web que se fueron publicando desde los años noventa, especialmente las pautas WCAG y la legislación derivada de las mismas, se redujeron significativamente las barreras tecnológicas de la red, logrando cada vez mayor concienciación social y legislativa, exigiendo el cumplimiento de ciertos requisitos mínimos. [INTECO, 2008]

Muchos países cuentan con leyes de cumplimiento obligatorio que permiten velar por la accesibilidad de las páginas de su administración pública, o bien de sitios de empresas que ofrecen servicios y productos para ella. A continuación se verán las más importantes, destacando la ley 26.653 sancionada en nuestro país el 3 de noviembre de 2010 y promulgada de hecho el 26 de noviembre de 2010.

2.5.1. Ley Nacional de Accesibilidad 26.653

La “Ley de Accesibilidad de la Información en las Páginas Web” [Ley 26.653, 2010] es un primer paso hacia la adecuación de la legislación argentina a la Convención sobre los Derechos de las Personas con Discapacidad, aprobada por la Ley 25.280 del año 2008. En su artículo 4º, la Convención compromete a los Estados parte a adoptar todas las medidas legislativas, administrativas y de otra índole que sean pertinentes para hacer efectivos los derechos que en ella se reconocen, y en su artículo 9º señala que a fin de que las personas con discapacidad puedan vivir en forma independiente y participar plenamente en todos los aspectos de la vida, los Estados parte adoptarán las medidas pertinentes para asegurar el acceso por parte de personas con discapacidad, en igualdad de condiciones con las demás, al entorno físico, al transporte, a la información y a las comunicaciones, incluyendo el acceso a los sistemas y las tecnologías de la información, y a las comunicaciones.

Además de ser un avance en el resguardo de los derechos de las personas con discapacidad, la Ley de Accesibilidad Web contribuye a reducir la brecha digital. En este sentido, la Ley se articula de un modo virtuoso con las políticas públicas de inclusión digital que está llevando adelante el Estado Nacional.

La ley entiende por accesibilidad a *“la posibilidad de que la información de la página web, pueda ser comprendida y consultada por personas con discapacidad y por usuarios que posean diversas configuraciones en su equipamiento o en sus programas”* [Ley 26.653, artículo 3º] . Las normas y los requisitos de accesibilidad serán determinados y actualizados regularmente por la Oficina Nacional de Tecnologías de la Información (ONTI).

Además establece que el Estado Nacional deberá cumplir con los requisitos de accesibilidad para

personas con discapacidad a la hora de efectuar compras y/o contrataciones de servicios tecnológicos en materia informática. Por otro lado, se prevé brindar asistencia técnica directa, capacitación y formación de personal a las instituciones u organizaciones de la sociedad civil que no cuenten con los recursos necesarios para cumplir con los requerimientos de la ley. Asimismo, se deben difundir las normativas y requisitos de accesibilidad a las instituciones privadas para que éstas las cumplimenten en el diseño de sus sitios web.

El plazo máximo establecido por la ley para la implementación de los requisitos de accesibilidad es de veinticuatro (24) meses para las páginas ya existentes a la fecha de entrada de vigencia de la ley, y de doce (12) meses para los sitios web en proceso de elaboración, priorizando aquellas que presten servicios públicos e informativos.

En Argentina, para auditar lo expuesto, existe el Centro de Referencia en Accesibilidad Web (CRAW), que fue creado con el objetivo de permitir el acceso a las páginas web del Estado Nacional al mayor número de personas, independientemente de sus propias limitaciones o de las derivadas de su entorno.

2.5.2. Normas internacionales

AENOR

La Asociación Española de Normalización y Certificación [AENOR, 2010] es una entidad dedicada al desarrollo de normas y certificaciones en todos los sectores industriales y de servicios.

Fue diseñada por la Orden del Ministerio de Industria y Energía, el 26 de febrero de 1986, de acuerdo con el Real Decreto 1614/1985 y reconocida como organismo de normalización y como entidad de certificación por el Real Decreto 2200/1995, en desarrollo de la Ley 21/1992, de Industria.

AENOR desarrolló la norma UNE 139802:1998 EX (experimental), en el año 1998, considerada como la primer norma de accesibilidad a nivel mundial. Luego, en el año 2004, publicó la versión actualizada de la norma anterior, UNE 139803:2004, basada en las pautas WCAG 1.0 y finalmente su sustituta UNE 139803:2012, compatible en su totalidad con las pautas WCAG 2.0.

Norma UNE 139803:2004

La norma UNE 139803:2004 publicada en diciembre de 2004, contempla las especificaciones que han de cumplir los contenidos web para que puedan ser accesibles. Esta norma es equivalente a las “Pautas de Accesibilidad al Contenido en la Web” (WCAG 1.0), pero estructuradas de forma diferente. Existe una correspondencia directa entre los requisitos de esta norma y los puntos de verificación de las pautas WCAG 1.0 de la WAI. [UNE 139803, 2004]

Norma UNE 139803:2012

Luego del desarrollo de las pautas WCAG 2.0 surgió la necesidad de actualizar el contenido de la norma UNE 139803:2004 para que sus requisitos sean acordes con el contenido de las WCAG 2.0. Esta norma señala directamente qué partes de WCAG 2.0 se consideran requisitos y con qué nivel

de prioridad. [UNE 139803, 2012]

Ley Española 56/2007

En la Ley Española 56/2007 del 28 de diciembre de 2007, de “*Medidas de Impulso de la Sociedad de la Información*”, se introducen una serie de cambios en materia de facturación electrónica y de refuerzo de los derechos de los usuarios y se hacen las modificaciones necesarias en el ordenamiento jurídico para promover el impulso de la sociedad de la información. [Ley Española 56/2007, 2007]

En esta ley se expresa lo siguiente:

“Las páginas de Internet de las empresas de especial trascendencia económica deberán satisfacer como mínimo el nivel medio de los criterios de accesibilidad generalmente reconocidos. Las empresas consideradas de especial trascendencia económica son aquellas que agrupen a más de cien trabajadores y que operen en los siguientes sectores económicos:

- i. Sectores de comunicaciones electrónicas a consumidores.*
- ii. Servicios financieros destinados a consumidores, donde se incluyen:*
 - i. Servicios bancarios, de crédito o de pago.*
 - ii. Servicios de inversión.*
 - iii. Operaciones de seguros privados.*
 - iv. Planes de pensiones.*
 - v. Actividad de corredor de seguros.*
- iii. Servicios de suministro de agua a consumidores.*
- iv. Servicios de suministro de gas al por menor.*
- v. Servicios de suministro eléctrico a consumidores finales.*
- vi. Servicios de agencias de viajes.*
- vii. Servicios de transporte de viajeros (ya sea por carretera, ferrocarril, por vía marítima, o por vía aérea).*
- viii. Actividades de comercio al por menor.*

Como conclusión, se indica que a partir de esta ley, cumplir con la accesibilidad web es una obligación. Su no cumplimiento es un delito, con las consecuencias judiciales, e incluso sociales que ello supone para la imagen de cualquier empresa.” [Ley Española 56/2007 Disposición adicional 5ta, apartado cinco]

Sección 508 - Estados Unidos de América

La legislación mencionada como "Sección 508" es un componente de la Ley de Rehabilitación (29 US Code, 794d - Electronic and information technology) enmendada por la Ley Pública 106-246 publicada en julio del año 2000 [Ley 106–246, 2000]. La enmienda fue firmada como ley por el Presidente Bill Clinton el 7 de agosto de 1998.

La Sección 508 fue decretada con el fin de eliminar las barreras de la tecnología de la información, haciendo disponibles nuevas oportunidades para las personas con discapacidad, y para fomentar el desarrollo de tecnologías que ayuden a alcanzar estos objetivos. La ley se aplica a todas las agencias federales que desarrollan, adquieren, mantienen o utilizan tecnologías electrónicas e informáticas. Bajo la Sección 508, los empleados federales y los miembros del público con discapacidades deben tener el mismo acceso y uso de la información y datos que los empleados federales que no son personas con discapacidad.

Hay una exhaustiva comparación [Jim Thatcher, 2007] entre el apartado de la Sección 508 dedicado a la Web y los puntos de verificación de prioridad 1 de las WCAG 1.0, pero si se satisfacen los puntos de verificación de prioridad 1 y 2 (lo que se le exige a la administración pública europea) se cumple a la vez con los requerimientos de la Sección 508.

Real Decreto 1494/2007

Extiende la Ley Española 56/2007 y obliga a que cumplan con las normas de accesibilidad no sólo las páginas de la administración pública, sino las de otras muchas empresas. Vigente actualmente, aprueba el Reglamento sobre las Condiciones Básicas para el Acceso de las Personas con Discapacidad a las Tecnologías, Productos y Servicios relacionados con la Sociedad de la Información y Medios de Comunicación social.

Esta ley, en el Capítulo III - Artículo 5, expresa:

“De igual modo, serán exigibles, y en los mismos plazos, estos criterios de accesibilidad para las páginas de Internet de entidades y empresas que se encarguen, ya sea por vía concesional o a través de otra vía contractual, de gestionar servicios públicos, en especial, de los que tengan carácter educativo, sanitario y servicios sociales.” [Real Decreto 1494/2007, 2007]

Por lo tanto, que las empresas cumplan con las normas de accesibilidad pasa a ser una obligación, especialmente para aquellas empresas que participen o quieran participar en contratos con la Administración o Financiación pública.

La intención es que poco a poco todas las empresas adapten sus contenidos web para que sean accesibles. No sólo las empresas que se consideran en el Real Decreto 1494/2007 deben aplicar las normas de accesibilidad. La tendencia futura es que todas las páginas de Internet cumplan con los criterios de accesibilidad. Esta información implica que el mundo empresarial tenga especial cuidado en adaptar los contenido de sus páginas web lo antes posible.

Este decreto reconoce que las WCAG son el documento de referencia de la accesibilidad web y permite validar de acuerdo a normas distintas siempre y cuando permitan alcanzar un nivel de accesibilidad similar al que garantiza la Norma UNE 139803, y éste es un requisito que evidentemente cumplen las WCAG 2.0.

Norma Técnica Colombiana 5854

El 15 de junio de 2011, el Instituto Colombiano de Normas Técnicas y Certificación (ICONTEC) ratificó la Norma Técnica 5854 que busca establecer los requisitos de accesibilidad que deben implementar las páginas web en Colombia. Esta norma es equivalente a las WCAG 2.0 [Norma técnica 5854, 2011].

Norma CWA 15554:2006

La Norma CWA 15554 es una norma europea avalada por el Comité Europeo de Normalización (CEN) establecida para obtener un acuerdo en un esquema de evaluación de la conformidad relativa a la entrega de una "marca de calidad" para la accesibilidad web, de conformidad con las Directrices de Accesibilidad para el Contenido Web de la W3C. Esta norma constituye la base de la certificación europea en accesibilidad web. [CWA 15554, 2006]

2.6 Conclusión

En base al estudio de los estándares de accesibilidad anteriormente mencionados, se concluye que los más importantes y con mayor aceptación son las pautas WCAG 1.0 y WCAG 2.0 por ser la base de la mayoría de las normas y estándares vigentes, adoptados por países que se rigen por determinadas normas como son las normas ISO; motivo por el cual el análisis intensivo de accesibilidad realizado por SiMor tiene sus bases en dichas pautas, puntualmente en las WCAG 2.0.

Por otra parte se concluye que, si bien el desarrollo de estándares de accesibilidad es un paso muy importante en la evolución de la sociedad en este aspecto, es necesario e indispensable el desarrollo de normas y leyes que exijan la adopción y el cumplimiento de los mismos; por eso la importancia de las normativas anteriormente mencionadas.

Capítulo 3: Validadores de accesibilidad

Hasta ahora se definió el panorama actual en relación a la accesibilidad de los contenidos disponibles en la Web, desde su importancia a nivel social y los beneficios de su aplicación, hasta las normas y estándares que conforman su marco de referencia.

A lo largo de este capítulo se explica el objetivo de los validadores de accesibilidad, se analizan distintas características que permiten su clasificación y se describen los más populares mediante una comparación de los resultados de evaluación de un sitio web de referencia.

3.1. Introducción

Con la creciente necesidad de las organizaciones, las entidades educativas y los negocios que ofrecen servicios en línea accesibles a todos los individuos, muchos desarrolladores de sitios web recurren a herramientas "automáticas" al momento de llevar a cabo la tarea de construcción de sitios web accesibles.

Estas herramientas automáticas de evaluación son "programas de software o servicios en línea que ayudan a determinar si el contenido web cumple con las normas de accesibilidad" [Eric Eggert and Shadi Abou-Zahra - W3C, 2014], y su utilización puede reducir significativamente el tiempo y el esfuerzo para evaluar sitios web.

3.2. Clasificación de validadores

En la actualidad existe una gran variedad de herramientas, que pueden clasificarse, según el artículo Accessibility Evaluation Tools publicado el 27 de Agosto del 2013 en el sitio <http://webaim.org> [WebAIM, 2013] por los siguientes criterios:

Normas y/o estándares utilizados

Una de las claves en la comprensión de las herramientas de accesibilidad es entender las normas bajo las que se desarrolló la herramienta.

Actualmente hay dos estándares que son los más utilizados, el WCAG 2.0 y la Sección 508 de la Ley de Rehabilitación de EE.UU.

Plataforma

Algunas herramientas están disponibles como sitios web, otras como extensiones que permiten añadir al navegador funciones de validación, o herramientas que forman parte de otra herramienta de diseño web.

Por último, están aquellas que requieren ser instaladas como aplicación en el sistema operativo del usuario.

Alcance

Estas herramientas también pueden clasificarse en función de su capacidad de validación o su ámbito de aplicación.

Algunas herramientas son simples y limitadas en su alcance por lo que sólo evalúan una página a la vez y no realizan un análisis completo.

Otras herramientas son más específicas en la evaluación y se centran en sólo un elemento de un sitio web, por ejemplo, en cómo percibe el sitio una persona ciega o con problemas de visión, y están aquellas herramientas que son capaces de examinar sitios complejos y su evaluación de errores es más completa.

Evaluación y corrección

Otra clasificación importante se refiere a la capacidad de corrección implementada por estas herramientas. Algunas sólo pueden realizar una evaluación del código, pero otras son capaces de realizar la evaluación y además guiar al usuario (en este caso el programador y/o diseñador web) en el proceso de corrección de errores.

Estilos de informe generados

Las herramientas de accesibilidad generan una gran variedad de informes basados en los resultados de los sitios web analizados. Existen diferentes tipos de informes:

- **Basado en texto:** Son el tipo más común de informe de accesibilidad. Estos informes generalmente listan la directriz específica que se utiliza para escanear la página y las instancias de cada tipo de error de accesibilidad. Algunos también muestran la porción de código fuente de la página en la que se produce el error (figura 3.2.1).
- **Basado en íconos gráficos:** Este estilo de informe utiliza íconos especiales para identificar los errores de accesibilidad y problemas de verificación manuales en una página web. Con este tipo de informe, los íconos están integrados en la interfaz gráfica de usuario de la página web junto al elemento de la página afectado por un problema de accesibilidad (figura 3.2.2).
- **Informe EARL:** El formato EARL (Lenguaje de evaluación y reporte) [Shadi Abou-Zahra and W3C/WAI, 2011] fue creado por la W3C con el objetivo de facilitar el intercambio de resultados de las pruebas realizadas por herramientas de validación, con un formato independiente de la herramienta que lo produce y la plataforma.
 - El informe EARL, a diferencia del resto de informes que son formato HTML, es un archivo de texto con formato XML.
 - Este tipo de informe dispone de un grupo de etiquetas para identificar la herramienta, la persona evaluadora, el contenido evaluado y además una serie de declaraciones que contiene el resultado de realizar una evaluación (figura 3.2.3).









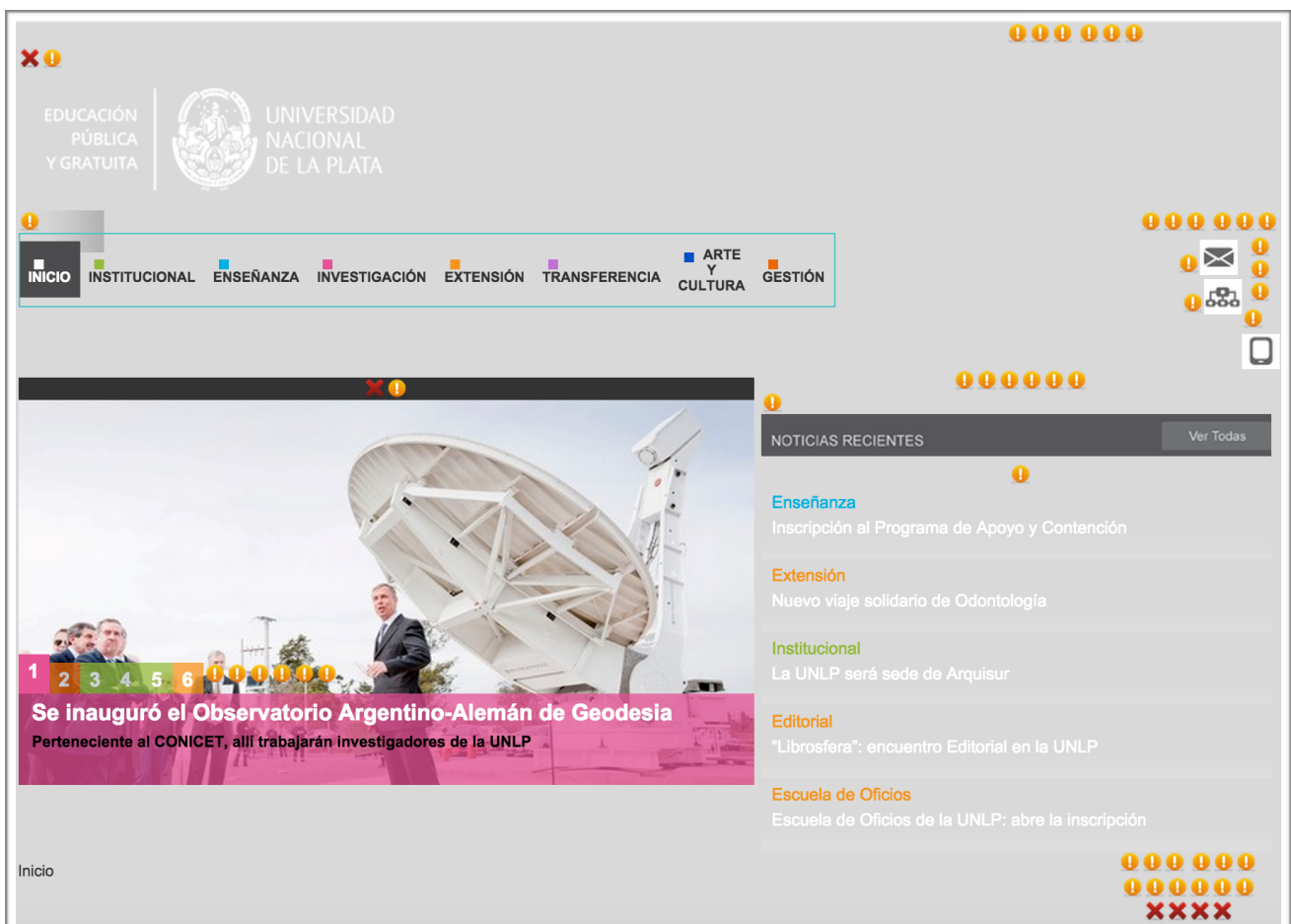
Typology	Comprobation	Technique	Result Problems	Line numbers
1.1.1 - Non-text Content				
Images	Images that may require a long description 	[H45]	 29	99, 100, 106, 189, 312, 316, 320, 322, ... →
	Images with empty "alt" attribute 	[H67]	 20	80, 174, 354, 356, 358, 362, 364, 366, ... →
Forms	Form controls without label 	[H44, H65]	 1	285
Objets	Links to audio files that may need transcriptions 	[H53]	 1	688

Figura 3.2.1: Informe basado en texto.



The screenshot shows a website for the Universidad Nacional de La Plata (UNLP). The header features the university's logo and navigation links: INICIO, INSTITUCIONAL, ENSEÑANZA, INVESTIGACIÓN, EXTENSIÓN, TRANSFERENCIA, ARTE Y CULTURA, and GESTIÓN. A main news article is titled "Se inauguró el Observatorio Argentino-Alemán de Geodesia" and mentions the CONICET. A sidebar on the right lists "NOTICIAS RECIENTES" with categories like Enseñanza, Extensión, Institucional, Editorial, and Escuela de Oficios. The page is overlaid with several accessibility icons: a red 'X' with an exclamation mark in the top left, a row of six yellow warning icons in the top right, a row of six yellow warning icons above the news sidebar, a red 'X' with an exclamation mark above the main article image, a row of six yellow warning icons above the sidebar, and a row of six yellow warning icons and four red 'X' with exclamation marks in the bottom right corner. A horizontal bar with numbers 1 through 6 is visible over the main article image.

Figura 3.2.2: Informe basado íconos gráficos.

```

<earl:TestResult rdf:about="#result">
  <earl:outcome rdf:resource="http://www.w3.org/ns/earl#failed"/>
  <dct:title xml:lang="en">Invalid Markup (code #353)</dct:title>
  <dct:description rdf:parseType="Literal" xml:lang="en">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>The <code>table</code> element is not allowed to appear
        inside a <code>p</code> element</p>
    </div>
  </dct:description>
  <earl:pointer rdf:resource="#pointer"/>
  <earl:info rdf:parseType="Literal" xml:lang="en">
    <div xmlns="http://www.w3.org/1999/xhtml">
      <p>It seems the <code>p</code> element has not been closed</p>
    </div>
  </earl:info>
</earl:TestResult>

```

Figura 3.2.3: Informe EARL.

Sea cual fuere el tipo y modelo de informe, éste debe ser lo suficientemente claro y expresivo para que le sea de utilidad al programador para poder corregir con seguridad lo necesario y así cumplimentar con la accesibilidad requerida.

Licencia

Podemos encontrar básicamente cuatro (4) tipos de licencia, aplicables a cualquier software: privativas, semi-libres, libres recíprocas y libres. Las privativas suelen ser usadas para fines comerciales debido a que son aquellas que no permiten modificar nada del código, limitándolo sólo a su uso, y bajo ciertas condiciones; como por ejemplo limitar el uso sólo a fines educativos. Las semi-libres contienen ciertas restricciones, como por ejemplo la imposibilidad de crear obras derivadas pero permitiendo la libre distribución de copias del original. Libres recíprocas son aquellas que obligan a que las obras derivadas queden también libres. Y las libres son las que permiten incluso licenciar las obras derivadas como privativas y utilizarlas con fines comerciales.

3.3. Características relevadas a profesionales del área

Como desarrolladores y usuarios de estas tecnologías, se conocen aspectos que pueden ser mejorados o anexados a las herramientas de validación para que la tarea de generar contenido web accesible resulte menos tediosa, compleja y costosa.

Con el fin de conocer las opiniones de personas familiarizadas con este tipo de herramientas se confeccionó, a través de Google Forms, una encuesta (incluida en el Anexo de este informe) bilingüe (español e inglés) publicada en el mes de Julio del 2014 en sitios y listas de mails relacionados a la accesibilidad web como lo son: <http://foros.discapnet.es/> y <http://www.accessifyforum.com>, lista de mails de <http://list.webaim.org/>, y en <https://>

www.facebook.com/groups/sidar/?fref=ts, página de la red social Facebook, perteneciente a la fundación SIDAR.

Los datos recopilados en dicha encuesta indican qué aspectos resultan importantes a la hora de analizar una herramienta de validación, qué características de los validadores disponibles en la actualidad no resultan completamente adecuadas o cómodas, y cuáles son aquellas funcionalidades que resultan imprescindibles.

A continuación, en la Figura 3.3.1, se muestran los resultados relacionados a las opiniones registradas en base a experiencias en el uso de diferentes validadores web.

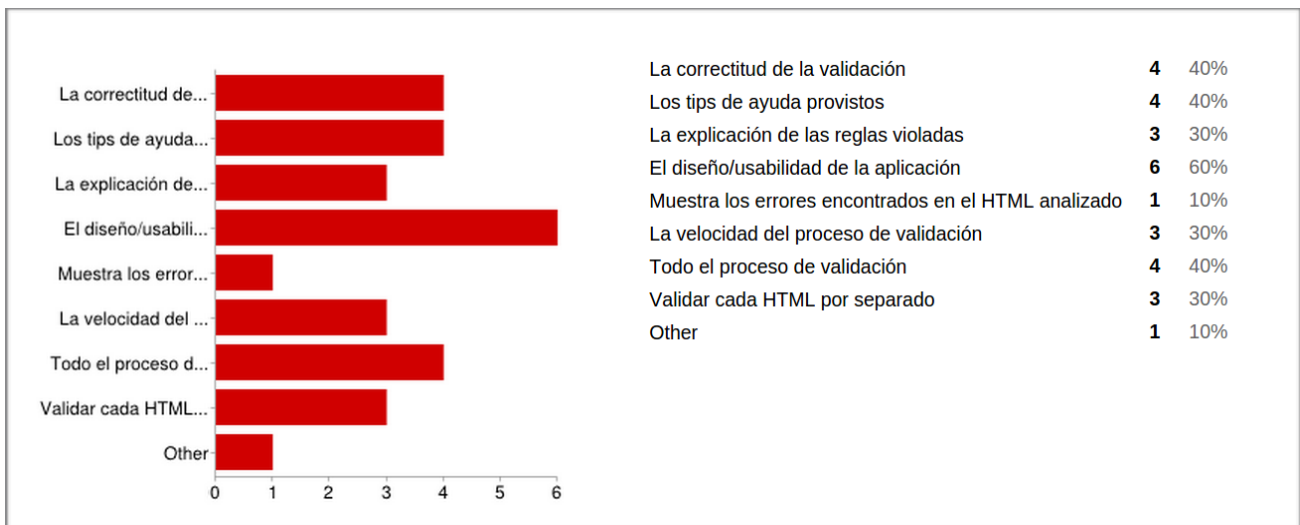


Figura 3.3.1: Características negativas encontradas

Además, en la Figura 3.3.2, se exponen las expectativas que los usuarios tienen para con las herramientas de este tipo (la encuesta completa puede encontrarse en el anexo de este informe).

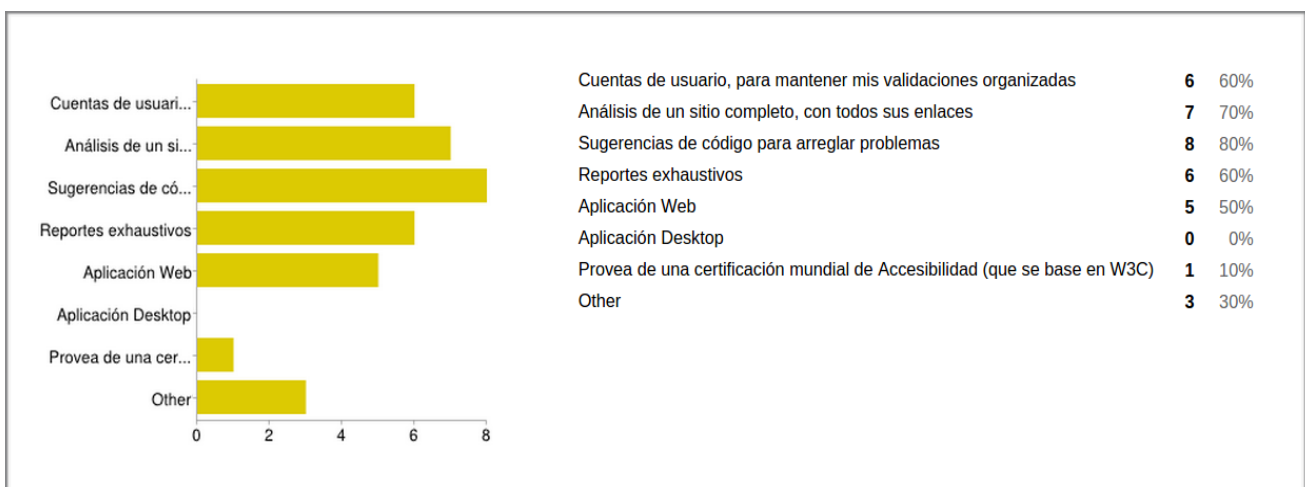


Figura 3.3.2: Características sugeridas

3.4. Evaluación y revisión

Durante el desarrollo de esta tesina se investigaron más de veinte (20) validadores de accesibilidad web de diversas características, muchos de ellos publicados por la W3C en Web Accessibility Evaluation Tools List [Eric Eggert and Shadi Abou-Zahra - W3C, 2014], y otros que fueron utilizados y recomendados por los gobiernos de algunas naciones como es el caso del software ASES.

Teniendo en mente las características mencionadas anteriormente y la experiencia como desarrolladores web de los tesisistas, varias de estas herramientas fueron descartadas en una primer instancia debido a deficiencias de funcionamiento encontradas durante su utilización. En muchos casos, no fue posible su uso debido a errores críticos en la aplicación al momento de escribir esta tesina.

Otras características importantes tenidas en cuenta a la hora de analizar las diferentes herramientas fue su capacidad de correr sobre múltiples plataformas y la posibilidad de evaluar sitios completos, a fin de recabar los aspectos positivos de cada uno de estos a ser incluidos en SiMor, y los negativos para solucionarlos.

Se decidió además, contemplar sólo aquellos que validarán contra las WCAG y fueran validadores web; características que permitieron realizar una comparación razonable con la herramienta propuesta.

Dado que la idea fue siempre generar una herramienta de código abierto que cubra las características deseadas por los desarrolladores web, la lista se concentró además en aquellas herramientas que podrían ser extensibles. En esta búsqueda sólo quedó una única herramienta, por lo que se decidió analizar algunas con otro tipo de licencia a fin de recuperar algunas otras funcionalidades incluidas en estas herramientas.

Luego de este primer análisis, las herramientas seleccionadas para realizar un estudio más exhaustivo fueron las siguientes:

1. AChecker
2. FAE
3. WAVE
4. TAW

A continuación se realiza una comparación entre estos validadores, en la cual se analizan las características particulares de cada uno en función de las mencionadas en la sección 3.2 y se analizan los resultados obtenidos al evaluar un caso testigo en todos ellos. En este caso se tomó el sitio de la Universidad Nacional de La Plata (www.unlp.edu.ar) en su versión correspondiente al año 2014.

3.4.1. Características de los validadores evaluados

AChecker - Web Accessibility Checker

AChecker [Greg Gay et al., 2011] es una aplicación web con licencia GPL que permite evaluar problemas de accesibilidad en contenido HTML mediante el ingreso de la url de la página, la carga de un archivo HTML, o la introducción directa del código fuente HTML. Además, este validador brinda sugerencias de cómo solucionar el problema encontrado.

Permite la validación contra las WCAG 1.0, WCAG 2.0 y Sección 508 entre las más importantes.

La figura 3.4.1.1 ilustra el resultado de una validación de una página del sitio de la UNLP, mediante aChecker.

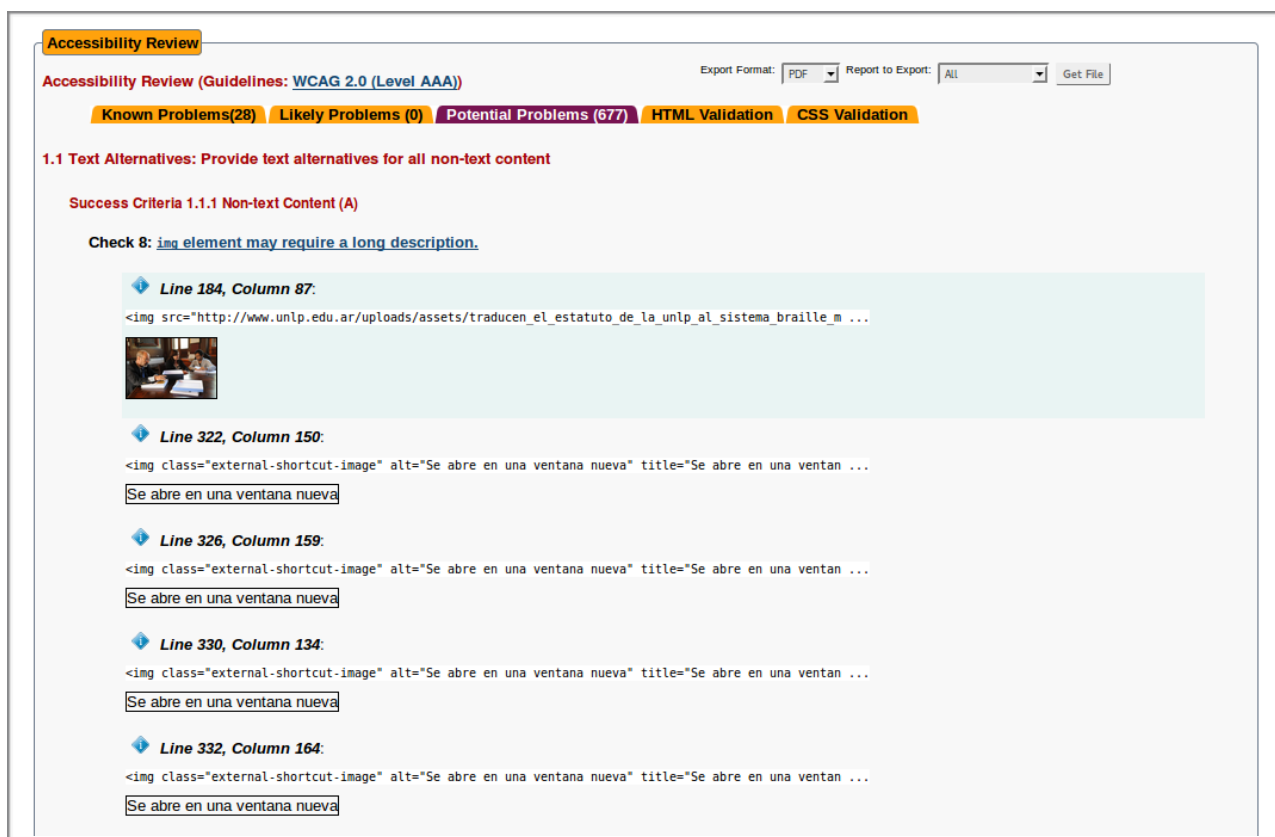


Figura 3.4.1.1: Informe HTML de aChecker.

Esta herramienta se encuentra disponible en idioma inglés, alemán e italiano.

Por otra parte, este validador identifica tres (3) tipos de problemas en lo que refiere al incumplimiento de las pautas de accesibilidad:

- **Problemas conocidos:** Estos son problemas que se han identificado con certeza como barreras de accesibilidad. Se debe modificar el sitio para solucionar estos problemas.
- **Posibles problemas:** Se trata de problemas que han sido identificados como barreras probables, pero requieren de un ser humano para tomar una decisión. Es probable que tengan que realizarse modificaciones en el sitio evaluado para arreglar este tipo de problemas.

- **Problemas potenciales:** Estos son problemas que aChecker no puede identificar, que requieren una decisión humana. Es posible que tenga que modificarse el sitio para resolver estos problemas, pero en muchos casos no es necesario.

Los reportes emitidos por este validador cuentan con la siguiente información:

- Cantidad de problemas encontrados.
- Criterios para los que se encontraron problemas.
- Porción de código que no cumple con la evaluación, junto con el número de línea y columna donde se encontró el problema.

Permite la exportación de los resultados en 4 formatos:

- PDF
- CSV
- EARL
- HTML

En la figura 3.4.1.2 se muestra una captura de aChecker en un dispositivo móvil (iPhone 6). Como puede observarse, aChecker no dispone de una interfaz adaptable al dispositivo, lo cual decrementa su nivel de accesibilidad.



Figura 3.4.1.2: aChecker visto desde un iPhone 6.

F AE - Functional Accessibility Evaluator 1.1

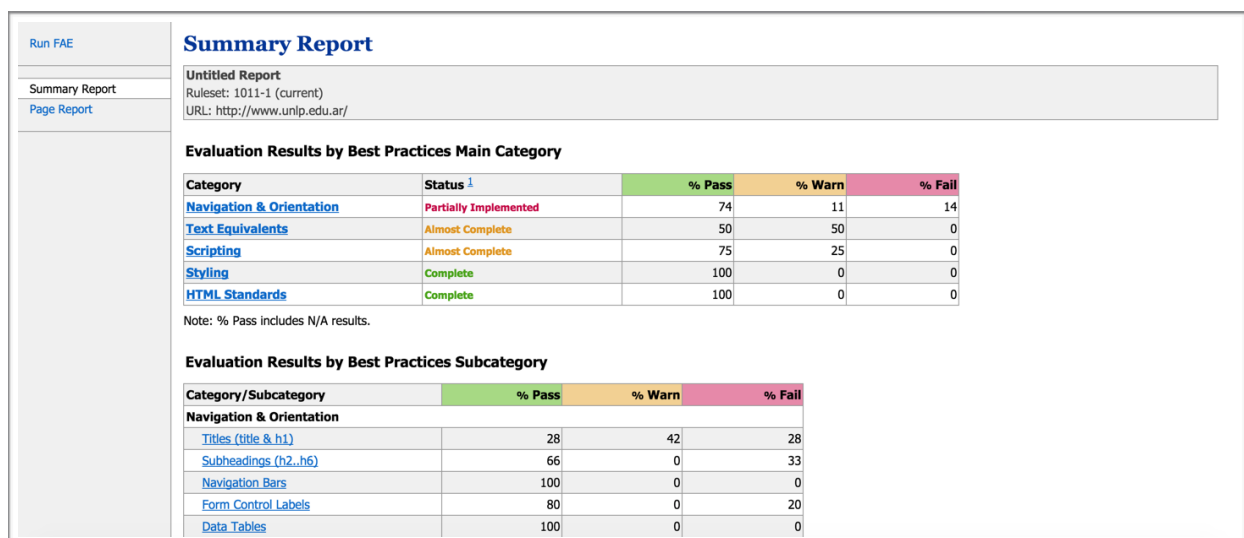
El validador FAE [Open Accessibility Alliance y OpenAjax Accessibility Task Force, 2014] es una aplicación web que analiza el código HTML de los enlaces de un sitio web contra el estándar ICITA HTML.

Las pautas ICITA HTML no son un estándar nuevo, sino más bien una declaración de técnicas para la aplicación de la WCAG 1.0, la Sección 508 e Illinois Information Technology Accessibility Act (IITAA). Esta herramienta no dispone de sugerencias de corrección a los errores encontrados. La aplicación, al no brindar un acceso al código fuente ni ningún enlace de descarga, no puede ser utilizada en este proyecto para ser extendida.

Este validador utiliza la siguiente clasificación en su análisis de accesibilidad:

- **Violación:** Uno o más elementos no cumplen los requisitos de la regla evaluada.
- **Advertencia:** Uno o más elementos no cumplen los requisitos recomendados por la regla evaluada.
- **Chequeo manual:** No se puede determinar automáticamente el resultado. La evaluación requiere intervención humana.
- **Aprobada:** Se cumplen todos los requerimientos de la regla.
- **Ocultos:** Indica los elementos que no fueron evaluados por estar ocultos (ya sea visualmente o por las ayudas técnicas).
- **No aplicable:** La regla no se aplicó porque no existían elementos objetivos pertinentes.

La figura 3.4.1.3 ilustra el resultado de una validación de una página del sitio de la UNLP, mediante FAE.



The screenshot shows the 'Summary Report' interface of the FAE tool. It includes a navigation sidebar on the left with options like 'Run FAE', 'Summary Report', and 'Page Report'. The main content area displays the report title 'Untitled Report', the ruleset '1011-1 (current)', and the URL 'http://www.unlp.edu.ar/'. Below this, there are two tables: 'Evaluation Results by Best Practices Main Category' and 'Evaluation Results by Best Practices Subcategory'. The first table has columns for Category, Status, % Pass, % Warn, and % Fail. The second table has columns for Category/Subcategory, % Pass, % Warn, and % Fail.

Category	Status	% Pass	% Warn	% Fail
Navigation & Orientation	Partially Implemented	74	11	14
Text Equivalents	Almost Complete	50	50	0
Scripting	Almost Complete	75	25	0
Styling	Complete	100	0	0
HTML Standards	Complete	100	0	0

Note: % Pass includes N/A results.

Category/Subcategory	% Pass	% Warn	% Fail
Navigation & Orientation			
Titles (title & h1)	28	42	28
Subheadings (h2..h6)	66	0	33
Navigation Bars	100	0	0
Form Control Labels	80	0	20
Data Tables	100	0	0

Figura 3.4.1.3: Informe HTML de FAE.

F AE se encuentra disponible sólo en idioma inglés.

En la figura 3.4.1.4 se muestra una captura de FAE en un dispositivo móvil (iPhone 6).

Como puede observarse, FAE no dispone de una interfaz adaptable al dispositivo, lo cual decrementa su nivel de accesibilidad.

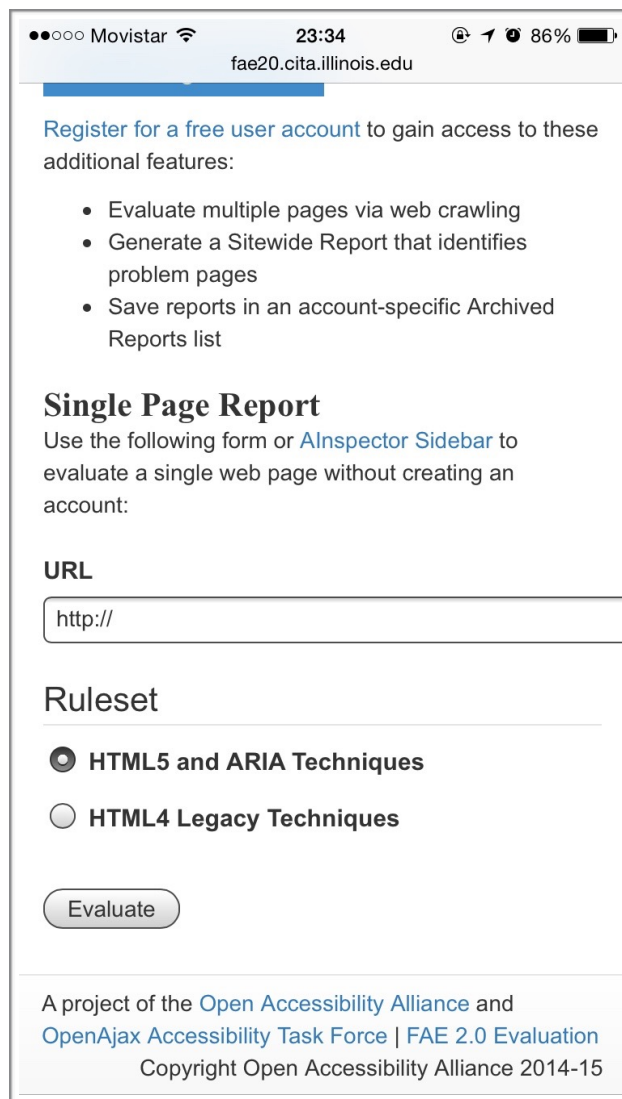


Figura 3.4.1.4: FAE accedido desde un iPhone 6.

WAVE - Web Accessibility Evaluation Tool

WAVE [WebAIM, 2014] es una aplicación web que muestra la página web original con íconos e indicadores integrados que revelan la accesibilidad de la página, en lugar de emitir reportes como es el caso de los otros validadores analizados. Acepta el ingreso de páginas por URL, carga de archivos, o código fuente. Este validador provee sugerencias de corrección a los errores encontrados. Esta herramienta está disponible sólo en idioma inglés. Al no brindar un acceso al código fuente ni ningún enlace de descarga, no puede ser utilizada en este proyecto para ser extendida.

Además del servicio web, el validador WAVE también está disponible como una barra de herramientas de Firefox y extensión para Dreamweaver.

WAVE clasifica los problemas encontrados en 3 categorías:

- **Error:** Uno o más elementos no cumplen los requisitos de la regla evaluada.
- **Advertencia:** Uno o más elementos no cumplen los requisitos recomendados por la regla

evaluada.

- **Estructural:** No se puede determinar automáticamente el resultado. La evaluación requiere intervención humana.

Además, WAVE informa sobre las características de accesibilidad encontradas.

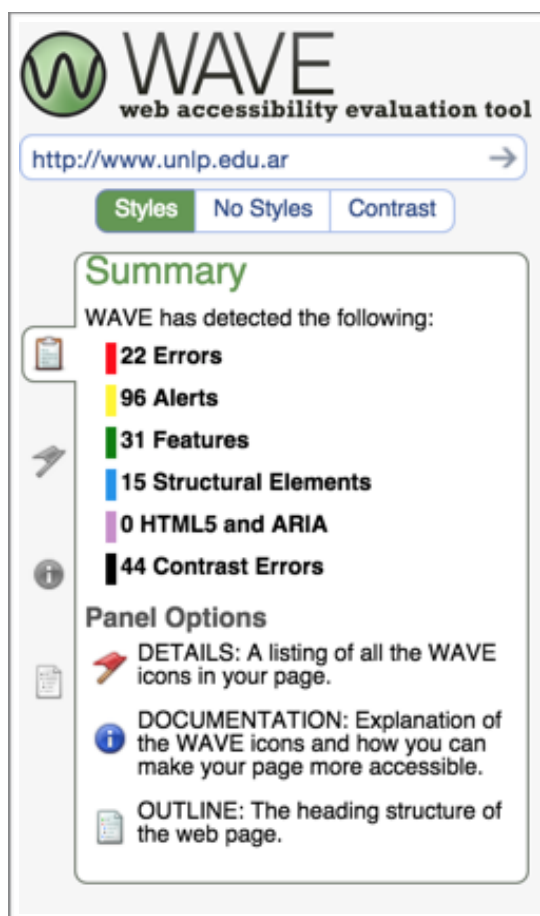
Esta herramienta permite visualizar los resultados en tres (3) tipos de vista:

- **Con estilos:** Se muestra el sitio web validado con los íconos y los indicadores de accesibilidad incorporadas.
- **Sin estilos:** Se muestra el sitio web validado con los estilos deshabilitados, incluyendo también los indicadores de accesibilidad. Este tipo de visualización permite ver el orden en el que teclado y los lectores de pantalla navegan el sitio web.
- **Con contraste:** Muestra todos los aspectos del sitio relacionados con cierto contraste.

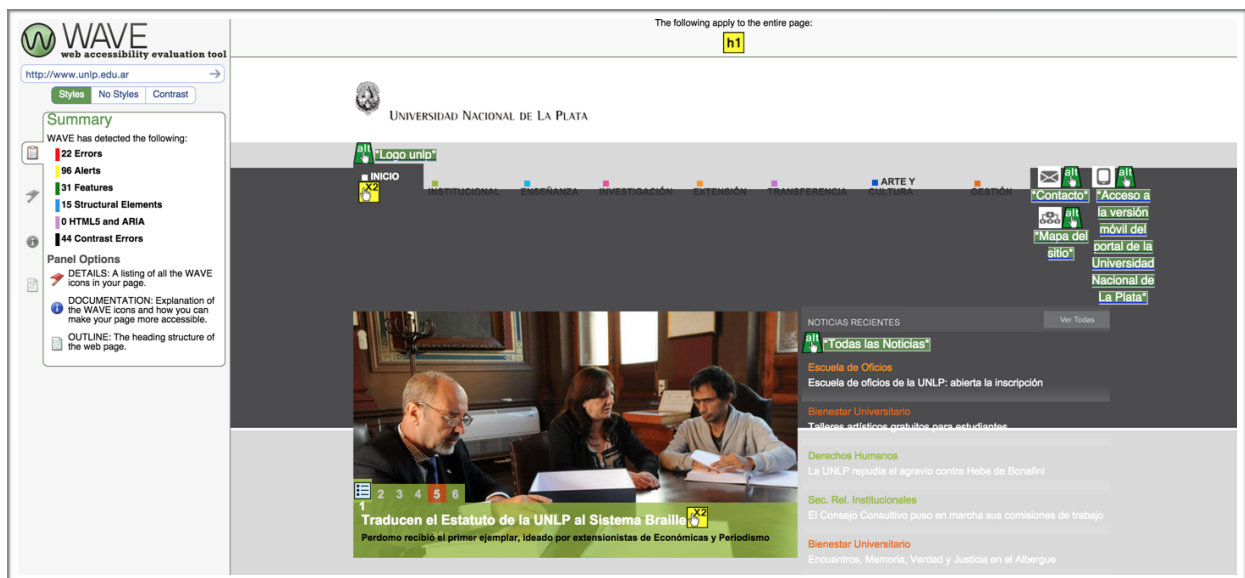
Además cuenta con:

- Un listado de íconos, ordenados por tipo, que indican los resultados de la validación del sitio.
- Explicación de los íconos y recomendaciones para solucionar los problemas encontrados.
- La estructura de encabezados del sitio web.

Las figuras 3.4.1.5.a y 3.4.1.5.b ilustran el resultado de una validación de una página del sitio de la UNLP, mediante WAVE.



Captura 3.4.1.5.a: Informe HTML de WAVE (Sumario).



Captura 3.4.1.5.b: Informe HTML de WAVE (Errores marcados en el documento).

En la figura 3.4.1.6 se muestra una captura de WAVE en un dispositivo móvil (iPhone 6). Como puede observarse, WAVE no dispone de una interfaz adaptable al dispositivo, lo cual decremента su nivel de accesibilidad.

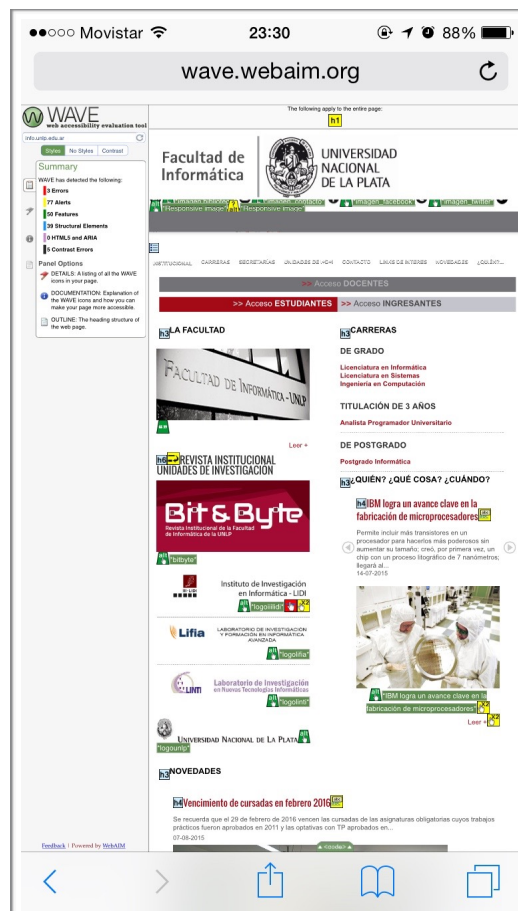


Figura 3.4.1.6: WAVE accedido desde un iPhone 6.

TAW - Test de Accesibilidad Web

TAW [CTIC, 2014] es una herramienta privativa que permite comprobar de forma automática ciertos aspectos de la accesibilidad web.

Este validador dispone de dos (2) modalidades de ejecución: analizador en línea (desde su sitio web), o como aplicación instalable de forma local. La versión web permite analizar solamente enlaces aislados, mientras que otras versiones pagas de TAW, las cuales no se estudiaron en esta tesina por no ser de libre acceso, permiten otros modos de análisis. Además, se presenta en los idiomas inglés y español.

El analizador en línea funciona introduciendo una URL del sitio web que se pretende analizar, generando un informe HTML con información sobre el resultado del análisis.

Este último está disponible para las siguientes normativas:

- Pautas de accesibilidad al Contenido Web 1.0 (WCAG 1.0).
- Pautas de accesibilidad al Contenido Web 2.0 (WCAG 2.0).
- Requisitos de accesibilidad para contenidos en la Web (UNE 139803).

La aplicación instalable de forma local, analiza únicamente las WCAG 1.0 pero incluye una serie de características adicionales como son:

- Análisis multipágina.
- Permite la revisión de las incidencias manuales, incluyendo la posibilidad de añadir comentarios.
- Informe del resumen de accesibilidad de las páginas analizadas.
- Configuración individual de los puntos de verificación a analizar.
- Creación de verificaciones personalizadas.

La clasificación de problemas que este validador realiza en su evaluación es la siguiente:

- **Problemas:** Es necesario realizar correcciones.
- **Advertencias:** Deben revisarse manualmente.
- **Puntos no verificados:** Requieren un análisis manual completo.

El resultado de la validación es emitido en un documento resumen, donde se especifican los problemas, advertencias y puntos que requieren un análisis manual completo organizados por cada principio de evaluación (Perceptible, Operable, Comprensible y Robusto).

Además, TAW permite visualizar los resultados en tres tipos de vistas:

- **Marcada:** Se señalan sobre la página web las incidencias detectadas.
- **Detalle:** Agrupadas por cada uno de los principios básicos, se indican las comprobaciones atómicas realizadas mostrando las líneas de código donde se detectan incidencias. A nivel informativo, se muestran las técnicas relacionadas con cada una de las comprobaciones.
- **Listado:** Es un resumen agrupado por cada uno de los principios básicos, en formato de tabla que indica, a nivel de normativa, el resultado obtenido en las comprobaciones a nivel de pauta.

t.a.w.
CTIC Centro Tecnológico

Summary | Marked view | **Details** | Listing

Details view

Analysis information

Resource: <http://www.unlp.edu.ar>
 Date: 28/03/2015 16:37
 Guidelines: WCAG 2.0
 Analysis level: AAA
 Technologies: HTML, CSS

Perceivable | Operable | Understandable | Robust

Information and user interface components must be presentable to users in ways they can perceive.

Tipology	Comprobation	Technique	Result	Problems	Line numbers
1.1.1 - Non-text Content					
Images	Images that may require a long description ?	[H45]	!	30	84, 109, 110, 116, 199, 322, 326, 330, ...
	Images with empty "alt" attribute ?	[H67]	!	15	184, 362, 364, 366, 370, 372, 374, 378, ...
Forms	Form controls without label ?	[H44, H65]	✘	1	295
Objets	Links to audio files that may need transcriptions ?	[H53]	!	1	688
1.2.1 - Audio-only and Video-only (Prerecorded)					
Multimedia	Prerecorded media that may need equivalent alternatives ?	[G159, G166]	!	1	688
1.2.2 - Captions (Prerecorded)					
Multimedia	Prerecorded media that may need equivalent alternatives ?	[G93, G97]	!	1	688

Captura 3.4.1.9: Informe HTML de TAW (Vista detallada).

En la figura 3.4.1.10 se muestra una captura de TAW en un dispositivo móvil (iPhone 6). Como puede observarse, TAW no dispone de una interfaz adaptable al dispositivo, lo cual decreta su nivel de accesibilidad.



Captura 3.4.1.10: TAW accedido desde un iPhone 6.

3.4.2. Comparativa de características

En la *Tabla 3.4.2.1* se muestran las diferencias existentes entre los validadores elegidos en base a las características explicadas al comienzo de este capítulo y según la encuesta realizada.

	AChecker	FAE	WAVE	TAW
Alcance	Un enlace por vez	Completo (sólo con registraci3n gratuita)	Un enlace por vez	Un enlace por vez
Propuesta Correctiva	Si	No	Si	Si
Estilo de informe (exportable)	HTML PDF EARL CSV	HTML	HTML	HTML
Múltiples idiomas	Sí	No	No	Sí
Cuenta de usuario con historial	No	No	No	No
Plataforma	Múltiples plataformas (web)	Múltiples plataformas (web)	Múltiples plataformas (web)	Múltiples plataformas (web)
Interfaz adaptable	No	No	No	No
Licencia	GPL	No disponible	No disponible	Privativa

Tabla 3.4.2.1: Tabla comparativa de validadores en base a las características analizadas.

3.4.3. Comparaci3n de los resultados de validaci3n emitidos

Para completar el análisis de estos validadores, se realizaron pruebas sobre la versi3n accesible del sitio de la Universidad Nacional de La Plata. A continuaci3n se resumen los resultados de estas pruebas.

Página evaluada: Página inicial

Fecha: Septiembre del 2014

URL: <http://www.unlp.edu.ar>

Reglas: WCAG 2.0 A, AA y AAA.

En la *Tabla 3.4.3.1* se muestran los resultados de las evaluaciones realizadas en base a los tres (3) criterios que especifica la W3C:

- **No pasadas:** este criterio agrupa todos los resultados que involucren reglas no cumplidas.
- **Pasadas:** este criterio agrupa todos los resultados que involucren reglas cumplidas.
- **Intervención humana:** este criterio agrupa todos los resultados que involucren reglas cuyo resultado no se puede determinar programáticamente, y se requiere la intervención del desarrollador.

Como puede observarse en dicha tabla, no existe consistencia en los resultados de las validaciones efectuadas por estas herramientas, lo que nos demuestra que los procesos de validación no respetan estrictamente las pautas establecidas por la W3C para determinar si el código evaluado es o no accesible.

	AChecker	FAE	WAVE	TAW
No pasadas	7	4	3	36
Pasadas	N/A	17	6	N/A
Intervención humana	45	16	6	253

Tabla 3.4.3.1 - Resultado de criterios evaluados en el mes de Septiembre del 2014

A continuación, la figura 3.4.3.2 muestra una representación gráfica de los datos de la *Tabla 3.4.3.1*:

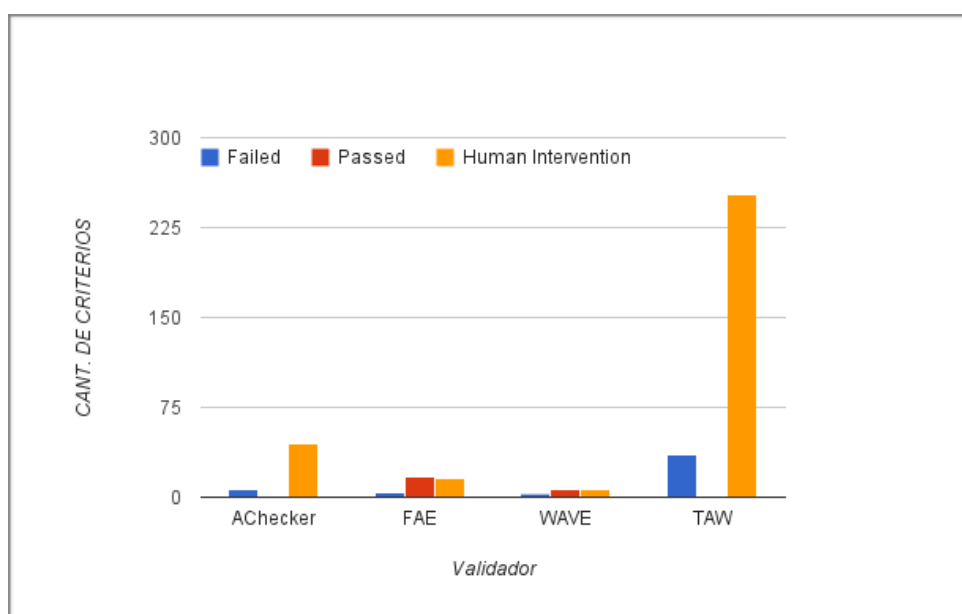


Figura 3.4.3.2 - Representación gráfica de los datos de la Tabla 3.4.3.1

Una medida importante de la eficacia de estos productos es su capacidad para producir resultados precisos y definitivos sin la necesidad de la interpretación humana.

3.4.4. Correcciones que requieren intervención humana

Como puede verse en la figura 3.4.4.1 algunas herramientas dejan en manos del programador aspectos que podrían ser evaluados automáticamente, clasificando la mayoría de los criterios como de “Interpretación Humana”.

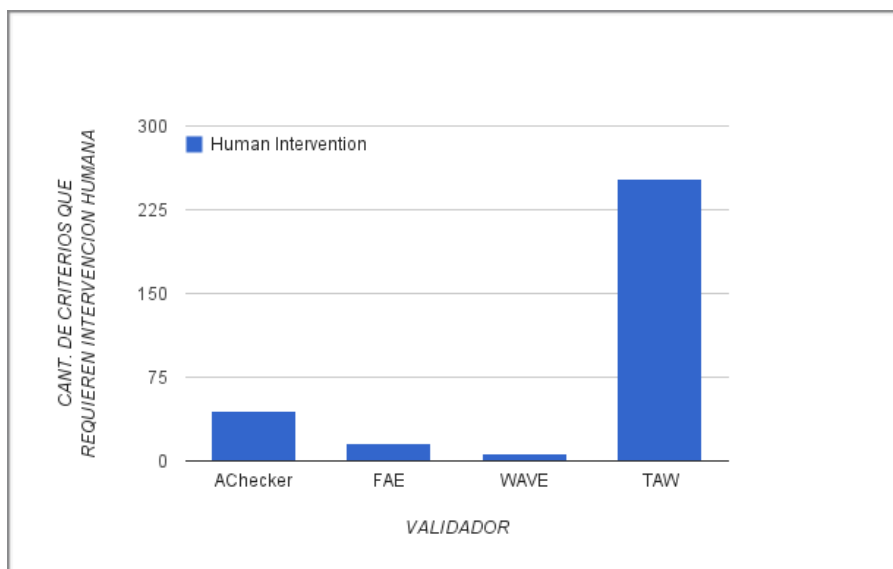


Figura 3.4.4.1 - Cantidad de criterios que requieren intervención humana

En la Tabla 3.4.4.2 se citan algunos de los criterios que son interpretados como barreras de accesibilidad por algunos validadores y como potenciales barreras en otros.

Validador	Sección	Tipo de violación
aChecker	1.3.1	Potencial
FAE	1.3.1	NO EXISTE
WAVE	1.3.1	No agrupa por sección
TAW	1.3.1	Violación
aChecker	3.2.3	Potencial
FAE	3.2.3	Violación
WAVE	3.2.3	No agrupa por sección
TAW	3.2.3	NO EXISTE

Tabla 3.4.4.2 - Resultado de criterios evaluados en el mes de Septiembre del 2014
 * Sólo se incluyen algunas secciones, con el fin de ejemplificar la disparidad entre validadores

3.5. Conclusión

Luego de haber examinado minuciosamente la oferta de mercado actual en materia de validadores de accesibilidad, se encontró una marcada diferencia en la exposición de los análisis que realizan cada uno de éstos; es aquí donde nace la necesidad de contar con una herramienta que no sólo realice un análisis exhaustivo de accesibilidad, sino que además lo haga respetando todos los estándares definidos por los entes correspondientes, como es el caso de la W3C. Si bien cada uno intenta aportar algo significativo a la construcción de una Web accesible, lo hacen meramente a su manera dejando de lado incluso hasta la terminología y las secciones definidas en el estándar.

Un informe de accesibilidad, o cualquier informe que describa un grado de cumplimiento en cuanto a especificaciones o normas, debe ser de interpretación general, y poder ser comparado con otros informes de la misma índole; eso es lo que se propone con SiMor: un análisis de accesibilidad de alcance global, en donde no importe el producto o la versión sobre la cual se realizó, sino que se lo analice directamente con lo que especifica el ente regulatorio en cuestión (haciendo uso del formato EARL).

Dicho esto, y finalizando la propuesta de SiMor en cuanto al esquema del informe a utilizar, se deja nota de que el proceso de validación, detallado en el próximo capítulo, está basado en cada una de las reglas WCAG 2.0 tal cual fueron especificadas, definidas por el ente anteriormente mencionado. Esto garantiza la máxima fidelidad entre los estándares definidos y el resultado obtenido.

Capítulo 4: SiMor

4.1. ¿Qué es SiMor?

SiMor, acrónimo de las palabras en inglés See (ver) y More (más), es un software de código abierto publicado bajo la licencia GNU/GPLv3. Nace como una iniciativa que intenta cubrir algunas de las debilidades encontradas en los validadores analizados, enfocadas desde el punto de vista del desarrollador web. Con esto se pretende lograr un nuevo enfoque en materia de validación integral de accesibilidad web. La versión de uso público de SiMor está disponible en <http://simorwai.com>.

Un validador de accesibilidad, tal como lo conocemos hoy en día, permite analizar el código HTML estático de un sitio web con el objetivo de identificar errores que impliquen una disminución en la accesibilidad del mismo, emitiendo reportes con los resultados de esas validaciones (en muchos casos sin respetar el formato estándar EARL recomendado por la W3C), y en algunos casos con referencias a la especificación de las reglas no cumplidas.

A pesar de la utilidad de estas herramientas, y en base a al análisis realizado previamente, se pudo ver que no siempre cumplen con todas las expectativas y/o necesidades que un desarrollador web espera, y tampoco dan el soporte necesario para realizar un seguimiento de su trabajo. Además, de los validadores estudiados con licencia GPL que posibilitan su libre uso y modificación, ninguno está desarrollado de manera legible y extensible, resultando casi imposible plasmar todas las características propuestas por SiMor en un tiempo y forma considerables, dejando como única opción el desarrollo de una nueva herramienta.

A partir de esto se entiende que el proceso de evaluación de accesibilidad de un sitio implica algo más que una *evaluación – resultado*, es en cambio un proceso en donde se necesita conocer lo que se está evaluando, por qué una regla es considerada válida o no, y cómo lograr que esa validación sea exitosa. Por eso SiMor, además de permitir el análisis parcial o total de un sitio basado en estándares internacionalmente aceptados, permite realizar el seguimiento de su evolución en el tiempo, emitiendo reportes claros y concisos, donde además de describir el grado de accesibilidad de un sitio en particular, brinda una propuesta de solución ad hoc (específica) a cada error detectado.

SiMor es un analizador intensivo basado en reglas porque su desarrollo está basado en la especificación de las reglas WCAG 2.0 de la W3C, donde cada regla se corresponde con una instancia de análisis independiente, y cada una de ellas consiste en la definición de las condiciones para su cumplimiento, además de la sugerencia de una posible solución.

Plantear el uso de reglas hace que SiMor sea extensible ante cambios o nuevas especificaciones en las WCAG, y provea un modo de análisis que no se encuentra en los validadores actuales.

Para permitir el seguimiento de las validaciones, SiMor requiere la creación previa de una cuenta de usuario, lo que permite ver el progreso del desarrollo en aspectos de accesibilidad, lo que hace

que, a través de un sistema de versionado, las validaciones realizadas no sean efímeras, sino que formen parte de un proceso de evaluación continuo, pudiendo realizarse comparaciones con validaciones previas, analizar causas de violación de reglas mediante la información proporcionada, tanto de la especificación de cada pauta evaluada como de las propuestas de solución y ver el código fuente del sitio en el momento de la validación.

Otra característica que permite el funcionamiento a través de una cuenta de usuario, es que cada vez que se valida un sitio web, SiMor realiza el crawling, es decir el recorrido de todos los enlaces del sitio y la obtención de cada código HTML respectivo, en segundo plano. Otra cualidad que diferencia a SiMor respecto de los demás validadores, es que permite la definición de wildcards, que son una suerte de comodines en expresiones regulares para buscar patrones en cadenas de caracteres, y así excluir enlaces que no desean validarse.

El proceso de crawling se basa en recorrer todos los enlaces válidos y excluir aquellos que se especificaron a través de los wildcards. Una vez finalizado dicho proceso, se cuenta con una copia exacta del código HTML que representa al sitio en cuestión, y queda ligado al usuario que lo generó. Todo esto es persistido en una base de datos.

Cada vez que un usuario realiza una validación, SiMor procesa la solicitud en segundo plano, emitiendo alertas para la notificación de eventos de interés, como puede ser la finalización de una evaluación. Esto da libertad al usuario al permitirle la utilización de otras funcionalidades de SiMor mientras el análisis de un sitio está en progreso, como por ejemplo ver el resultado de otra validación, realizar otra, o cualquier otra funcionalidad provista.

Otra característica de SiMor que actualmente no se encuentra en otros validadores es la posibilidad de analizar reglas que requieren el acceso a declaraciones de estilos externas (estilos no definidos en línea) para los elementos del documento analizado. Esto posibilita validar, por ejemplo, el grado de contraste entre elementos.

Además, se obtiene el código HTML resultante tal cual sucede en un navegador, ya que el proceso de obtención trabaja mediante un navegador real, permitiendo la ejecución de scripts que modifican el DOM¹⁷ resultante.

Otra característica es la de generar una captura de pantalla de cada enlace del sitio (renderizado por un navegador en particular).

Además, como se mencionó anteriormente, SiMor permite la emisión de reportes en dos formatos principales: PDF y EARL. La principal ventaja de la utilización del formato EARL es permitir el intercambio, la combinación y comparación de resultados provenientes de distintas herramientas de evaluación (que ofrezcan dicho estándar de reporte) y analizar el resultado en contraste con lo especificado por la W3C.

Antes de adentrarnos en la siguiente sección, en la cual se describen en detalle las funcionalidades mencionadas anteriormente, junto con las particularidades de la implementación en materia profesional y tecnológica, cabe destacar uno de los puntos más innovadores que SiMor presenta en lo que respecta al resultado de una validación de accesibilidad web.

¹⁷ *Interfaz de programación de aplicaciones (API) que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos*

Si bien se incluyen los elementos básicos de un informe de validación, como son las reglas válidas e inválidas, se tienen además las reglas que requieren la atención humana, y son las que la W3C especificó de esta manera: aspectos meramente subjetivos, como puede ser que el texto de cierto elemento tenga correlación con el párrafo anterior, o que un vídeo deba incluir los subtítulos en perfecta sincronización con el audio. A priori, SiMor marca como inválida una regla de este tipo, para forzar la intervención del desarrollador web, pero en caso de ser incorrecta la validación sobre esa regla, el usuario puede marcarla como un falso positivo, dando lugar a que impacte en el veredicto final de la validación. Esto es una cualidad que evita tener validaciones erróneas, cuando en realidad no lo son (como pasa en los validadores que hoy se pueden encontrar en el mercado).

Como complemento, SiMor permite la visualización del código fuente de cada enlace del sitio, resaltando en dicho código el elemento validado por cada regla a fin de facilitar al desarrollador la localización del punto de falla.

A continuación, se detallan los aspectos funcionales y de implementación.

4.2. Definición del proceso de desarrollo

Para definir el proceso de desarrollo de SiMor se empleó la metodología ágil Scrum [Kenneth S. Rubin, 2012]. Este proceso se dividió en diez (10) sprints¹⁸ de veintiún (21) días de duración cada uno. Dado que el equipo de este proyecto es de tamaño reducido (dos integrantes), ambos ejecutaron el rol de Scrum Master¹⁹ y Product Owner²⁰. En cada fin de sprint se hicieron las reuniones de retrospectiva del sprint finalizado y planeamiento del siguiente.

Para la planificación y el seguimiento de cada uno de los sprints se utilizó la herramienta Engageful²¹, la cual es un sistema web de gestión de proyectos de uso gratuito que implementa Scrum, y es desarrollada por FluxIT²².

4.3. Decisiones de diseño

Durante la implementación de SiMor surgieron diversos inconvenientes que obligaron a investigar diferentes herramientas con el objetivo de encontrar aquellas que permitan cumplir con las especificaciones planteadas, con la mayor eficiencia posible.

El primer inconveniente surgió al momento de definir la forma de modelar las reglas a evaluar: al

¹⁸ *Sprint: periodo entre una (1) y cuatro (4) semanas, definido por el equipo, en donde se establecen objetivos a cumplir.*

¹⁹ *Mantiene los procesos y trabaja de forma similar al director de proyecto.*

²⁰ *Representa a los stakeholders (interesados externos o internos) del proyecto.*

²¹ <http://www.engageful.com.ar>

²² <http://www.fluxit.com.ar>

ser el *modelo de reglas* el núcleo de SiMor, se tuvo especial cuidado en su definición, ya que de esto depende el buen funcionamiento y fiabilidad del proyecto.

A partir de esto se decidió la utilización de los Requerimientos de Codificación de Reglas de Accesibilidad de OpenAjax Alliance²³ (OAA de ahora en más), que definen la organización de las mismas a partir de los requisitos de accesibilidad web definidos por las WCAG 2.0. El listado completo de las reglas que se incluyen en SiMor se corresponde al publicado en <http://oaa-accessibility.org>; en el Anexo de este informe se puede encontrar el listado completo de las mismas.

Cada regla de validación OAA se asocia con los requerimientos de un archivo del conjunto de reglas de las WCAG 2.0.

La utilización de este modelo permitió disminuir la complejidad tanto de la interpretación y definición de las reglas como su implementación, como se detalla en la sección 4.4.3.

Cabe destacar que el listado de reglas fue analizado y verificado contra el documento de las pautas WCAG 2.0 de la W3C, y completado con casos de éxito y de falla, además de su traducción completa al español, para orientar al usuario sobre cada una de las reglas.

En una primera instancia optamos por la utilización de una base de datos relacional (en concreto, MySQL) que posteriormente, a causa del volumen de datos utilizados y la performance obtenida, fue reemplazado.

Como alternativa se utilizó una base de datos no relacional, bajo el concepto de NoSQL [Prmod J. Sadalage and Martin Fowler, 2012], y se optó por MongoDB que es un sistema de base de datos orientado a documentos y desarrollado bajo licencias de código abierto, la cual se detalla en la sección 4.4.2.

4.4. Implementación de SiMor

SiMor es una aplicación web monolítica y autocontenida, desarrollada utilizando Java 8 [Raoul-Gabriel Urma et al., 2014] como plataforma base. Dichas tecnologías se decidieron en base a los requerimientos funcionales de SiMor. Dado que es una aplicación multiusuario, con alto grado de procesamiento y alta velocidad, fueron numerosos los aspectos que motivaron el uso de una plataforma que ofrezca multithreading, rápida respuesta y, sobre todo, escalabilidad.

Además, su código fuente cumple con los estándares propuestos por la plataforma [Oracle, 1999].

4.4.1. Java como plataforma

Se decidió la utilización de este lenguaje porque posee un conjunto de características que son sumamente relevantes para el tipo de aplicación desarrollada.

A continuación se explican las ventajas de cada una de dichas características y cómo aplican al proyecto.

²³ <http://oaa-accessibility.org>

Gestión de Memoria

Cuando el desarrollo de un software implica cierta cantidad de memoria RAM para albergar datos, es necesario solicitar esa memoria al sistema operativo. En algunos lenguajes, como puede ser C, cuando el programa no necesita utilizar más esa memoria, es necesario que explícitamente la devuelva al sistema, por lo que si el programador se olvida de liberar algo de memoria cuando el programa termina, esa memoria queda ocupada (aunque el programa ya no siga ejecutándose), provocando un consumo excesivo y agotando la cantidad disponible para otros usos.

En Java, a diferencia de otros lenguajes, existe un proceso independiente denominado Garbage Collector que se encarga de liberar automáticamente toda la memoria que ya no se utiliza, siendo este proceso transparente al programador.

SiMor utiliza mucha memoria en cada validación, por lo que esta característica es muy importante a la hora de optimizar los recursos, ya que una incorrecta administración de memoria haría que la aplicación no funcione de la manera esperada, o incluso deje de funcionar.

Comunidad Open Source

Otro de los motivos de gran potencial es la gran comunidad de desarrolladores existente. Esto es de vital importancia en el desarrollo de SiMor a la hora de consultar información acerca de las tecnologías más utilizadas y cuáles son importante incluir o no en el desarrollo de la aplicación.

Multithreading y Sincronización

Existen diversos mecanismos para sincronizar los hilos en el acceso a los recursos del sistema, de manera que dos hilos no puedan acceder a la vez al mismo recurso. Java da soporte a esto de manera ad hoc.

Dado que SiMor requiere que todas las reglas a validar en un documento HTML sean ejecutadas en paralelo, esta característica fue esencial para definir las tecnologías a utilizar en su desarrollo.

4.4.2. Tecnologías utilizadas

Además de la propia plataforma, se utilizaron tecnologías (todas de código abierto), que serán descritas a lo largo de esta sección.

Spring framework

Según la última encuesta de RebelLabs sobre frameworks Web Java [Oliver White, 2014], Spring es el framework más utilizado en el mundo Java. Su funcionalidad básica es la de actuar como un contenedor de objetos (beans), encargándose de su ciclo de vida y de cómo estos se relacionan entre sí. Además provee Inyección de Dependencias (DI) y Aspectos, lo cual permite el desacople de los componentes de la aplicación.

Además de estas funciones, dispone de un conjunto de módulos para usos específicos, siendo los siguientes los utilizados por SiMor:

- **Security:** provee Autenticación y Autorización para aplicaciones.
- **Data:** capa de acceso para distintos tipos de bases de datos.
- **Boot:** favorece la convención sobre la configuración para crear aplicaciones listas para un ambiente productivo, además de tener el servidor web embebido.

Gradle

Gradle [Tim Berglund, 2013] es una herramienta de automatización de proyectos construida sobre Apache Maven y Apache Ant, introduciendo un DSL basado en Groovy (en lugar del tradicional en XML utilizado por sus predecesores) para describir la configuración del proyecto. Además se introduce el concepto de Grafo Acíclico Dirigido (DAG) para determinar el orden en que las tareas deben ejecutarse. Gradle fue diseñado para dar soporte a múltiples proyectos que pueden tener una amplia línea de crecimiento, y soporta builds incrementales determinando inteligentemente qué partes están actualizadas para que las tareas que dependen de dicho build no sean nuevamente ejecutadas.

MongoDB

MongoDB (de la palabra en inglés “humongous”, que significa enorme) es un sistema de base de datos NoSQL orientado a documentos [Kristina Chodorow, 2013]. Este tipo de base de datos, en lugar de guardar los datos en tablas como en las base de datos relacionales, guardan estructuras de datos en documentos tipo JSON con un esquema dinámico (BSON en MongoDB), lo cual alivia la sobrecarga necesaria en caso de, por ejemplo, exponer una API RESTful de servicios (ya que éstas exponen su contrato en formato JSON).

Según el estudio *Comparativa MongoDB vs. MySQL* [José Manuel Ciges Regueiro, 2013], se concluye que MongoDB:

- Es más rápido en las operaciones de lectura y escritura que MySQL.
- Para operaciones de escritura sencillas, es hasta 4 veces más rápido que MySQL.
- Para un número de escrituras elevado, duplica en rendimiento a MySQL.
- Es más rápido en escrituras y lecturas concurrentes que MySQL.

Las anteriores características hicieron que MongoDB sea el almacenamiento utilizado por SiMor en lugar de MySQL.

Thymeleaf

Thymeleaf²⁴ es una librería Java que implementa un motor de plantillas (templates). Sin adentrarse en vastos detalles acerca de lo que significa una utilidad como ésta, sí se dan a conocer sus ventajas principales, que son las que dieron iniciativa a la elección de esta tecnología en la implementación de SiMor.

La principal característica que lo diferencia de las demás implementaciones es que es un validador de XML/HTML5, esto garantiza que el código HTML generado es válido; se lanza una excepción en

²⁴ <http://www.thymeleaf.org/>

ejecución en caso de no serlo. La otra cualidad importante es su sintaxis escrita en HTML5.

A partir de la estructura HTML5 expuesta en el código 4.4.2.1 , para no tener que reescribir toda la maqueta en una sintaxis distinta, Thymeleaf propone dejar intacta la estructura HTML para que el documento resultante una vez procesado sea un listado dinámico igual que el provisto, sólo con el agregado de atributos, como se muestra en el código 4.4.2.2.

```
<!DOCTYPE html>
<html>
<head>
  <title>Listado de frutas</title>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th>Nombre</th>
        <th>Precio</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <td>Naranjas</td>
        <td>0.99</td>
      </tr>
    </tbody>
  </table>
</body>
</html>
```

Código HTML 4.4.2.1: Estructura HTML5 de ejemplo.

```

<!DOCTYPE html>
<html>
<head>
  <title th:title="#{frutas.listado.title}">Listado de frutas</title>
</head>
<body>
  <table>
    <thead>
      <tr>
        <th th:text="#{msgs.headers.name}">Name</th>
        <th th:text="#{msgs.headers.price}">Price</th>
      </tr>
    </thead>
    <tbody>
      <tr th:each="prod : ${allProducts}">
        <td th:text="{prod.name}">Oranges</td>
        <td th:text="{#numbers.formatDecimal(prod.price,1,2)}">0.99</td>
      </tr>
    </tbody>
  </table>
</body>
</html>

```

Código HTML 4.4.2.2: Estructura HTML5 resultante llevada a una plantilla Thymeleaf.

Como se puede notar, el navegador (al abrir el archivo directamente en él) ignora los atributos que comiencen con **th:** ya que no los puede interpretar, pero el servidor hace el procesamiento correspondiente.

Con esto se logra eliminar la barrera desarrollador - diseñador, permitiendo evitar por completo el re-trabajo que conlleva implementar un diseño en particular.

Apache Camel

Apache Camel es un motor de ruteo y mediación basado en reglas. Básicamente, sirve para enviar datos hacia diferentes lugares (endpoints), y recibirlos. Además, se pueden aplicar transformaciones en el envío y/o recepción de los datos. En SiMor se utiliza este componente para la ejecución de reglas en paralelo (ya que también provee este tipo de ejecución, de manera atómica) [Claus Ibsen, 2011].

Siendo que esta tesina es una aplicación monolítica que no consume ningún servicio externo, se puede poner en tela de juicio la justificación del uso de este componente. Al hacer uso de esta tecnología se tiene la posibilidad, con poco esfuerzo, de poder enviar documentos hacia servicios externos de terceros, o bien desacoplar el motor de validación de SiMor y publicarlo como una API para que luego la aplicación web haga uso de ésta mediante Camel.

Selenium WebDriver y PhantomJS

Selenium WebDriver [Satya Avasarala, 2014] es un servidor de control remoto (RC) que expone una API para recibir comandos vía HTTP, el cual recibe comandos y los envía a un navegador web. Selenium es utilizado para manejar un navegador web de manera programática, sin necesidad de un usuario intermedio. Esto se implementa a través de un controlador específico para cada tipo de

navegador que envía los comandos y trae los resultados de regreso.

En SiMor se decidió utilizar PhantomJS [Aries Beltran, 2013] como navegador a interactuar con Selenium, por su performance y popularidad. Se trata de un navegador headless (sin interfaz gráfica, utilizado mediante comandos locales o remotos) basado en WebKit, y con esta combinación se obtiene el documento HTML real de la aplicación analizada por SiMor, ya que se ejecuta en un navegador junto con todos los scripts que contenga.

Esta necesidad surge porque hoy, la mayoría de las aplicaciones genera código HTML dinámicamente en el lado cliente a través de Javascript, y para interpretar dicho código es necesario un navegador real. Lo más importante es que se tiene acceso al DOM completo “en vivo”, pudiendo tener control del modelo de eventos generados.

Lo más importante es que se trabaja con un documento resultante de la interpretación sobre un navegador real, no mediante una interpretación de código HTML estático. Esto nos brinda la posibilidad de que la evaluación de accesibilidad se acerque mucho más a la realidad. Cabe destacar que esta funcionalidad no está provista por ningún validador actual.

CometD

En el desarrollo web, Comet es un neologismo para describir un modelo de aplicación web en el que una petición HTTP mantenida abierta permite a un servidor web enviar datos a un navegador mediante tecnología Push, sin que el navegador los solicite explícitamente.

En teoría, el enfoque Comet difiere del modelo original de la web, en el que un navegador solicita una página web completa o trozos de datos para actualizar la misma. En cualquier caso, en la práctica, las aplicaciones Comet usan tradicionalmente programación AJAX con una petición prolongada para detectar nueva información en el servidor.

La diferencia que existe entre AJAX y Comet es que en este modelo se mantiene una conexión abierta entre el cliente y el servidor web; el cliente no solicita los datos, pero sí envía información al servidor, y el servidor no le responde al cliente con un bloque de datos, sino que se espera por algún evento del lado del servidor para enviar la información.

CometD²⁵ es una implementación del patrón AJAX Push, conocido como Comet. SiMor utiliza este enfoque para proveer notificaciones en el navegador para la notificación de eventos, tales como la finalización de una validación de accesibilidad de un sitio web.

En SiMor se decidió utilizar el enfoque propuesto por CometD, ya que el envío de notificaciones al usuario es más eficiente comparado con el polling tradicional.

²⁵ <http://cometd.org>

Jsoup

Jsoup²⁶ es un analizador (parser) de HTML que utiliza selectores CSS3 para recorrer el árbol. Éste es el componente base que SiMor utiliza para analizar código HTML de manera estática. Cuenta con soporte nativo para HTML5, permitiendo además obtener el código desde una cadena de caracteres, una dirección web o un archivo.

Para mostrar la sintaxis de selección, se cita el ejemplo del autor expuesto en el código Java 4.4.2.3:

```
Document doc = Jsoup.connect("http://en.wikipedia.org/").get();
Elements newsHeadlines = doc.select("#mp-itn b a");
```

Código 4.4.2.3: Selección de elementos en un documento HTML mediante Jsoup.

Apache JCS Cache

JCS es una caché²⁷ distribuida escrita en Java. Provee caché en memoria, disco o a través de la red, ya que escala horizontalmente por descubrimiento de nodos. SiMor necesita hacer uso de un sistema de caché dado que sería totalmente ineficiente pedir los mismos recursos una y otra vez durante toda la validación de un sitio web, dado que naturalmente en un sitio se van a repetir los archivos CSS y las imágenes. Se tienen dos (2) cachés: una para las hojas de estilo y otra para las imágenes. Para el alcance de esta tesina, se estipula un tiempo de validez de los datos al equivalente en horas de una (1) semana; pasado este tiempo los recursos se eliminarán y, en caso de ser necesario, se tendrán que solicitar nuevamente.

4.4.3. El modelo de reglas

A modo de referencia cada regla se identifica por un identificador (ID), que se corresponde con el ID de regla establecido por las OAA, junto con la descripción de lo que debe validar, el grupo de requerimientos al que pertenece y el enlace a su especificación.

La implementación de SiMor consta de clases de tipo Rule que contienen la siguiente sección JavaDoc describiendo el propósito de la regla, como se muestra en el código 4.4.3.1:

²⁶ <http://jsoup.org>

²⁷ <http://commons.apache.org/proper/commons-jcs>

```

/**
ID = 3
DESCRIPTION = Data tables must use summary attribute
REQUIREMENT_NUMBER = 1.3.1
REQUIREMENT_URL={@link<a href="http://www.w3.org/TR/WCAG20/#content-structure-
separation-programmatic">http://www.w3.org/TR/WCAG20/#content-structure-
separation-programmatic</a>}
*/

```

Código Java 4.4.3.1 : implementación de una regla.

La implementación de cada regla, está estructurada como se muestra en el código 4.4.3.2:

```

@RuleType(level = RuleLevel.WCAG2_A)
public class RuleWCAG2A014 extends Rule {

    @Override
    protected String context() {
        return "body";
    }

    @Override
    public RuleSeverity severity() {
        return RuleSeverity.RECOMMENDATION;
    }
}

```

Código Java 4.4.3.2: implementación de una regla.

```

@Override
protected boolean breaks(Element element, String htmlCode) {
    Elements h1Elements = element.select("h1");
    boolean breaks = h1Elements.isEmpty();

    Iterator<Element> iterator = h1Elements.iterator();
    while ((iterator.hasNext()) && (!breaks)) {
        Element h1Elem = iterator.next();
        breaks = h1Elem.text().isEmpty();
    }
    return breaks;
}

@Override
protected void suggestElement(Element element) {
    Elements h1Elements = element.select("h1");
    if(!h1Elements.isEmpty()){
        for (Element h1Elem : h1Elements) {
            if (h1Elem.text().isEmpty()){
                h1Elem.appendText(
                    this.getCommonUtils().getMessage(
                        "code."+this.code()+".suggestion.content"));
            }
        }
    } else {
        Element h1Node = new Element(Tag.valueOf("h1"), "");
        h1Node.appendText(this.getCommonUtils().getMessage(
            "code."+this.code()+".suggestion.content"));
        element.appendChild(h1Node);
    }
}
}
}

```

Código Java 4.4.3.2 (cont.): implementación de una regla.

Como puede observarse en el código 4.4.3.2, cada especificación de regla cuenta con un “contexto”, el cual indica el conjunto de elementos del documento HTML sobre el cual debe realizarse la validación. La “severidad”, por otro lado, indica qué tipo de error implica el no cumplimiento de la regla, y puede tomar tres (3) valores: *VIOLATION*, *RECOMMENDATION* y *POTENCIAL VIOLATION*; *VIOLATION* indica que es de carácter obligatorio el cumplimiento de la regla para validar las pautas WCAG 2.0, *RECOMMENDATION* indica que se sugiere el cumplimiento de la regla, y *POTENCIAL VIOLATION* es utilizado cuando una regla no puede ser evaluada programáticamente y requiere la intervención del programador.

El método **break()** es en donde se verifica si el código evaluado cumple con la regla en cuestión o no. Los detalles de su funcionamiento se explican en el inciso 4.4: Implementación de SiMor.

Luego, y por último, cada regla cuenta con un método llamado **suggestElement()**, donde en caso de no cumplirse la regla, se genera una sugerencia válida y relativa al código evaluado.

Otro de los problemas iniciales fue la manera de almacenar los datos utilizados por SiMor. Cada vez que se evalúa un sitio, el código completo es analizado sintácticamente (parsing) y

almacenado en la base de datos, lo que conlleva el manejo de grandes volúmenes de información y una cantidad considerable de escrituras para su posterior lectura.

4.5. Frontend HTML responsive y accesible

Al momento de diseñar la interfaz de la aplicación se puso especial énfasis en hacer de SiMor un buen ejemplo de Web accesible. A raíz de esto se desarrolló una interfaz adaptativa (responsive) a múltiples dispositivos, cumpliendo además con las pautas WCAG 2.0. Esto permite su utilización independientemente de la condición particular de quien la utilice, y el medio utilizado para su visualización.

El maquetado realizado tiene sus bases en Bootstrap²⁸, lo cual garantiza compatibilidad cross-device y soporte a largo plazo por parte del equipo de desarrollo responsable.

Con una sola versión en HTML y CSS se cubren todas las resoluciones de pantalla, es decir que el sitio web creado es optimizado para todo tipo de dispositivos: PC's, tablets, teléfonos móviles, etc. Esto garantiza el uso de SiMor desde cualquier medio, y no obliga su uso bajo ciertas condiciones.

De esta forma se reducen los costos de desarrollo y mantenimiento cuando el diseño de las pantallas es similar entre diferentes dispositivos.

El diseño fluido se hace posible gracias a la introducción de *media queries* en las propiedades de las hoja de estilos (CSS). Las *media queries* son una serie de reglas que se incluyen en los CSS e indican al documento HTML cómo debe comportarse en diferentes resoluciones de pantalla.

Las figuras 4.5.1.a, 4.5.1.b, 4.5.1.c y 4.5.1.d muestran capturas de la aplicación en diferentes dispositivos móviles a fin de ilustrar lo mencionado en esta sección.

²⁸ <http://getbootstrap.com/>

iPad 19:03 100%

Reglas inválidas

ej.: "img[role != presentation]";"img 1.1.1";a 2.4.2"















Etiqueta	Línea	Regla	Acciones
a	393	Sección WCAG 2.1.1 - Regla #095	
table	274	Sección WCAG 1.3.1 - Regla #003	
table	120	Sección WCAG 1.3.1 - Regla #003	
table	706	Sección WCAG 1.3.1 - Regla #003	
table	274	Sección WCAG 1.3.1 - Regla #004	
table	120	Sección WCAG 1.3.1 - Regla #004	
table	706	Sección WCAG 1.3.1 - Regla #004	
td	291	Sección WCAG 1.3.1 - Regla #008	
td	600	Sección WCAG 1.3.1 - Regla #008	
td	571	Sección WCAG 1.3.1 - Regla #008	
td	286	Sección WCAG 1.3.1 - Regla #008	
td	279	Sección WCAG 1.3.1 - Regla #008	
td	660	Sección WCAG 1.3.1 - Regla #008	
td	703	Sección WCAG 1.3.1 - Regla #008	

Figura 4.5.1.a: Captura del listado de reglas inválidas en un iPad 3 Mini.

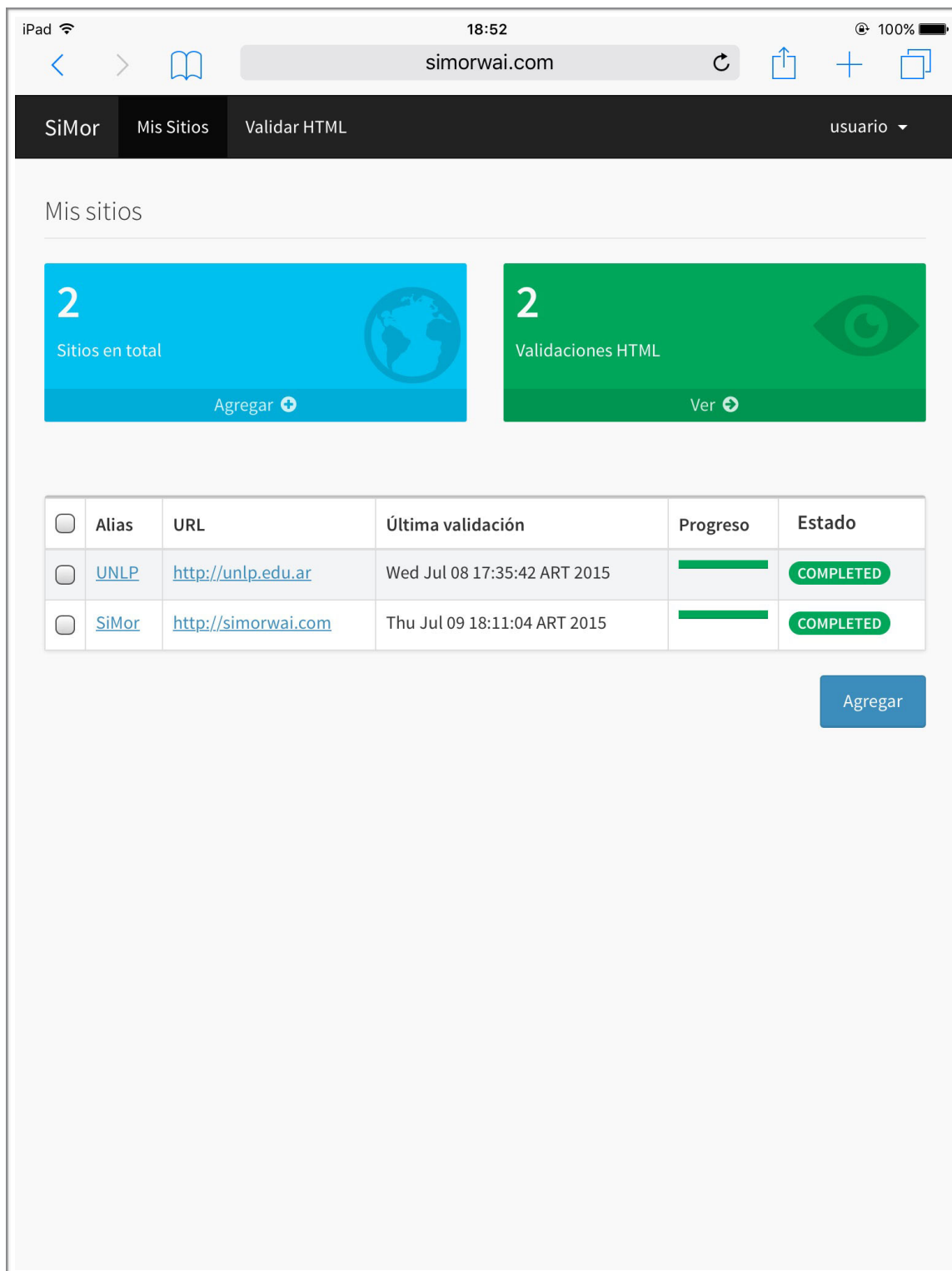


Figura 4.5.1.b: Captura de SiMor desde un iPad 3 Mini.

SiMor Mis Sitios Validar HTML About Contacto usuario

Validar HTML

53 %
Reglas válidas

Ver más

21 %
Reglas con advertencias

Ver más

26 %
Reglas inválidas

Ver más

Tipo de validación

WCAG 2.0 - Nivel A

URL

http://unlp.edu.ar


Código HTML

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <meta name="robots" content="index, follow" />
    <meta name="keywords" content="Universidad Nacional de La Plata,
Institucional, Enseñanza, Investigación, Extensión, Inicio, Pagina,
Inicio, Alumnos, Docentes, Docentes, Graduados, Ingreso, Datos,
Ingresantes, Posgrado, Posgrados, Pregrado, Colegios, sistema,
Preuniversitario, Capacitación, Docente, Capacitación, Docente,
Educación, Distancia, Educación, Distancia, Evaluación, Evaluación,
Articulación, Articulación, Ceprom, Ceprom, Prebi-Sedici, Prebi-Sedici,
Historia, Historia, Gobierno, Gobierno, Cultura, Secretaria, Cultura,
Discapacidad, Discapacidad, Salud, Salud, Sistemas, Unidad, Sistemas,
Becas, Becas, Investigación/Facultades, Secretarías, Facultades,
Vinculación, Tecnológica, Vinculación, Tecnológica, Propiedad,
Intelectual, Propiedad, Intelectual, Servicios, Sociales, Dirección,
Servicios, Sociales, Cátedras, Libres, Cátedras, Libres, Autoridades,
Autoridades, Consejo, Superior, Consejo, Superior, Asistencia,
Municipios, Asuntos, Municipales, Higiene, Seguridad, Taller, Teatro,
Música, Programas/Proyectos, Secretaría, Extensión, Secretaría,
Extensión, Derechos, Humanos, Editorial, Editorial, Biblioteca,
Pública, Biblioteca, Pública, Radio, Universidad, Radio, Universidad,
Comunicación, Visual, Comunicación, Visual, Portal, Portal, Palabra,
Palabra, Universitaria, Salud, sexual, Atención, Personal, Salud,
Personal, Atención, médica, Resume, todos, servicios, atención, médica,
Dirección, Salud, Seguro, Salud, Describe, Servicio, seguro, Público,
Salud, estudiantes, Dirección, Especialización, Especialización,
Docencia, Universitaria, Capacitación, UNLP-ADULP, Capacitación,
Docente, Proyectos, Proyectos, Jóvenes, Investigadores, Jóvenes,
Investigadores. Objetivos. Objetivos. Servicios. Servicios. Normativa.

```

Captura



[Descargar informe](#)

Figura 4.5.1.c: Captura de SiMor desde una computadora personal (laptop) con una resolución FullHD (1920x1080).

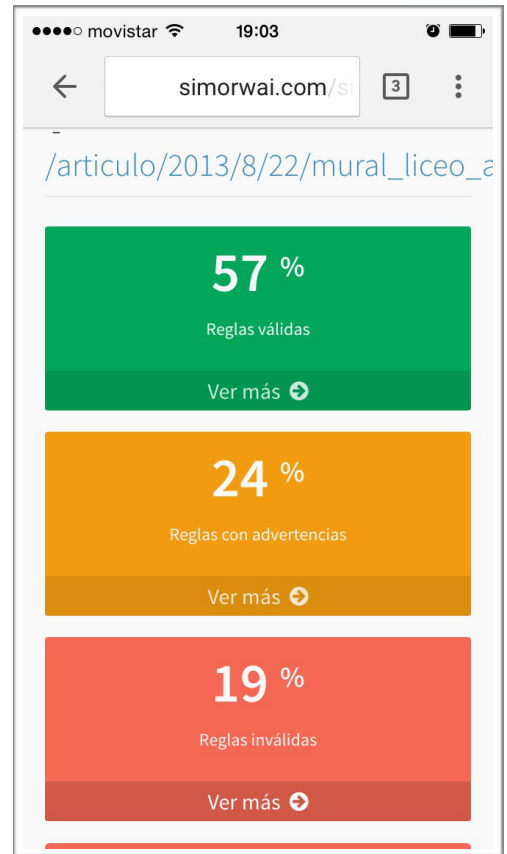


Figura 4.5.1.d: Capturas de SiMor realizadas en un iPhone 5.

4.6. Licencia y disponibilidad

Como se mencionó anteriormente, SiMor es una aplicación de código abierto, publicada bajo la licencia GNU/GPL 3.0, con lo que se tiene una herramienta de libre uso y modificación. Además, se costeó la compra de un nombre de dominio y un servidor web para que la aplicación esté disponible para su uso por parte de la comunidad. El mismo se encuentra publicado en <http://simorwai.com>.

El código fuente se encuentra disponible en <https://github.com/gaspo53/simor.git>, donde cualquier miembro de la comunidad podrá continuar su desarrollo si así lo desea.

4.7. Comparación de SiMor

En la tabla 4.7.1 se presenta una comparación de SiMor con los validadores analizados en el Capítulo 3.

	AChecker	FAE	WAVE	TAW	SiMor
Alcance	Un enlace por vez	Completo (sólo con registración gratuita)	Un enlace por vez	Un enlace por vez	Un enlace por vez o sitio completo
Propuesta Correctiva	Sí	No	Sí	Sí	Sí
Estilo de informe (exportable)	HTML PDF EARL CSV	HTML	HTML	HTML	HTML PDF EARL
Múltiples idiomas	Sí	No	No	Sí	Sí
Omisión de enlaces	No	No	No	No	Sí
Cuenta de usuario con historial	No	No	No	No	Sí
Plataforma	Múltiples plataformas (web)	Múltiples plataformas (web)	Múltiples plataformas (web)	Múltiples plataformas (web)	Múltiples plataformas (web)
Interfaz adaptable	No	No	No	No	Sí
Licencia	GPL	No disponible	No disponible	Privativa	GPLv3

Tabla 4.7.1: Tabla comparativa entre los validadores analizadas y SiMor.

4.8. Conclusión

En base a las características detectadas como necesarias a través de un sondeo de opiniones a profesionales de la materia, se definieron las cualidades requeridas para mejorar y complementar el proceso de validación brindado por los validadores existentes.

Teniendo en cuenta lo mencionado anteriormente, se pueden identificar los siguientes aspectos como mejoras alcanzadas:

- Análisis de accesibilidad de alcance global, en donde no importa el producto o la versión sobre la cual se realizó, sino que se lo analiza directamente con lo que especifica el ente regulatorio en cuestión.
- Las pautas de validación son independientes de la implementación, permitiendo escalabilidad y contingencia a cambios en la especificación.
- Full-scan a partir de una URL; informa la accesibilidad por cada link del sitio.
- La creación de cuentas de usuario, que permite el alojamiento de todos los resultados obtenidos en las validaciones, pudiendo acceder a las mismas tantas veces como sea necesario.
- Sistema de versionado que permite ver la evolución del sitio evaluado en relación a validaciones anteriores.
- Sistema de avisos mediante el cual se notifica al usuario ante eventos de su interés, como por ejemplo la finalización de una validación.
- Detalles completos de validación con información relacionada a reglas y elementos evaluados, cantidad de problemas encontrados, ubicación de los problemas en el código original, y capturas de pantallas de cada link analizado.
- Se proveen sugerencias para resolver problemas existentes.
- Omisión de enlaces: permite la utilización de filtros durante el análisis, pudiendo el usuario indicar qué páginas del sitio no deben ser analizadas, siendo útil a la hora de rever un sitio luego de alguna modificación ya que aumenta la velocidad de análisis, y evita evaluaciones innecesarias.
- Cada link evaluado se renderiza en un navegador, y esta salida es la que SiMor analiza, por lo que no sólo se contempla el código HTML estático, sino que también con javascript ejecutado, por lo que el HTML analizado es el mismo que percibe un usuario final.
- Las validaciones se realizan en background, ejecutándose cada regla en un thread propio, lo cual acelera significativamente el tiempo de ejecución, y permite seguir trabajando con SiMor realizando otras tareas en simultáneo.
- Dispone de una interfaz accesible.
- Emisión de reportes en formato EARL.
- Software de código abierto bajo licencia GPLv3.

A partir de estas características, se puede concluir que se obtuvo una aplicación acorde al concepto de validador buscado en la comunidad, a través de la cual se facilita la generación de contenido web accesible de una manera más sencilla, clara y eficiente.

Como adición, al ser un software de código abierto, cualquier desarrollador puede tener acceso al

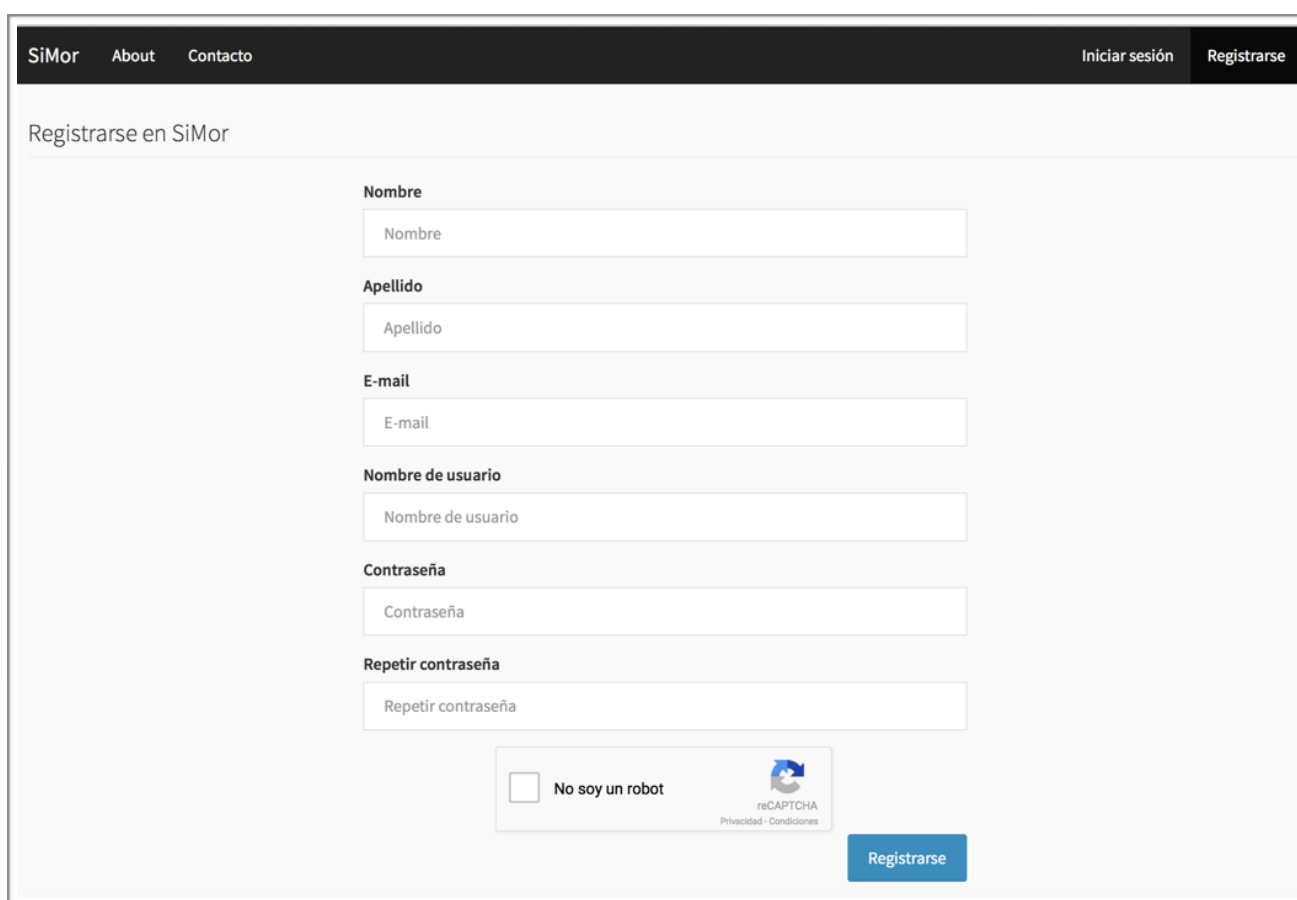
código fuente a fin de modificarlo y/o extenderlo para que se adapte a nuevos estándares, o requerimientos propios de un organismo o ley en particular.

Capítulo 5: Uso de SiMor

En el Capítulo 3 se describieron los resultados de la comparación de las validaciones realizadas por las herramientas de evaluación seleccionadas.

Con el fin de exponer la funcionalidad y características definidas en los objetivos de esta tesina y así poder marcar las diferencias para con los validadores analizados, mencionadas a lo largo de este informe, se detalla a continuación el proceso de validación realizado por SiMor, nuevamente sobre el sitio www.unlp.edu.ar.

Para poder realizar una validación se requiere de un registro previo. Para llevarlo a cabo, es necesario completar un formulario como se indica en la figura 5.1:



The screenshot shows the registration page for SiMor. At the top, there is a navigation bar with 'SiMor', 'About', and 'Contacto' on the left, and 'Iniciar sesión' and 'Registrarse' on the right. The main heading is 'Registrarse en SiMor'. The form contains the following fields:

- Nombre
- Apellido
- E-mail
- Nombre de usuario
- Contraseña
- Repetir contraseña

Below the fields is a reCAPTCHA widget with the text 'No soy un robot' and a 'Registrarse' button.

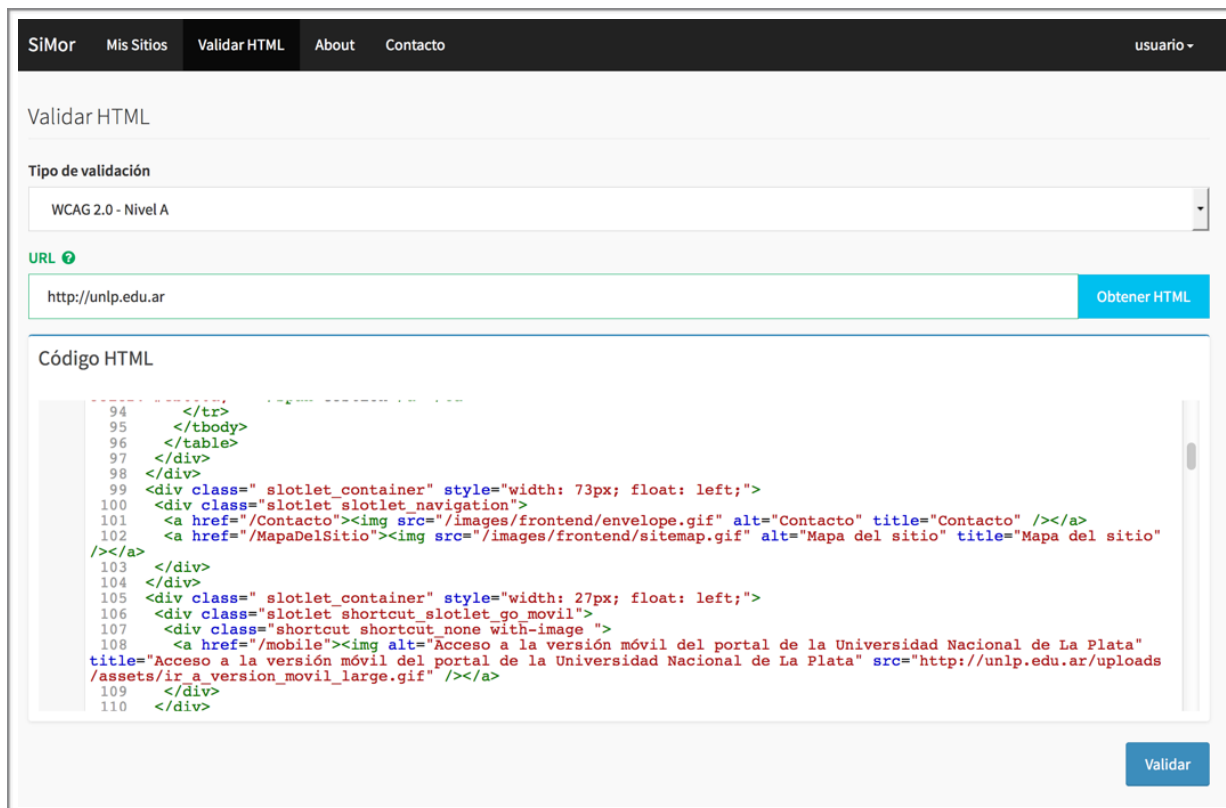
Figura 5.1: Pantalla de registro de SiMor.

Una vez realizado el registro, éste debe ser confirmado mediante el email recibido en la dirección indicada en el formulario; esto permite comenzar a utilizar la aplicación y así poder realizar validaciones. Para esto, antes se debe iniciar sesión con el nombre de usuario y contraseña indicados en el formulario, para acceder a la aplicación.

Como se mencionó anteriormente, SiMor permite dos (2) tipos de validaciones: validar un solo enlace o código HTML aislado, y la validación de un sitio web completo.

5.1. Validación de código HTML

La validación de un solo enlace o código HTML se realiza a través de la opción “Validar HTML”, disponible menú principal de la aplicación, como puede verse en la figura 5.1.1.



The screenshot shows the 'Validar HTML' page in the SiMor application. The navigation bar at the top includes 'SiMor', 'Mis Sitios', 'Validar HTML' (highlighted), 'About', and 'Contacto'. The user is logged in as 'usuario -'. The main content area is titled 'Validar HTML' and contains a dropdown menu for 'Tipo de validación' set to 'WCAG 2.0 - Nivel A'. Below this is a 'URL' input field containing 'http://unlp.edu.ar' and a blue 'Obtener HTML' button. The 'Código HTML' section displays a code editor with the following HTML code:

```
94     </tr>
95   </tbody>
96 </table>
97 </div>
98 </div>
99 <div class=" slotlet_container" style="width: 73px; float: left;">
100 <div class="slotlet_slotlet_navigation">
101   <a href="/Contacto"></a>
102   <a href="/MapaDelSitio"></a>
103 </div>
104 </div>
105 <div class=" slotlet_container" style="width: 27px; float: left;">
106 <div class="slotlet shortcut_slotlet_go_movil">
107   <div class="shortcut shortcut none with-image ">
108     <a href="/mobile"></a>
109 </div>
110 </div>
```

A blue 'Validar' button is located at the bottom right of the code editor area.

Figura 5.1.1: Pantalla de validación HTML.

Antes de realizar una validación HTML se debe seleccionar el tipo de nivel de accesibilidad (A, AA, AAA) en la lista de selección titulada “Tipo de validación”, indicar la URL perteneciente al código, y por último el código mismo. Además, con sólo ingresar la URL, SiMor automáticamente puede obtener dicho HTML si es que el usuario así lo requiere. En ambos casos se corrobora que la URL ingresada tenga un formato correcto y que, además, dicho enlace exista y tenga conectividad; caso contrario se avisa al usuario del problema y no se continúa con el proceso.

Luego de haber completado los datos requeridos, se procede a realizar la validación sobre el código, dando como resultado el detalle de la validación que muestra la figura 5.1.2.

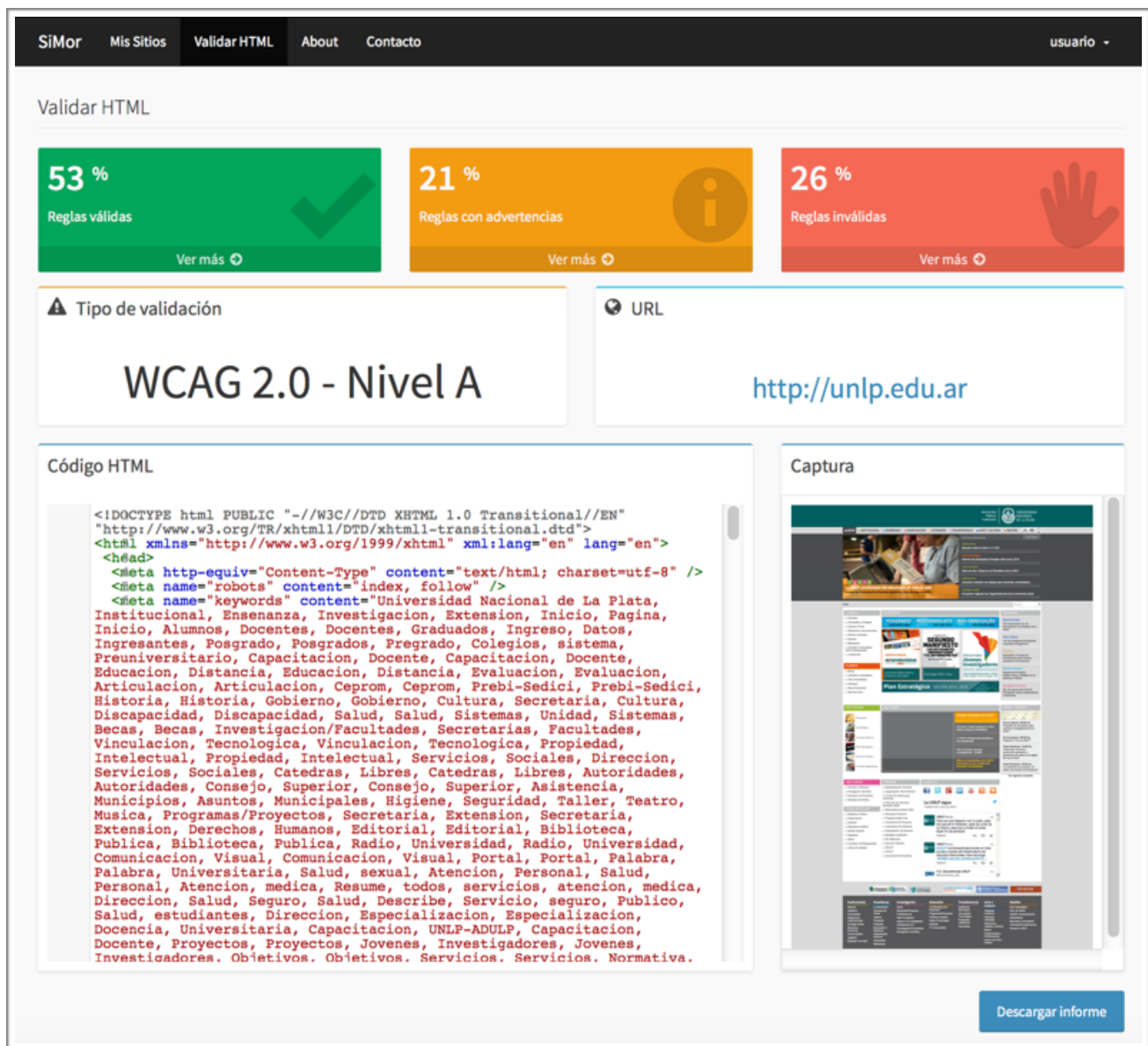


Figura 5.1.2: Pantalla de detalle de validación HTML.

La figura 5.1.2 expone la pantalla de SiMor que muestra la validación realizada, en donde se detalla el índice de cumplimiento contra el nivel de accesibilidad previamente seleccionado, el cual se divide en tres (3) partes conformando un total del 100%:

- Porcentaje de reglas cumplidas, en color verde.
- Porcentaje de reglas con advertencias, en color anaranjado.
- Porcentaje de reglas no cumplidas, en color rojo.

Cada una de estas secciones contiene un enlace que permite acceder al detalle de las reglas evaluadas junto con su resultado, el código evaluado, la sugerencia de corrección en las reglas no cumplidas y la posibilidad de efectuar un **falso positivo** en las reglas con advertencias.

En la figura 5.1.3, se detalla la información inherente a la especificación de la regla, el enlace a la documentación de la sección correspondiente a la W3C, un ejemplo de código válido e inválido y el código HTML analizado.

Etiqueta	Línea	Detalles de la evaluación	Acciones
img	313	<h3>WCAG2 Regla 027</h3> <p>Regla Código HTML</p> <p>Descripción</p> <p>El nombre de archivo de la imán no es válido como texto alternativo</p> <p>Un nombre de archivo, como texto alternativo para una imágen, no estáá permitido. Se debe especificar un texto alternativo significativo, como lo especifica la Regla WCAG2 001.</p> <p>Contexto</p> <p><code>img[role != presentation]</code></p> <p>Ejemplo de código válido</p> <p>Correcto</p> <pre></pre> <p>Ejemplo de código inválido</p> <p>Incorrecto</p> <pre></pre> <p>Necesitas más información?</p> <p>Especificación del requerimiento 1.1.1 de la W3C</p> <p>Cerrar</p>	🔧
a	1050		🔧
a	1142		🔧
a	313		🔧
a	318		🔧
a	323		🔧
img	537		🔧
a	328		🔧
a	1145		🔧
a	333		🔧
a	133		🔧
img	399		🔧
a	135		🔧

Figura 5.1.3: Detalle de regla cumplida.

En el caso de las reglas no cumplidas, existe una solapa denominada “Sugerencia” donde se brinda una porción de código como posible solución al problema encontrado; en este caso el usuario puede integrarla directamente en su código.

Dicha solapa, se accede directamente haciendo click sobre el botón con forma de llave, situado en la columna “Acciones”. Esto puede apreciarse en la figura 5.1.4.

Etiqueta	Línea	Detalles de la evaluación	Acciones
a	133	<p>WCAG2 Regla 040</p> <p>Regla Sugerencia</p> <p>Código fuente original</p> <p>Elemento <a></p> <p>Línea 130</p> <pre><a href="/articulo/2015/7/3/la_unlp_observara_las_elecciones_er</pre> <p>Sugerencia SiMor</p> <pre><a href="/articulo/2015/7/3/la_unlp_observara_las_elecciones_er</pre> <p>Puede que tengas que prestar atención</p> <p>Hacemos nuestro mayor esfuerzo en brindarte la sugerencia más automatizada posible, pero SiMor no es igual a un humano (todavía). Así que por favor, antes de copiar y pegar el código sugerido, sé cauteloso y analiza la solución para ver si se ajusta a tus necesidades.</p> <p>Necesitas más información?</p> <p>Especificación del requerimiento 2.4.4 de la W3C</p> <p>Cerrar</p>	🔧
a	130		🔧
a	1053		🔧
a	138		🔧
a	1194		🔧
a	143		🔧
a	148		🔧
a	81		🔧
a	109		🔧
a	1012		🔧
a	1021		🔧
a	81		🔧
a	94		🔧

Figura 5.1.4: Detalle de regla no cumplida.

Cabe recordar que las reglas con advertencias son aquellas que no pueden ser validadas automáticamente y que, por lo tanto, necesitan de la atención del programador. Como puede observarse en la figura 5.1.5, para estos casos se muestra el código HTML evaluado en la regla, y se da la posibilidad de marcarla como falso positivo, aumentando el porcentaje de reglas cumplidas.

Etiqueta	Línea	Detalles de la evaluación	Acciones
a	361	<p>WCAG2 Regla 057</p> <p>Regla Sugerencia</p> <p>Código fuente original</p> <p>Elemento <object></p> <p>Línea 922</p> <pre><object title="" width="180" height="450" quality="high" plugin</pre> <p>Falta la sugerencia?</p> <p>Esta es una regla que no puede ser evaluada automáticamente, así que tienes que leer los comentarios descritos debajo, y la especificación de la W3C. En caso de que sea una falsa alarma, puedes marcarlo como falso positivo.</p> <p>Necesitas más información?</p> <p>Especificación del requerimiento 1.4.2 de la W3C</p> <p>Cerrar</p>	🔧
a	100		🔧
a	366		🔧
a	101		🔧
a	1062		🔧
a	109		🔧
a	371		🔧
a	1065		🔧
object	922		🔧
a	376		🔧
a	381		🔧
a	601		🔧

Figura 5.1.5: Detalle de una regla con advertencia.

Además de toda la información detallada de las reglas, se muestra el código fuente analizado, una captura de pantalla del enlace provisto y la posibilidad de descargar el informe en formato EARL, incluido en el Anexo, mediante el botón *Descargar informe* que muestra la figura 5.1.6.

The image shows a software interface with two main panels. The left panel, titled 'Código HTML', displays a snippet of HTML code with line numbers 129 through 139. The code includes links to news articles from UNLP, such as 'La UNLP monitoreará las elecciones en todo el país' and 'Día Histórico. Se aprobó y es ley el Boleto Educativo Gratuito'. The right panel, titled 'Captura', shows a screenshot of a website with a header, navigation menu, and several news articles. At the bottom right of the interface, there is a blue button labeled 'Descargar informe'.

Figura 5.1.6: Detalle del código HTML y captura de pantalla.

Cabe destacar que haciendo click en el número de línea del detalle de cada regla evaluada, se muestra resaltado el código HTML correspondiente al elemento evaluado por dicha regla, como se observa en la figura 5.1.7.

The image shows a snippet of HTML code with line numbers 100, 101, and 102. Line 100 is highlighted in red. The highlighted code is: ` Arte y Cultura`. The surrounding code includes other links and spans with different background colors.

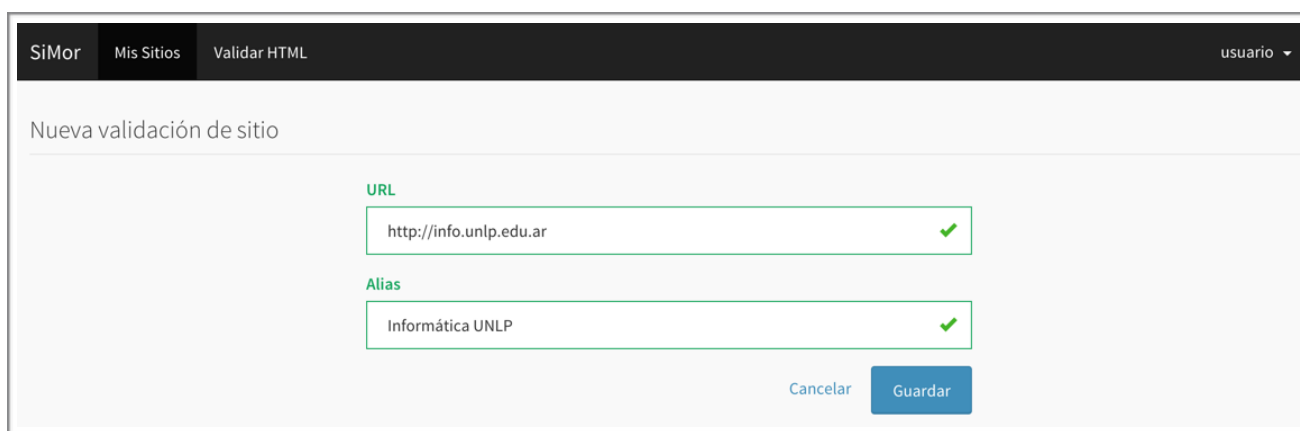
Figura 5.1.7: Resaltado del elemento evaluado por una regla.

5.2. Validación de sitio web

La validación de un sitio web tiene algunas particularidades respecto a la validación HTML, como es el concepto de **Versión**.

Cada vez que se valida un sitio, la aplicación genera una nueva versión del mismo, compuesta por la estructura HTML completa del sitio en ese instante. El objetivo de este procedimiento es el de tener disponible todo el histórico de validaciones que permiten analizar la evolución del sitio en una línea de tiempo.

El acceso a esta funcionalidad se realiza a través de la opción “Mis Sitios”, disponible en el menú principal de la aplicación, como puede verse en la figura 5.2.1.



La imagen muestra una interfaz de usuario con un encabezado oscuro que contiene los botones "SiMor", "Mis Sitios" y "Validar HTML", y un menú de usuario "usuario". El contenido principal está titulado "Nueva validación de sitio" y contiene dos campos de entrada con validación visual:

- Un campo etiquetado "URL" con el valor "http://info.unlp.edu.ar" y un ícono de checkmark verde a la derecha.
- Un campo etiquetado "Alias" con el valor "Informática UNLP" y un ícono de checkmark verde a la derecha.

En la parte inferior derecha del formulario, hay dos botones: "Cancelar" (desactivado) y "Guardar" (activado).

Figura 5.2.1: Agregado de un sitio web para ser validado.

Cabe aclarar que además de la propia URL del sitio a validar se pide el ingreso de una alias, que es un nombre descriptivo, a elección del usuario, para poder identificarlo en el listado general de sitios.

Una vez agregado el sitio, se agrega automáticamente una nueva versión (serie generada comenzando desde el número 1), que es sobre la cual se realiza la validación, como se muestra en la figura 5.2.2:

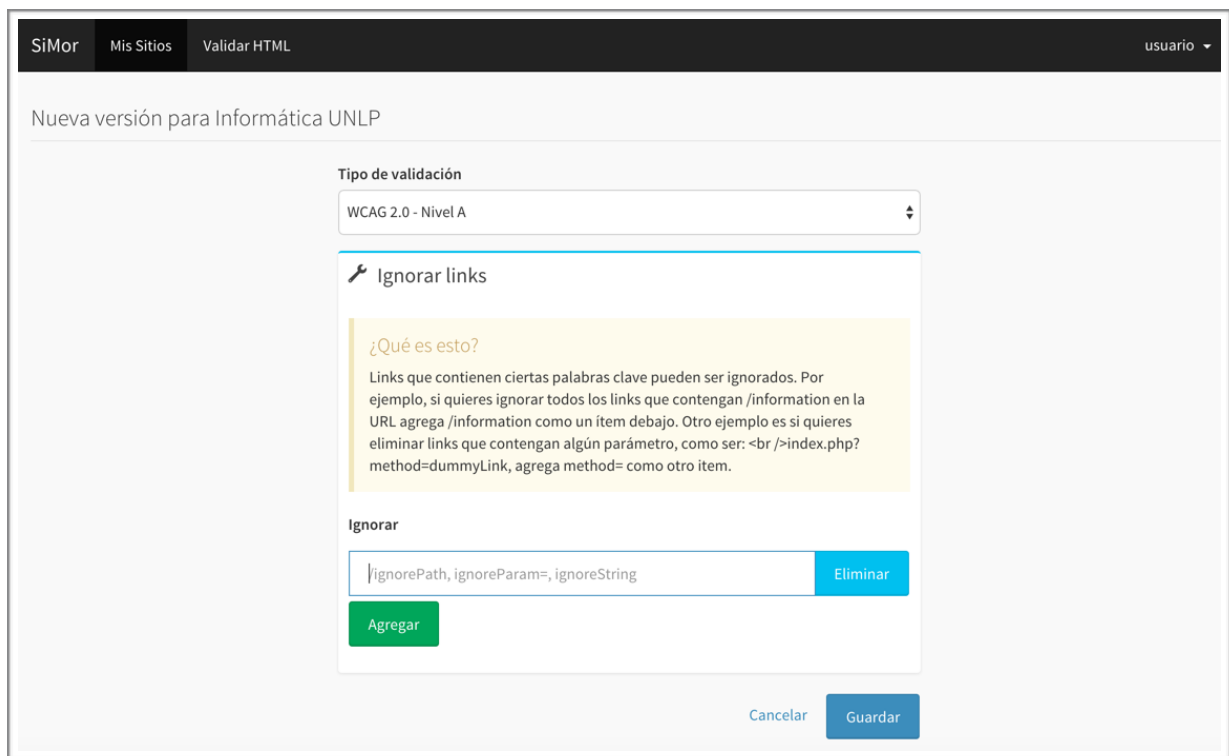


Figura 5.2.2: Agregado de una versión para el sitio web añadido.

En esta sección, se pueden agregar palabras clave para ignorar enlaces cuya URL contenga alguna de esas palabras, lo cual es de suma utilidad para ignorar enlaces que están en el sitio y que no son candidatos para ser validados, o incluso para evitar el volver a validar secciones que ya fueron analizadas previamente y que ya cumplen con el nivel de accesibilidad deseado.

Además, se define el nivel de accesibilidad bajo el cual se realizará la validación (A, AA y AAA). Una vez establecidos estos parámetros, se procede al análisis y validación sobre la versión del sitio recién agregada, retornando a la pantalla del listado de validaciones del usuario, que se observa en la figura 5.2.3.

En la figura expuesta, pueden notarse dos columnas particulares, que son las de Progreso y Estado, que muestran el porcentaje alcanzado por la validación del sitio (si es que hay alguna en curso).

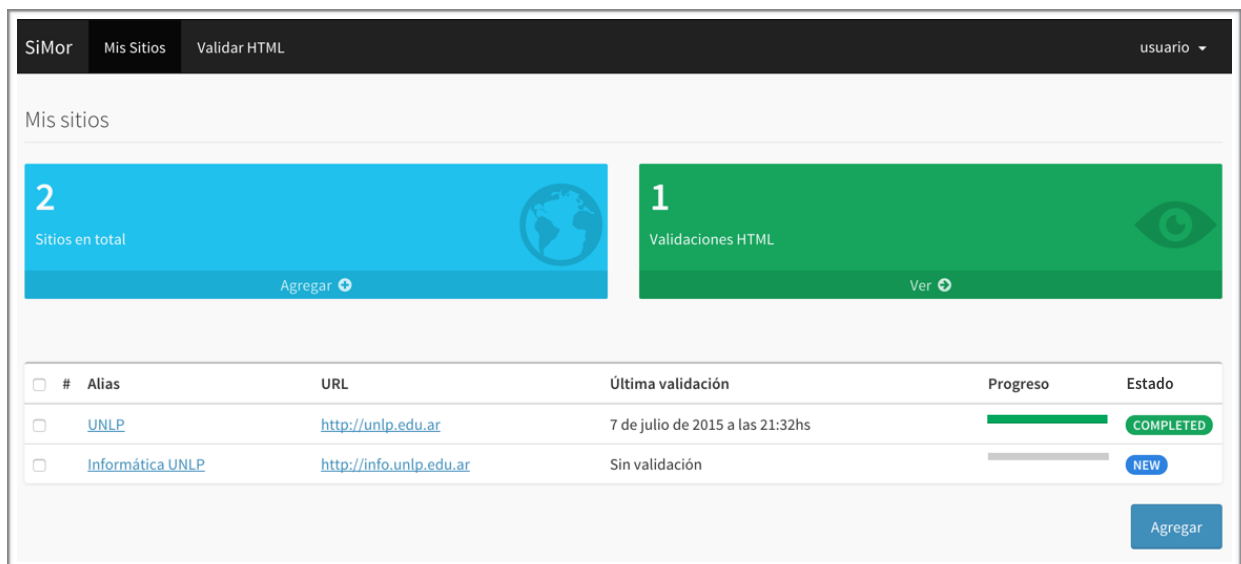


Figura 5.2.3: Listado y estado de validaciones de sitios.

Una vez realizada la validación, se puede acceder al detalle del sitio y ver el resultado de la última versión, y/o de cualquiera de las anteriores (figuras 5.2.4.a y 5.2.4.b).

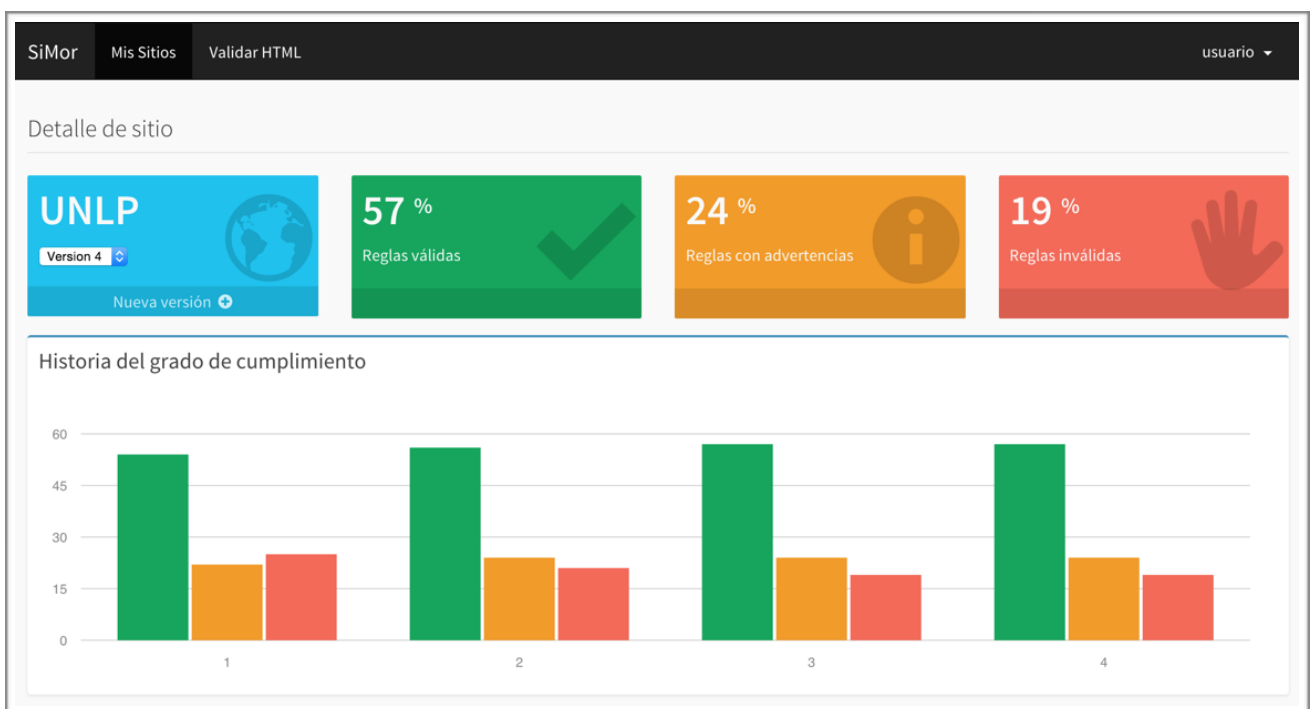


Figura 5.2.4.a: Detalle de una validación de sitio.

La figura 5.2.4.a muestra el detalle de la validación de un sitio web en la que se presenta la siguiente información:

Encabezado principal: los cuatro (4) recuadros indican el sitio analizado y su versión actual (con la posibilidad de cambiar a otra), el porcentaje total de reglas válidas, reglas con advertencias y reglas inválidas, en ese orden.

Historia del grado de cumplimiento: este gráfico indica la evolución de los resultados de las validaciones realizadas a cada versión (indicada al pie de cada uno) en cuanto a reglas válidas (verde), con advertencias (amarillo) e inválidas (rojo).

Enlaces del sitio		
#	URL	Elementos analizados
1	/feed/news/section/plan_estragico	69% 8% 23%
2	/feed/news/section/bienestar	78% 4% 17%
13	/articulo/2011/11/24/pensar_la_unlp_conclusiones	58% 26% 16%
14	/articulo/2013/9/25/jornada_pregrado_plan_estragico_2014_2018_septiembre_de_2013	58% 24% 18%
15	/articulo/2015/6/26/convocatoria_abierta_para_estudiantes_guias	57% 24% 19%
16	/articulo/2015/4/29/concurso_no_docente_en_la_facultad_de_periodismo_y_comunicacion_social	57% 25% 18%
17	/articulo/2015/3/25/convocatoria_a_concurso_no_docente_en_ingenieria	58% 25% 17%
18	/articulo/2015/4/14/concurso_no_docente_en_la_facultad_de_periodismo	58% 26% 17%
19	/articulo/2008/7/17/informe_de_actividades_trimestrales_plan_estragico_de_gestion_2014_2018	57% 24% 18%
20	/articulo/2015/6/15/reunion_comite_ejecutivo_cin_junio_2015	56% 24% 20%

1 2 » »

[Volver a Mis sitios](#)
[Descargar informe](#)

Figura 5.2.4.b: Detalle de una validación de sitio.

Esta figura muestra los enlaces validados. Este listado representa la totalidad de enlaces evaluados, junto con el porcentaje el elementos analizados para cada tipo de regla. En cada enlace se tiene la posibilidad de ir hacia la dirección web real, o de ver el detalle de la validación del enlace (la información detallada en esta sección es la misma que la provista en la figura 5.1.2).

5.3. Auto evaluación de SiMor

Para fundamentar la accesibilidad de SiMor, y luego de haber dado a conocer el funcionamiento de la aplicación, en las figuras 5.3.1.a y 5.3.1.b se detallan los resultados de la validación aplicados a la herramienta a través de sí misma.

Como puede observarse en la figura 5.3.1.a SiMor cumple con todas las reglas WCAG 2.0 evaluadas, quedando sólo revisar las reglas con advertencias, que son las que requieren intervención humana.

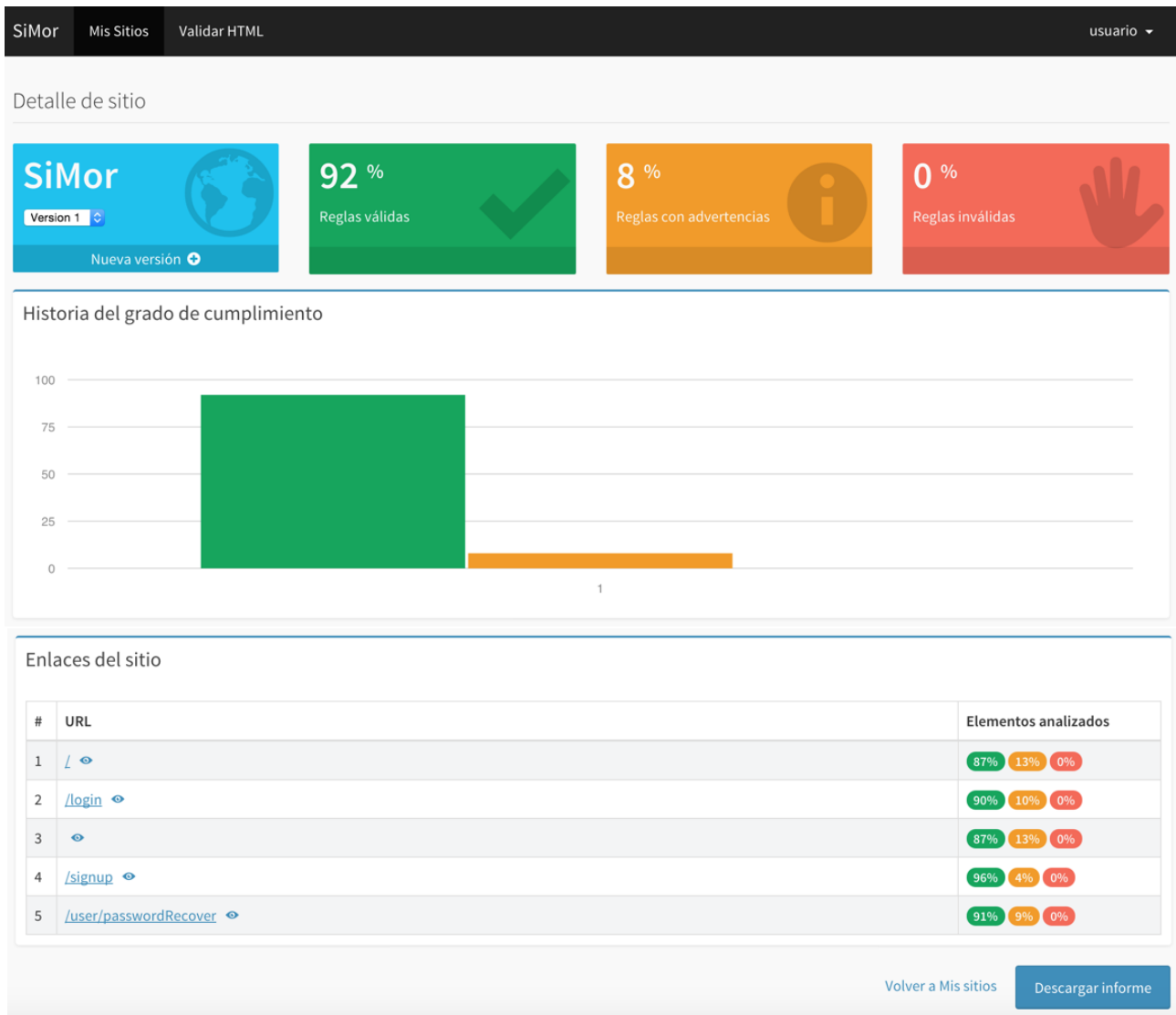


Figura 5.3.1.a: Detalle de una validación de SiMor sobre sí mismo.

La figura 5.3.1.b expone el detalle de validación sobre un enlace particular de SiMor, en la que se puede ver que sólo hay un 4% de reglas con advertencias, que son las que se deben revisar mediante intervención humana.

SiMor Mis Sitios Validar HTML
usuario ▾

Detalle de validación de enlace - /signup

96 %

Reglas válidas

Ver más ↕

4 %

Reglas con advertencias

Ver más ↕

⚠ Tipo de validación

WCAG 2.0 - Nivel A

URL

http://simorwai.com/signup

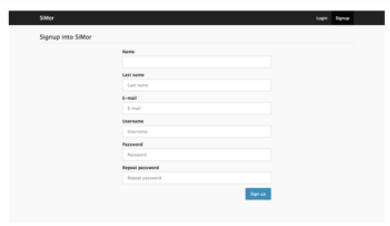
Código HTML

```

<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" lang="en">
<head>
  <title>Signup into SiMor - SiMor</title>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0" />
  <meta name="description" content="" />
  <meta name="author" content="" />
  <link rel="shortcut icon" type="image/x-icon" href="/static/favicon.ico" />
</head>
<script>
  11 var FALSE_POSITIVE_CONFIRM = 'Do you want to mark this validation as
  a False Positive? ';
  12 var YES = 'Yes';
  13 var NO = 'No';
</script>
<script src="https://www.google.com/recaptcha/api.js"></script>
<link rel="stylesheet" type="text/css" media="all"
href="/static/css/application.min.css" />
<script src="/static/js/application.min.js"></script>
<style type="text/css"></style>
<meta name="menu" content="user_signup" />
</head>
<body>
  24 <div class="navbar navbar-default" role="navigation">
  23 <div class="container">
  24 <div class="navbar-header">
  25 <button class="navbar-toggle" type="button" data-toggle="collapse"
data-target=".navbar-collapse"> <span class="sr-only">Toggle
navigation</span> <span class="icon-bar"></span> <span class="icon-bar">
</span> <span class="icon-bar"></span> </button>
  26 <a class="navbar-brand" href="/">SiMor</a>

```

Captura



[Volver a SiMor](#)
[Descargar informe](#)

Figura 5.3.1.b: Detalle de la validación de un enlace en particular de SiMor.

5.4. Evaluación de SiMor con WAVE

Dado que WAVE es el validador más popular de los que se estudiaron en esta tesina, según muestran los resultados de la encuesta realizada (en cuanto al uso de validadores) como se muestra en el gráfico 5.4.1, fue el elegido para analizar la accesibilidad de SiMor.

Además de la autoevaluación de SiMor, se realizó una evaluación al mismo enlace expuesto en las figuras 5.3.1.a y 5.3.1.b con el validador WAVE.

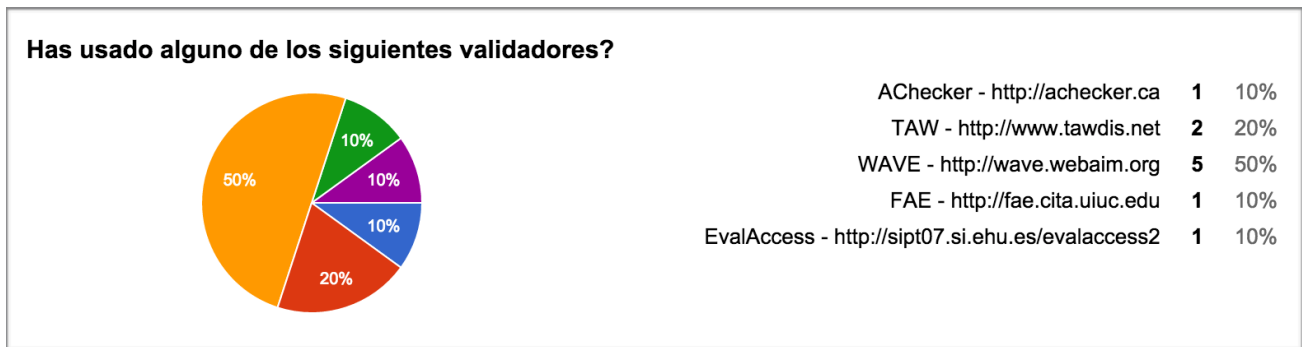


Gráfico 5.4.1: Validadores más utilizados según encuesta realizada.

Como se puede observar en la figura 5.4.2, WAVE no reporta reglas inválidas, ni reglas que requieren intervención humana (reglas con advertencias). Cabe aclarar que este último tipo de reglas son subjetivas, y como se explicó anteriormente en el Capítulo 3, todos los validadores interpretan a éstas de manera distinta. Para aclarar este escenario, SiMor muestra un 4% de reglas con advertencias en su validación, mientras que WAVE no identifica ninguna regla de este tipo.

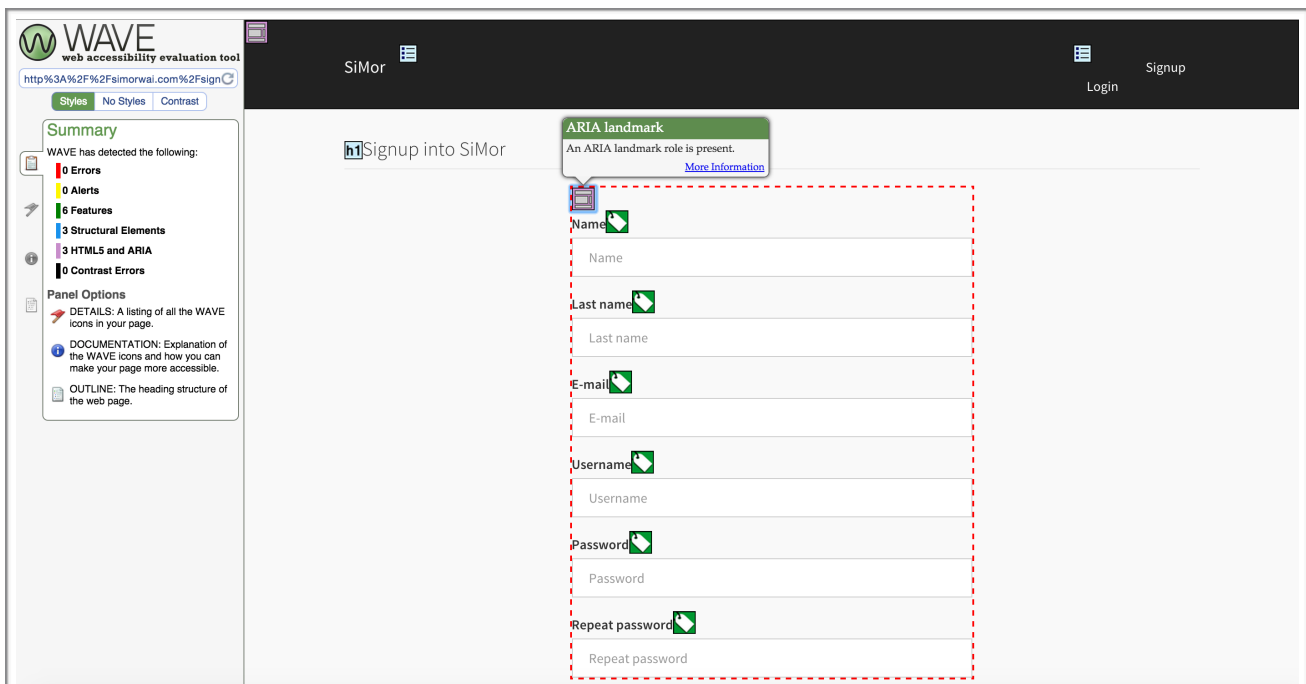


Figura 5.4.2: Validación de SiMor mediante el validador WAVE.

Dado que WAVE no dispone de un informe EARL, que es el que propone la W3C como estándar, no es posible realizar una comparación objetiva entre los resultados de SiMor y los de WAVE.

La figura 5.4.3 muestra el detalle de las reglas con advertencias detectadas mediante la auto-evaluación de SiMor. Las reglas #72²⁹ y #44³⁰ son, en este caso, las consideradas como problemas potenciales por el nivel A de las WCAG 2.0, según la W3C. Al realizar un contraste con la validación realizada por WAVE, expuesta en la figura 5.4.2, puede observarse que este último no detecta dichas reglas durante el proceso. Esto confirma la hipótesis de que los validadores analizados no cumplen en su totalidad los estándares de validación propuestos por la W3C.

Reglas con advertencias			
Etiqueta	Línea	Regla	Acciones
body	77	Sección WCAG 1.3.3 - Regla #072	
a	32	Sección WCAG 2.4.4 - Regla #044	
a	33	Sección WCAG 2.4.4 - Regla #044	
a	26	Sección WCAG 2.4.4 - Regla #044	
Etiqueta	Línea	Regla	Acciones

Figura 5.4.3. : Detalle de reglas con advertencias de la validación de SiMor.

5.5. Evaluación de la accesibilidad de los demás validadores con SiMor

Utilizando SiMor como analizador de accesibilidad, se muestran los resultados obtenidos luego de validar cada uno de las herramientas estudiadas en cuanto al nivel de accesibilidad que cada una posee.

La figura 5.5.1 muestra la validación realizada a aChecker por SiMor.

²⁹ Las instrucciones y funcionamiento deberán ser accesibles y no depender únicamente de color, tamaño, ubicación o sonido.

³⁰ El propósito del link debe estar determinado por el contexto .

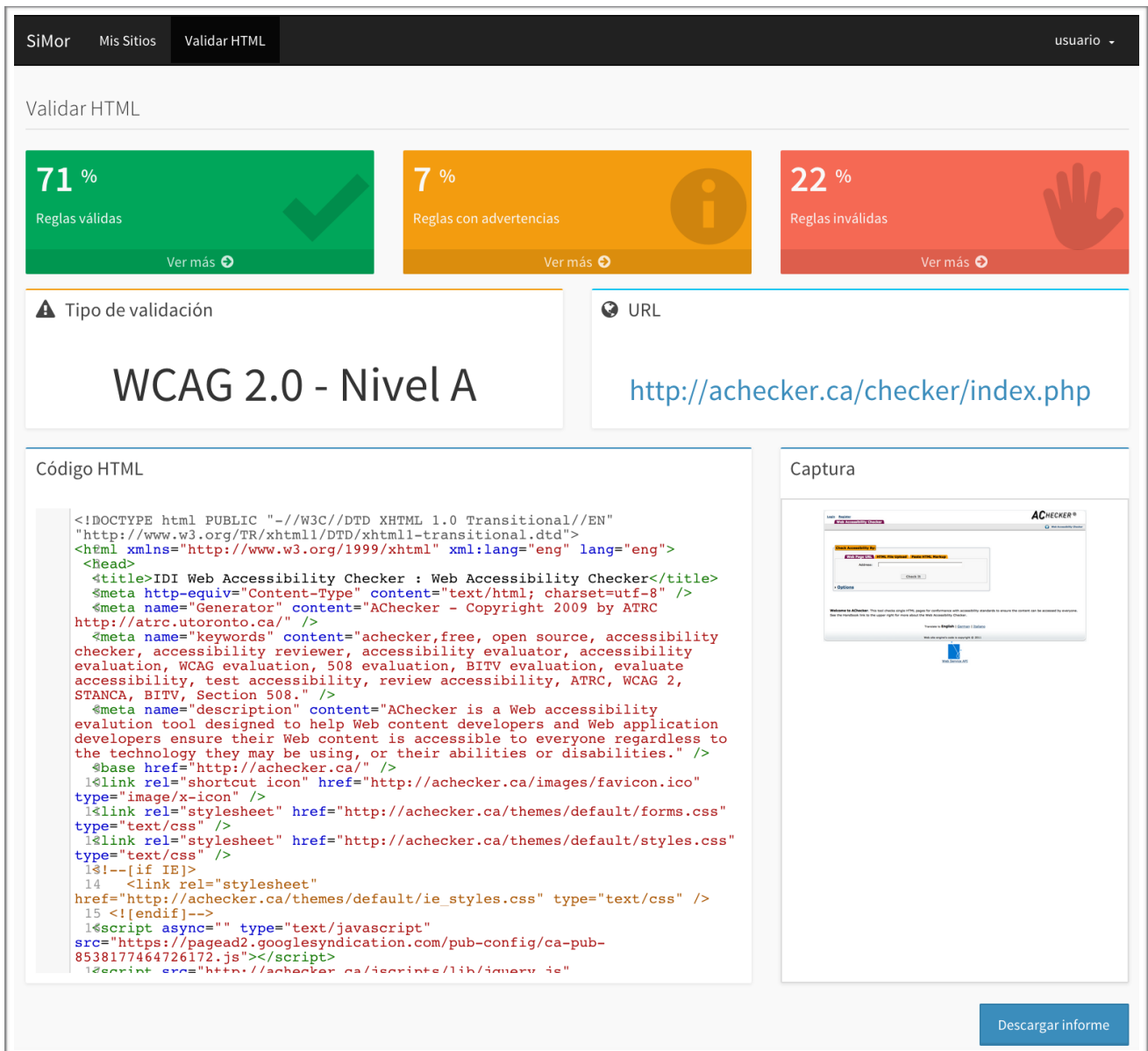


Figura 5.5.1. : Resultado de validación de accesibilidad de aChecker, realizada mediante SiMor.

Como puede verse, aChecker no cumple con el Nivel A de las WCAG 2.0 ya que contiene un 22% de reglas inválidas. Esto significa que la aplicación no garantiza el acceso universal.

A continuación, en la figura 5.5.2 se muestra la validación realizada a FAE por SiMor.

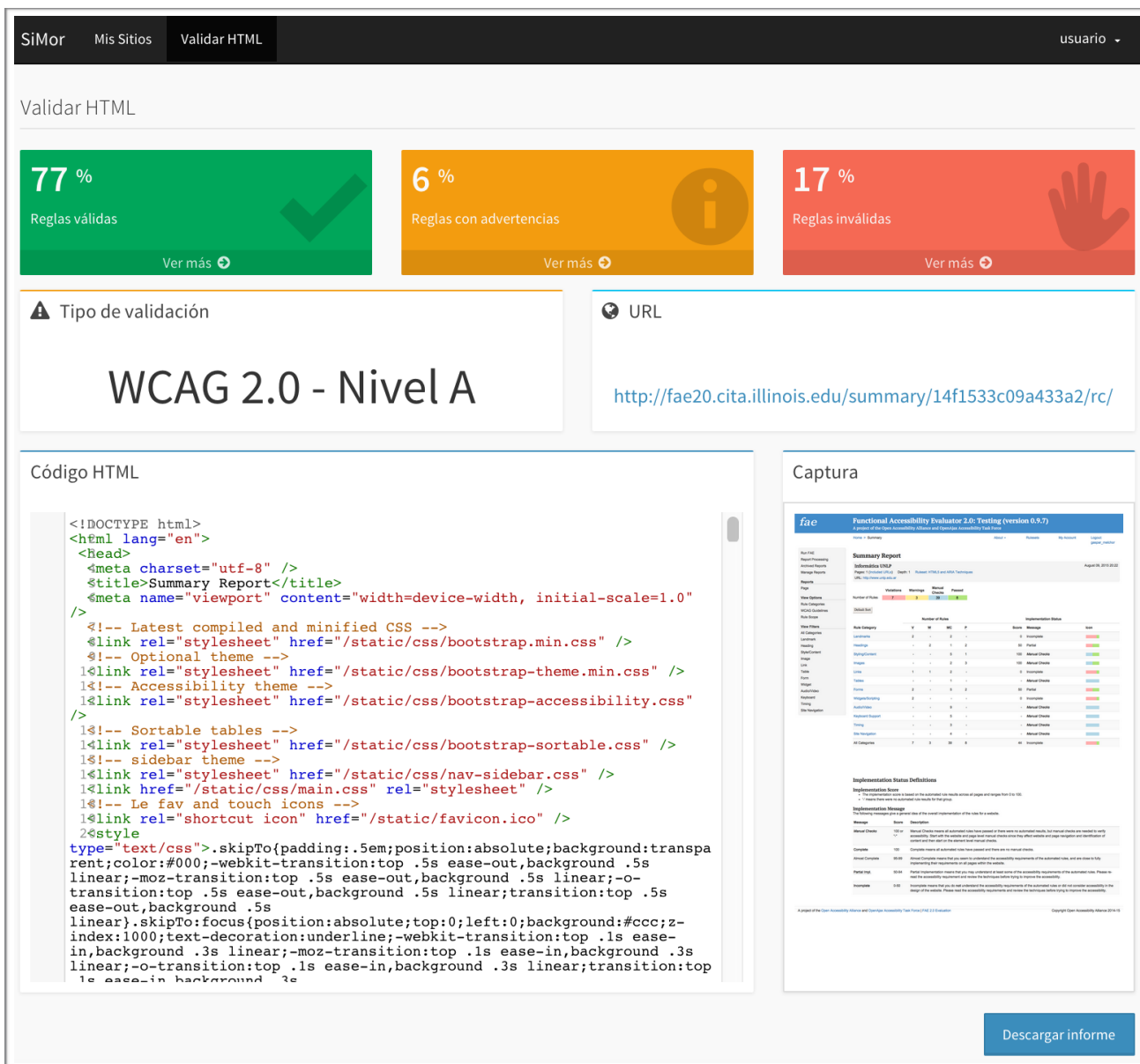


Figura 5.5.2. : Resultado de validación de accesibilidad de FAE, realizada mediante SiMor.

Como puede verse, FAE no cumple con el Nivel A de las WCAG 2.0 ya que contiene un 17% de reglas inválidas. Esto significa que esta aplicación tampoco garantiza el acceso universal.

A continuación, en la figura 5.5.3, se muestra la validación realizada a WAVE por SiMor.

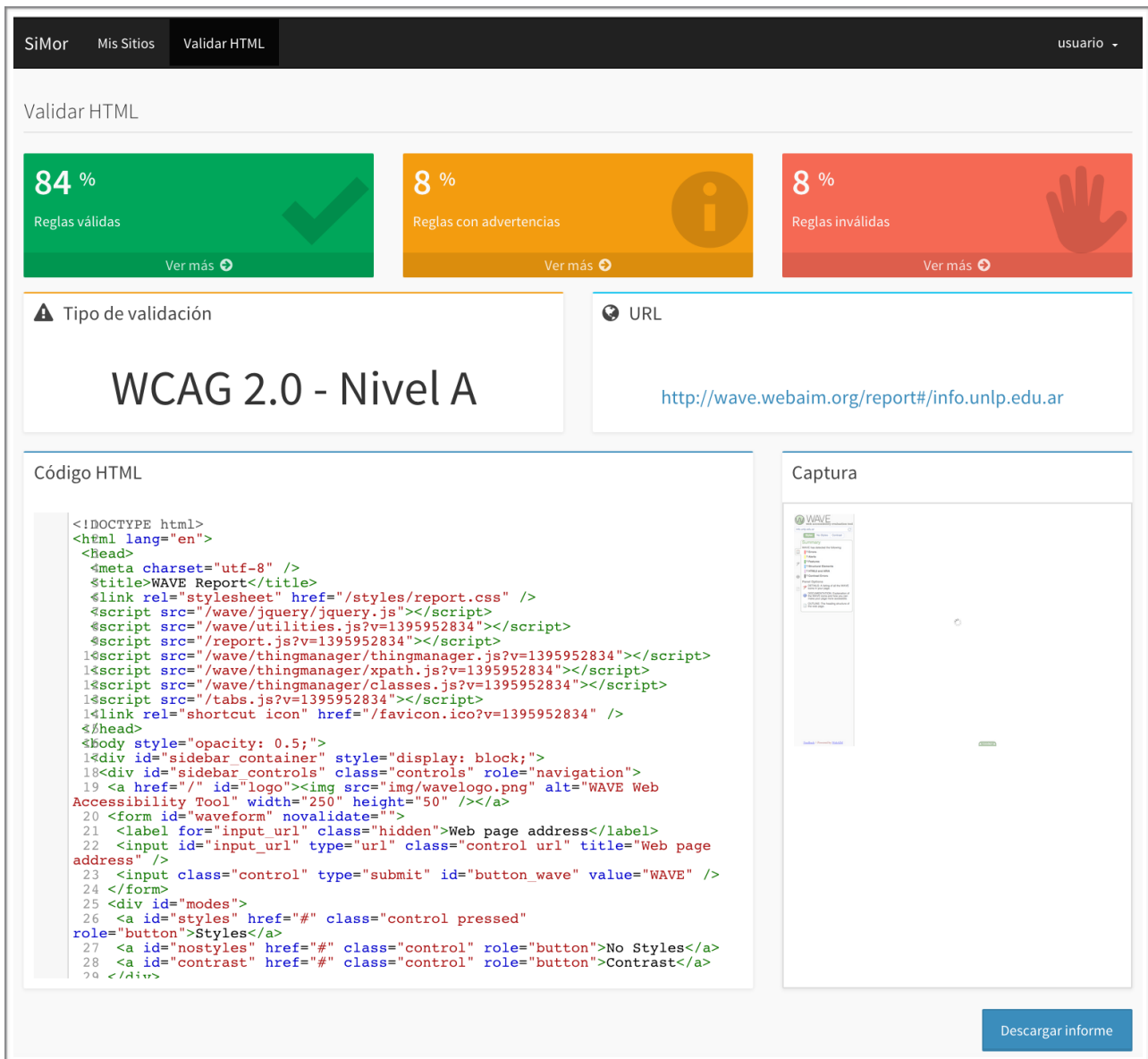


Figura 5.5.3. : Resultado de validación de accesibilidad de WAVE, realizada mediante SiMor.

Como puede verse, WAVE no cumple con el Nivel A de las WCAG 2.0 ya que contiene un 8% de reglas inválidas. Esto significa que la aplicación no garantiza el acceso universal, al igual que las anteriores.

La figura 5.5.4, a continuación, muestra la validación realizada a TAW por SiMor.

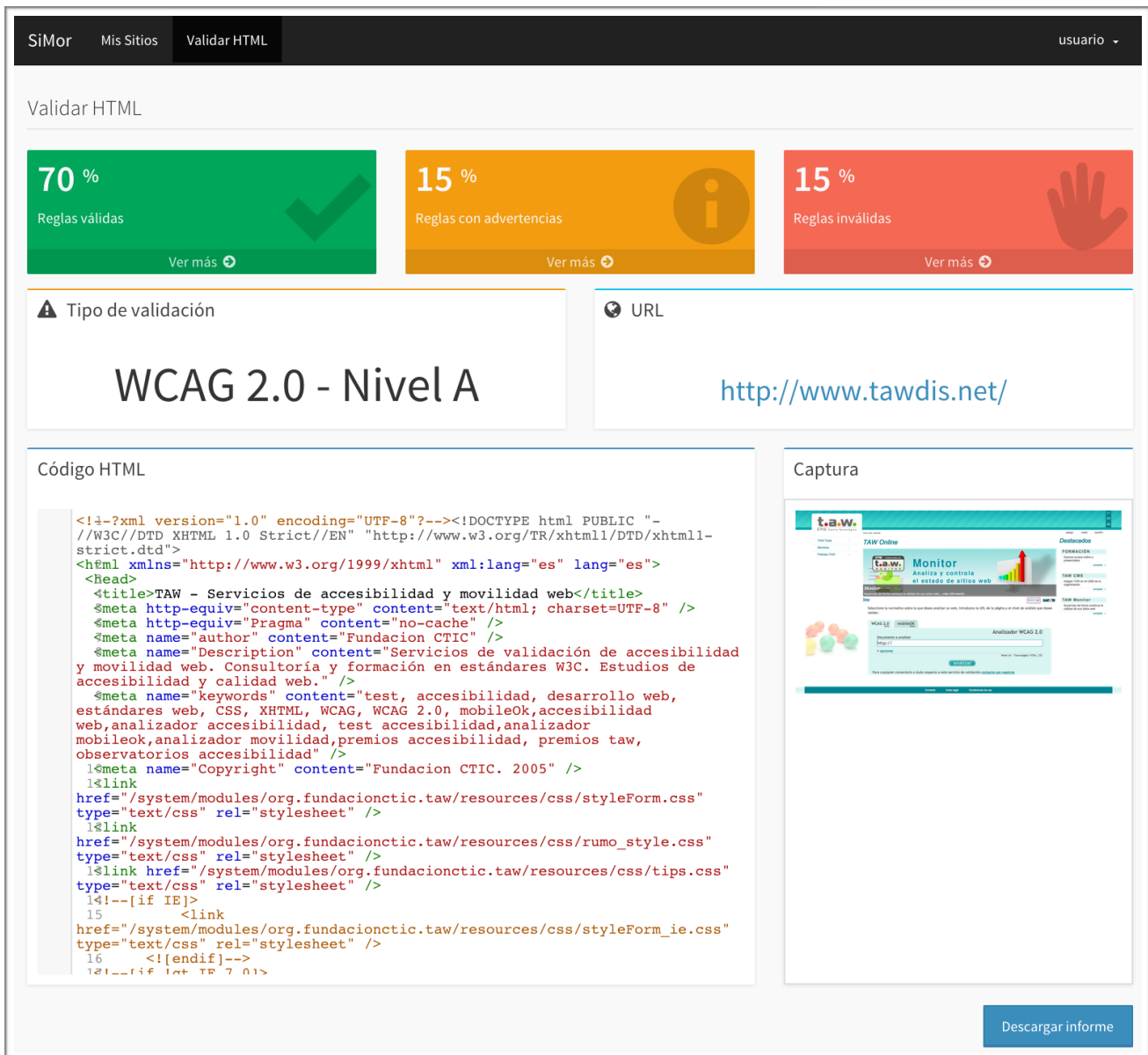


Figura 5.5.4. : Resultado de validación de accesibilidad de TAW, realizada mediante SiMor.

Como puede verse, TAW no cumple con el Nivel A de las WCAG 2.0 ya que contiene un 15% de reglas inválidas. Esto significa que la aplicación no garantiza el acceso universal.

5.6. Conclusión

A lo largo de este capítulo se explicó detalladamente el funcionamiento de la aplicación, y se mostraron las variantes de validaciones que se pueden realizar. Como puede verse en las figuras de capturas de SiMor, y en comparación con las herramientas analizadas en el Capítulo 3, puede notarse la mejora a nivel GUI, usabilidad y accesibilidad alcanzada por la herramienta desarrollada. En cuanto a la accesibilidad de la aplicación, analizada por la propia herramienta y por el validador WAVE, puede observarse de que ambos resultados no mostraron reglas inválidas, por lo cual se demuestra que se cumple el nivel mínimo de accesibilidad exigido por la W3C; siendo el único, respecto de los validadores estudiados, que es totalmente accesible.

Capítulo 6: Conclusiones

El objetivo principal planteado en esta tesina fue el de disponer de una herramienta de validación de accesibilidad web que integre y mejore las funcionalidades de productos similares y que, además, cumpla con una serie de requisitos, que los desarrolladores creen necesarios a la hora de hacer uso de una herramienta de validación.

Habiendo analizado en detalle las funcionalidades ofrecidas por los validadores actuales, y habiendo consultado el listado oficial de validadores reconocidos por la W3C [Eric Eggert and Shadi Abou-Zahra - W3C, 2014] que provee una búsqueda de validadores en base a filtros por características, se verificó que ninguno de los validadores estudiados contemplan todas las características que son de gran utilidad para el desarrollador web y que han sido incluidas en SiMor.

Por otra parte, cabe destacar que durante la etapa de investigación del proyecto se realizaron una serie de encuestas, diseñadas de manera tal que permitieron la obtención de información inherente a los intereses del estudio del proyecto. Una vez finalizadas dichas encuestas, se procedió a su publicación en los principales foros que conciernen a la accesibilidad web, lo que nos dio muestra de la resistencia de las personas de esta comunidad a aceptar nuevas propuestas y/o a adoptar nuevas herramientas que sugieren notables mejoras a las tecnologías existentes; característica que presentó un gran desafío a la hora de intercambiar opiniones y realizar consultoría profesional. Esto, sumado al trabajo de ingeniería efectuado para especificar el proceso de desarrollo de SiMor, permitió compensar las falencias encontradas; lo que pudo ser ratificado a través de las pruebas realizadas una vez concluida la etapa de desarrollo.

En base a lo expuesto, se puede notar que los objetivos propuestos para esta tesina fueron satisfactoriamente alcanzados al obtener como resultado una herramienta que no sólo cumple con la validación estándar ofrecida por los demás validadores, sino que además propone y lleva a cabo un enfoque distinto y más integral en cuanto a la típica *validación - resultado*, donde el usuario es parte del proceso e infiere en dicho resultado a través del empleo de su criterio sensible al contexto particular del contenido evaluado. Esto elimina el resultado muchas veces estricto, producido por las validaciones rígidas tradicionales, perjudicando en muchos casos la dinámica de trabajo que conlleva el cumplimiento de las pautas de accesibilidad que dependen meramente de la subjetividad del usuario.

Por otro lado, y sin restarle importancia, como se mencionó anteriormente se puede verificar la accesibilidad de SiMor a través de la validación a sí mismo, pudiendo concluir que se cumplen todas las reglas WCAG 2.0 descritas anteriormente.

Además de la importancia de los aspectos técnicos de la herramienta, que permiten que su uso sea más simple e intuitivo, se destaca que su desarrollo permite la generación de contenido accesible, donde el acceso al mismo no es limitado por los dispositivos utilizados o las capacidades del usuario; un aporte indudable en materia de inclusión social.

Habiendo concluido con la descripción del trabajo realizado, se procede a definir las actividades a realizar que garantizan la continuidad de este proyecto. Al tratarse de una aplicación de análisis de

sitios web, se debe realizar una revisión periódica de las tecnologías soportadas, los estándares actuales, tanto web como de accesibilidad, los nuevos dispositivos añadidos al mercado, entre otros puntos que condicionan al correcto funcionamiento de SiMor.

Haciendo hincapié una vez más en la libre disponibilidad del código fuente de la aplicación, cabe destacar que el proceso de desarrollo de la misma se fundó en esta premisa, con el objetivo de generar un código fuente claro, bien modularizado, estructurado, bien comentado y cumpliendo los estándares de código propuestos por la plataforma; lo que pretende fomentar la elección de esta herramienta como base para futuros desarrollos, en lugar de crear una nueva.

En lo que respecta a reglas WCAG 2.0, se revisarán los documentos que, a la fecha de presentación de esta tesina, no se encuentren en estado final, para verificar si han sufrido modificaciones sustanciales que afecten a la implementación corriente. Además, dicho proceso de revisión periódico deberá incluir las nuevas reglas (en caso de existir) especificadas en las WCAG.

Se propone una revisión tecnológica con una frecuencia de tres (3) meses para detectar errores, problemas de performance, características sugeridas por los usuarios, etc.

En lo que respecta a usos académicos, se piensa hacer uso de SiMor en las materias Proyecto de Software y Diseño Centrado en el Usuario para que alumnos y docentes de las carreras dictadas en esta facultad puedan incluir la aplicación a la lista de herramientas utilizadas.

Encuesta realizada a colegas

Validadores de Accesibilidad

Por favor, comparta su experiencia con Validadores de Accesibilidad Web

*Required

¿Ha usado alguno de estos validadores? *

Si ha usado más de uno de los que figuran en la lista, elija el más conveniente

- AChecker - <http://achecker.ca>
- TAW - <http://www.tawdis.net>
- WAVE - <http://wave.webaim.org>
- FAE - <http://fae.cita.uiuc.edu>
- EvalAccess - <http://sipt07.si.ehu.es/evalaccess2>

¿Qué es lo que más le gusta de este validador? *

- La correctitud de la validación
- Los tips de ayuda provistos
- La explicación de las reglas violadas
- El diseño/usabilidad de la aplicación
- Muestra los errores encontrados en el HTML analizado
- La velocidad del proceso de validación
- Todo el proceso de validación
- Other:

¿Qué es lo que NO le gusta de este validador? *

- La correctitud de la validación
- Los tips de ayuda provistos
- La explicación de las reglas violadas
- El diseño/usabilidad de la aplicación
- Muestra los errores encontrados en el HTML analizado
- La velocidad del proceso de validación
- Todo el proceso de validación
- Validar cada HTML por separado
- Other:

Como desarrollador, ¿Qué características le gustaría que tenga (además de las que tiene)? *

- Cuentas de usuario, para mantener mis validaciones organizadas
- Análisis de un sitio completo, con todos sus enlaces
- Sugerencias de código para arreglar problemas
- Reportes exhaustivos
- Aplicación Web
- Aplicación Desktop
- Provea de una certificación mundial de Accesibilidad (que se base en W3C)
- Other:

¿Por qué le importa la Accesibilidad Web? *

- Porque mi aplicación está enfocada hacia gente discapacitada
- Porque pienso que la Web es para todos, sin importar las condiciones individuales
- Porque me importan las personas con discapacidades
- Todo lo anterior
- Vendo a través de internet; cuanta más gente lo pueda usar, mayores serán las ventas
- Porque en mi trabajo me obligan
- Other:

Si HOY tuviera que recomendar un Validador de Accesibilidad, ¿Cuál sería? *

- AChecker
- TAW
- WAVE
- FAE
- EvalAccess
- Other:

Describa, separadas por coma, las características que le serían de gran importancia en un validador de accesibilidad *

Ej.: simpleza, velocidad...

La característica de Oro *

Describa cuál sería para Ud. la característica más importante que debería tener un validador de accesibilidad

Pagaría por un servicio serio y completo, que brinde Validación y consultoría sobre Accesibilidad Web? *

- Sí
- No
- Depende del servicio provisto

Indique validadores que haya usado y no figuren en las opciones del punto 1 junto a una breve opinión de cada uno

Submit

Extracto de reporte EARL

```
<rdf:RDF
  xmlns:earl="http://www.w3.org/ns/earl#"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"

  <!-- Assertor -->
  <earl:Software rdf:about="">
    <dct:title xml:lang="en">SiMor - Validador de Accesibilidad Web</dct:title>
    <dct:hasVersion>1.4</dct:hasVersion>
    <dct:description xml:lang="en">
      SiMor es un validador de accesibilidad web basado en reglas.
    </dct:description>
  </earl:Software>

  <!-- Test Subject -->
  <earl:TestSubject rdf:about="http://unlp.edu.ar">
    <dct:title xml:lang="en">Inicio - Universidad Nacional de La Plata (UNLP)</dct:title>
    <dct:date>2015-09-01</dct:date>
  </earl:TestSubject>

  <!-- Test Criterion -->
  <earl:TestRequirement rdf:about="http://www.w3.org/TR/WCAG20/#a">
    <dct:title xml:lang="en">WCAG2-AA</dct:title>
    <dct:description xml:lang="en">Web Content Accessibility Guidelines (WCAG), Version 2.0, Level AA</dct:description>
  </earl:TestRequirement>

  <!-- Test Result -->
  <!-- ===== Known problems ===== -->
  <earl:TestResult rdf:ID="result_known1">
    <earl:pointer rdf:resource="#pointer_known1_line" />
    <earl:pointer rdf:resource="#pointer_known1_error" />
    <earl:pointer rdf:resource="#pointer_known1_repair" />
    <earl:pointer rdf:resource="#pointer_known1_html" />
    <earl:pointer rdf:resource="#pointer_known1_css" />
    <earl:pointer rdf:resource="#pointer_known1_img" />
    <earl:outcome rdf:resource="http://www.w3.org/ns/earl#failed" />
  </earl:TestResult>
```

Web Content Accessibility Guidelines 2.0 Ruleset

1. Perceivable - Information and user interface components must be presentable to users in ways they can perceive.

1.1 Text Alternatives: Provide text alternatives for any non-text content so that it can be changed into other forms people need, such as large print, braille, speech, symbols or simpler language.

Requirement 1.1.1 Non-text Content (Level A)

ID	Status	Priority	Severity	Rule Description
26	Accepted	Priority 1	Violation	Rule 26: Images must have valid alt text.
27	Accepted	Priority 1	Violation	Rule 27: Image file name is not valid alt text.
28	Accepted	Priority 1	Violation	Rule 28: Certain words cannot be used as a part of valid alt text
29	Accepted	Priority 1	Recommendation	Rule 29: Length of alt text.
30	Accepted	Priority 1	Violation	Rule 30: Longdesc must have valid URI.
31	Accepted	Priority 1	Violation	Rule 31: If an image has an alt or title attribute, it should not have a presentation role.
32	Proposed	Priority 1	Potential Violation	Rule 32: ALT text must describe content or purpose of image

1.2 Time-based Media: Provide alternatives for time-based media.

Requirement 1.2.1 Audio-only and Video-only (Prerecorded) (Level A)

ID	Status	Priority	Severity	Rule Description
46	Proposed	Priority 1	Potential Violation	Rule 46: Prerecorded audio with no video or image tracks needs a text transcript which includes speaker information.
47	Proposed	Priority 1	Potential Violation	Rule 47: Prerecorded video with no audio track needs text or audio descriptions of the video content.

Requirement 1.2.2 Captions (Prerecorded) (Level A)

ID	Status	Priority	Severity	Rule Description
48	Proposed	Priority 1	Potential Violation	Rule 48: Prerecorded video with audio track needs synchronized captions.

Requirement 1.2.3 Audio Description or Media Alternative (Prerecorded) (Level A)

ID	Status	Priority	Severity	Rule Description
49	Proposed	Priority 1	Potential Violation	Rule 49: Prerecorded video with audio track needs a audio or text description of the video content.

Requirement 1.2.4 Captions (Live) (Level AA)

ID	Status	Priority	Severity	Rule Description
50	Proposed	Priority 1	Potential Violation	Rule 50: Live audio of speech requires realtime captioning of the speakers.

Requirement 1.2.5 Audio Description (Prerecorded) (Level AA)

ID	Status	Priority	Severity	Rule Description
51	Proposed	Priority 1	Potential Violation	Rule 51: Synchronized audio descriptons of video.

Requirement 1.2.6 Sign Language (Prerecorded) (Level AAA)

ID	Status	Priority	Severity	Rule Description
52	Proposed	Priority 1	Potential Violation	Rule 52: Synchronized sign language for audio track.

Requirement 1.2.7 Extended Audio Description (Prerecorded) (Level AAA)

ID	Status	Priority	Severity	Rule Description
53	Proposed	Priority 1	Potential Violation	Rule 53: Extended audio description if audio track does not provided enough time to fully describe video content.

Requirement 1.2.8 Media Alternative (Prerecorded) (Level AAA)

ID	Status	Priority	Severity	Rule Description
54	Proposed	Priority 1	Potential Violation	Rule 54: Text alternative to audio and video content

Requirement 1.2.9 Audio-only (Live) (Level AAA)

ID	Status	Priority	Severity	Rule Description
55	Proposed	Priority 1	Potential Violation	Rule 55: Provide text alternatives to live audio

1.3 Adaptable: Create content that can be presented in different ways (for example simpler layout) without losing information or structure.

Requirement 1.3.1 Info and Relationships (Level A)

ID	Status	Priority	Severity	Rule Description
3	Accepted	Priority 1	Violation	Rule 3: Data tables must use summary attribute.

4	Accepted	Priority 1	Violation	Rule 4: Data tables must use th elements
5	Accepted	Priority 1	Violation	Rule 5: Summary attribute content must be unique.
6	Accepted	Priority 1	Violation	Rule 6: Complex data tables must have ids on th elements.
7	Accepted	Priority 1	Violation	Rule 7: For complex data tables table ids must be unique.
8	Accepted	Priority 1	Violation	Rule 8: Complex data table td elements must have header attributes.
9	Accepted	Priority 1	Violation	Rule 9: Complex data tables header ids must be on the page.

Requirement 1.3.2 Meaningful Sequence (Level A)

ID	Status	Priority	Severity	Rule Description
36	Accepted	Priority 1	Potential Violation	Rule 36: Do not use nested tables for positioning.
56	Proposed	Priority 1	Potential Violation	Rule 56: Document reading order makes sense when layout tables and css positioning is disabled.

Requirement 1.3.3 Sensory Characteristics (Level A)

ID	Status	Priority	Severity	Rule Description
37	Proposed	Priority 1	Potential Violation	Rule 37: Document reading order makes sense when layout tables and css positioning is disabled.
72	Accepted	Priority 1	Potential Violation	Rule 72: Instructions and operation must be accessible and not depend on solely on color, size, location or sound .

1.4 Distinguishable: Make it easier for users to see and hear content including separating foreground from background.

Requirement 1.4.1 Use of Color (Level A)

ID	Status	Priority	Severity	Rule Description
64	Proposed	Priority 1	Potential Violation	Rule 64: Verify that color is not used as the only visual means of conveying information, indicating an action, prompting a response, or distinguishing a visual element.

Requirement 1.4.2 Audio Control (Level A)

ID	Status	Priority	Severity	Rule Description
57	Accepted	Priority 1	Potential Violation	Rule 57: Verify audio does not start playing automatically when document is loaded.

Requirement 1.4.3 Contrast (Minimum) (Level AA)

ID	Status	Priority	Severity	Rule Description
65	Accepted	Priority 1	Violation	Rule 65: Color contrast ratio must be > 3 for large text

Requirement 1.4.4 Resize text (Level AA)

ID	Status	Priority	Severity	Rule Description
58	Accepted	Priority 2	Potential Violation	Rule 58: Verify only relative font sizes are used to define font sizes.

Requirement 1.4.5 Images of Text (Level AA)

ID	Status	Priority	Severity	Rule Description
33	Accepted	Priority 2	Potential Violation	Rule 33: Verify that images are not being used only to stylize text.

Requirement 1.4.6 Contrast (Enhanced) (Level AAA)

ID	Status	Priority	Severity	Rule Description
66	Accepted	Priority 2	Violation	Rule 66: Color contrast ratio should be > 4.5

Requirement 1.4.7 Low or No Background Audio (Level AAA)

No rules

Requirement 1.4.8 Visual Presentation (Level AAA)

No rules

Requirement 1.4.9 Images of Text (No Exception) (Level AAA)

No rules

2. Operable - User interface components and navigation must be operable.

2.1 Keyboard Accessible: Make all functionality available from a keyboard.

Requirement 2.1.1 Keyboard (Level A)

ID	Status	Priority	Severity	Rule Description
59	Accepted	Priority 1	Violation	Rule 59: Focusable elements with MouseOver should also have OnFocus event handlers.
60	Accepted	Priority 1	Violation	Rule 60: Focusable elements with an OnMouseOut should also have OnBlur event handlers.
61	Accepted	Priority 1	Violation	Rule 61: Every onClick event handler should be on a focusable element.
62	Accepted	Priority 2	Violation	Rule 62: OnChange event handler should not be used with the select element.

63	Accepted	Priority 1	Violation	Rule 63: onmousedown, onmouseup and onmousemove event handlers should have keyboard equivalents.
94	Accepted	Priority 1	Violation	Rule 94: Check that non-form and non-anchor elements with event handlers have valid roles.
95	Accepted	Priority 1	Violation	Rule 95: Check that elements with mouse event handlers also have key event handlers
96	Accepted	Priority 1	Violation	Rule 96: Check that enabled elements with ACTIVE-DSCENDANT have valid tab index
97	Accepted	Priority 1	Violation	Rule 97: Check that elements without 'aria-activedescendant' that have roles requiring a container have focusable children
98	Accepted	Priority 1	Violation	Rule 98: Check that elements without 'aria-activedescendant' that have roles requiring a container have key event handlers

Requirement 2.1.2 No Keyboard Trap (Level A)

No rules

Requirement 2.1.3 Keyboard (No Exception) (Level AAA)

ID	Status	Priority	Severity	Rule Description
1	Accepted	Priority 2	Violation	Rule 1: Accesskey attribute values should be unique.
2	Accepted	Priority 2	Violation	Rule 2: Accesskey attribute values should not interfere with IE shortcuts.

2.2 Enough Time: Provide users enough time to read and use content.

Requirement 2.2.1 Timing Adjustable (Level A)

No rules

Requirement 2.2.2 Pause, Stop, Hide (Level A)

ID	Status	Priority	Severity	Rule Description
68	Accepted	Priority 1	Violation	Rule 68: The blink and marquee elements must not be used.

Requirement 2.2.3 No Timing (Level AAA)

No rules

Requirement 2.2.4 Interruptions (Level AAA)

No rules

Requirement 2.2.5 Re-authenticating (Level AAA)

No rules

2.3 Seizures: Do not design content in a way that is known to cause seizures.

Requirement 2.3.1 Three Flashes or Below Threshold (Level A)

No rules

Requirement 2.3.2 Three Flashes (Level AAA)

No rules

2.4 Navigable: Provide ways to help users navigate, find content, and determine where they are.**Requirement 2.4.1 Bypass Blocks (Level A)**

ID	Status	Priority	Severity	Rule Description
10	Accepted	Priority 1	Violation	Rule 10: Frame element must have a title attribute.
11	Accepted	Priority 1	Violation	Rule 11: Title attributes for frames must be unique.
12	Accepted	Priority 2	Recommendation	Rule 12: Frames should not be hidden or empty.

Requirement 2.4.2 Page Titled (Level A)

ID	Status	Priority	Severity	Rule Description
13	Accepted	Priority 1	Violation	Rule 13: Title element should not be empty.
14	Accepted	Priority 1	Recommendation	Rule 14: Missing or empty H1 element.
15	Accepted	Priority 1	Recommendation	Rule 15: H1 element content should not come only from the alt text of an image.
16	Accepted	Priority 1	Recommendation	Rule 16: H1 should match a subset of the words in the title element.
17	Accepted	Priority 1	Recommendation	Rule 17: No more than two h1 elements.
24	Accepted	Priority 1	Violation	Rule 24: Title content should be concise.
25	Accepted	Priority 1	Violation	Rule 25: Title text must contain more than one word.

Requirement 2.4.3 Focus Order (Level A)

No rules

Requirement 2.4.4 Link Purpose (In Context) (Level A)

ID	Status	Priority	Severity	Rule Description
38	Accepted	Priority 1	Recommendation	Rule 38: Link text should be at least four 4 characters long.
40	Accepted	Priority	Violation	Rule 40: Links that point to different HREFs

		1		should have different link text context.
42	Accepted	Priority 1	Recommendation	Rule 42: Images should be at least 16 pixels by 16 pixels when used as links.
43	Accepted	Priority 1	Violation	Rule 43: Links with images and text content, the alt attribute should be unique to the text content or empty.
44	Accepted	Priority 1	Potential Violation	Rule 44: Purpose of link must be determined from context.

Requirement 2.4.5 Multiple Ways (Level AA)

No rules

Requirement 2.4.6 Headings and Labels (Level AA)

ID	Status	Priority	Severity	Rule Description
18	Accepted	Priority 1	Violation	Rule 18: Headings must have text content.
19	Accepted	Priority 1	Recommendation	Rule 19: Text content for a heading must not come just from image alt text.
20	Accepted	Priority 1	Recommendation	Rule 20: Heading content should be concise.
21	Accepted	Priority 1	Violation	Rule 21: Heading elements should be properly nested.
22	Accepted	Priority 1	Recommendation	Rule 22: The content of the headings of the same level within the same section should be unique.
23	Accepted	Priority 1	Potential Violation	Rule 23: Heading elements (h1..h6) should be used for structuring information on the page.

Requirement 2.4.7 Focus Visible (Level AA)

No rules

Requirement 2.4.8 Location (Level AAA)

No rules

Requirement 2.4.9 Link Purpose (Link Only) (Level AAA)

ID	Status	Priority	Severity	Rule Description
39	Accepted	Priority 1	Violation	Rule 39: Links with the same HREF should have the same link text.
41	Accepted	Priority 1	Violation	Rule 41: Links that point to different HREFs should have different link text.
45	Accepted	Priority 1	Potential Violation	Rule 45: Purpose of link must be determined from link text.

Requirement 2.4.10 Section Headings (Level AAA)

No rules

3. Understandable - Information and the operation of user interface must be understandable.

3.1 Readable: Make text content readable and understandable.

Requirement 3.1.1 Language of Page (Level A)

ID	Status	Priority	Severity	Rule Description
34	Accepted	Priority 1	Violation	Rule 34: Each page must have a lang attribute on its html element.
35	Accepted	Priority 1	Violation	Rule 35: lang attribute on html element must have a valid two-character language code.

Requirement 3.1.2 Language of Parts (Level AA)

No rules

Requirement 3.1.3 Unusual Words (Level AAA)

No rules

Requirement 3.1.4 Abbreviations (Level AAA)

No rules

Requirement 3.1.5 Reading Level (Level AAA)

No rules

Requirement 3.1.6 Pronunciation (Level AAA)

No rules

3.2 Predictable: Make Web pages appear and operate in predictable ways.

Requirement 3.2.1 On Focus (Level A)

No rules

Requirement 3.2.2 On Input (Level A)

No rules

Requirement 3.2.3 Consistent Navigation (Level AA)

No rules

Requirement 3.2.4 Consistent Identification (Level AA)

No rules

Requirement 3.2.5 Change on Request (Level AAA)

No rules

3.3 Input Assistance: Help users avoid and correct mistakes.

Requirement 3.3.1 Error Identification (Level A)

No rules

Requirement 3.3.2 Labels or Instructions (Level A)

ID	Status	Priority	Severity	Rule Description
73	Accepted	Priority 1	Violation	Rule 73: Each fieldset element should contain a legend element.
74	Accepted	Priority 1	Recommendation	Rule 74: The label element should not encapsulate select and textarea elements.
75	Accepted	Priority 1	Violation	Rule 75: Each input element with type=text password checkbox radio file and each select and textarea element should either be referenced by the for attribute of a label element via its id attribute, or have a title attribute.
76	Accepted	Priority 1	Violation	Rule 76: Inputs element of type=[image] must have an alt or a title attribute.
77	Accepted	Priority 2	Recommendation	Rule 77: Input elements where type=[button submit reset] must have a value or title attribute.
78	Accepted	Priority 1	Violation	Rule 78: Each button element must contain content.
79	Accepted	Priority 1	Violation	Rule 79: Effective labels should be unique.
80	Accepted	Priority 1	Violation	Rule 80: Labels must have text content.
81	Accepted	Priority 1	Violation	Rule 81: Legends must have text content.
82	Accepted	Priority 1	Recommendation	Rule 82: Title attributes used for labeling form controls must have content.
83	Accepted	Priority 1	Violation	Rule 83: Form controls must have unique ids.

Requirement 3.3.3 Error Suggestion (Level AA)

No rules

Requirement 3.3.4 Error Prevention (Legal, Financial, Data) (Level AA)

No rules

Requirement 3.3.5 Help (Level AAA)

No rules

Requirement 3.3.6 Error Prevention (All) (Level AAA)

No rules

4. Robust - Content must be robust enough that it can be interpreted reliably by a wide variety of user agents, including assistive technologies.

4.1 Compatibility with Assistive Technologies: Maximize compatibility with current and future user agents, including assistive technologies.

Requirement 4.1.1 Parsing Content (Level A)

ID	Status	Priority	Severity	Rule Description
67	Accepted	Priority 2	Violation	Rule 67: Do not use the FONT element to style text
69	Accepted	Priority 2	Violation	Rule 69: Do not use the B element.
70	Accepted	Priority 2	Violation	Rule 70: Do not use the I element.
71	Accepted	Priority 2	Violation	Rule 71: Do not use the U element.
85	Accepted	Priority 1	Violation	Rule 85: ARIA attributes have valid values
86	Accepted	Priority 1	Violation	Rule 86: ARIA ID references must be valid IDRefs
87	Accepted	Priority 1	Violation	Rule 87: ARIA attributes can only be used with certain roles

Requirement 4.1.2 Name, Role, Value (Level A)

ID	Status	Priority	Severity	Rule Description
84	Accepted	Priority 1	Violation	Rule 84: Check aria properties and states for valid roles and properties
88	Accepted	Priority 1	Violation	Rule 88: Roles must contain their required child roles
89	Accepted	Priority 1	Violation	Rule 89: Child roles must be contained by the proper parent role
90	Accepted	Priority 1	Violation	Rule 90: Required properties and states should be defined
91	Accepted	Priority 1	Violation	Rule 91: Required properties and states must not be empty
92	Accepted	Priority 1	Violation	Rule 92: Role value must be valid
93	Accepted	Priority 1	Violation	Rule 93: Check that 'ARIA-' attributes are valid properties and states

Referencias

Fuentes Electrónicas

AENOR (2004). *Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web - UNE 139803:2004.*

<http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0032576#.VH6lvFx7h5Q>

AENOR (2010).

<http://www.aenor.es>

AENOR (2012). *Requisitos de accesibilidad para contenidos en la Web - UNE 139803:2012.*

<http://www.aenor.es/aenor/normas/normas/fichanorma.asp?tipo=N&codigo=N0049614&pdf=#.VH6l11x7h5Q>

Andrew Arch - W3C (2002). *Beneficios Auxiliares del Diseño Web Accesible.*

<http://www.w3.org/2003/11/benefits-es.html>

Ben Caldwell et al. - W3C (2008). *Web Content Accessibility Guidelines (WCAG 2.0).*

<http://www.w3.org/TR/WCAG20/>

CTIC Centro tecnológico (2014). *TAW Tools.*

<http://www.tawdis.net/tools/?lang=es>

Eric Eggert and Shadi Abou-Zahra - W3C (2014). *Web Accessibility Evaluation Tools List.*

<http://www.w3.org/WAI/ER/tools/>

Frecuencia Latinoamérica (2013). *Dispositivos móviles en hogares conectados de la Argentina.*

<http://noticias.frecuenciaonline.com/2013/06/13/dispositivos-moviles-en-hogares-conectados-de-la-argentina/>

Free Software Foundation Inc. (2007). *GNU General Public License 3.0*

<http://www.gnu.org/licenses/gpl-3.0.en.html>

Greg Gay et al. (2011). *Web Accessibility Checker.*

<http://achecker.ca/checker/index.php>

INDEC (2012). *Censo 2010.*

http://www.censo2010.indec.gov.ar/archivos/censo2010_tomo1.pdf

Info News (2013). *La conexión a internet argentina, entre las más lentas del mundo.*

<http://www.infonews.com/2013/02/21/sociedad-62052-la-conexion-a-internet-argentina-entre-las-mas-lentas-del-mundo.php>

- Instituto Nacional de Tecnologías de la Comunicación (2008). *Introducción a la accesibilidad Web*.
<http://www.inteco.es/file/bpoTr1nHdohApbHgFsyFSw>
- ISO/IEC 40500/2012 (2012). *Information technology - W3C Web Content Accessibility Guidelines (WCAG) 2.0*.
http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=58625
- Jim Thatcher (2007). *Side by Side WCAG vs. 508*.
<http://jimthatcher.com/sidebyside.htm>
- José Manuel Ciges Regueiro (2013). *Comparativa MongoDB vs MySQL*.
<http://www.ciges.net/comparativa-mongodb-vs-mysql-conclusiones-2-parte>
- Ley 106-246 (2000). *Public Law 106-246, 106th Congress*.
<http://www.gpo.gov/fdsys/pkg/PLAW-106publ246/pdf/PLAW-106publ246.pdf>
- Ley Argentina 26.653 (2010). *Acceso a la información pública*.
<http://www.infoleg.gov.ar/infolegInternet/anexos/175000-179999/175694/norma.htm>
- Ley Española 56/2007 (2007). *Medidas de Impulso de la Sociedad de la Información*.
<http://www.boe.es/buscar/doc.php?id=BOE-A-2007-22440>
- Michael Cooper et al. - W3C (2015). *Techniques and Failures for Web Content Accessibility Guidelines 2.0*.
<http://www.w3.org/TR/WCAG20-TECHS/>
- Norma CWA 15554 (2006). *Specifications for a Web Accessibility Conformity Assessment Scheme and a Web Accessibility Quality Mark*.
<ftp.cenorm.be/PUBLIC/CWAs/e-Europe/WAC/CWA15554-00-2006-Jun.pdf>
- Norma técnica colombiana 5854 (2011). *Accesibilidad a páginas web*.
<http://tienda.icontec.org/brief/NTC5854.pdf>
- Oficina Nacional de Tecnologías de Información (ONTI) (2011) . *Disposición 4/2011*.
<http://www.infoleg.gov.ar/infolegInternet/anexos/190000-194999/190867/norma.htm>
- Oliver White (2014). *Top 4 Java Web Frameworks Revealed: Real Life Usage Data of Spring MVC, Vaadin, GWT and JSF*.
<http://zeroturnaround.com/rebellabs/top-4-java-web-frameworks-revealed-real-life-usage-data-of-spring-mvc-vaadin-gwt-and-jsf/>
- OMS (2012). *Accesibilidad Web*.
<https://sites.google.com/site/accesibilizando/home>
- Open Accessibility Alliance y OpenAjax Accessibility Task Force (2014). *Functional Accessibility Evaluator 1.1*.
<http://fae.cita.uiuc.edu/about/>

- OpenAjax Alliance (2011). *Accessibility Rules Format 1.0*.
http://www.openajax.org/member/wiki/Accessibility_Rules_Format_1.0
- Oracle (1999). *Code convention for the Java Programming Language*.
<http://www.oracle.com/technetwork/java/codeconvtoc-136057.html>
- Real Decreto 1494/2007 (2007).
<http://www.boe.es/buscar/doc.php?id=BOE-A-2007-19968>
- Shadi Abou-Zahra and W3C/WAI (2011). *Evaluation and Report Language (EARL) 1.0 Schema*.
<http://www.w3.org/TR/EARL10-Schema/>
- Shawn Lawton Henry and Judy Brewer - W3C (2005). *WAI Mission and Organization*.
<http://www.w3.org/WAI/about.html>
- Shawn Lawton Henry and Matt May - W3C (2005). *User Agent Accessibility Guidelines (UAAG) Overview*.
<http://www.w3.org/WAI/intro/uaag.php>
- Shawn Lawton Henry et al. - W3C (2012). *Technical Factors in Developing a Web Accessibility Business Case for Your Organization*.
<http://www.w3.org/WAI/bcase/tech>
- Shawn Lawton Henry et al. - W3C (2013). *Authoring Tool Accessibility Guidelines (ATAG) Overview*.
<http://www.w3.org/WAI/intro/atag.php>
- Shawn Lawton Henry et al. - W3C (2014). *WAI-ARIA Overview*.
<http://www.w3.org/WAI/intro/aria.php>
- Shawn Lawton Henry et al. - W3C (2005). *Introduction to Web Accessibility*.
<http://www.w3.org/WAI/intro/accessibility>
- Shawn Lawton Henry et al. - W3C (2012). *Web Content Accessibility Guidelines (WCAG) Overview*.
<http://www.w3.org/WAI/intro/wcag>
- United states Access Board (2000). *Section 508 Standards for Electronic and Information Technology*.
<http://www.section508.gov/section-508-standards-guide>
- Web Accessibility Benchmarking Cluster (2012). *UWEM, Unified Web Evaluation Methodology version 1.2*.
http://www.wabcluster.org/uwem1_2/
- WebAIM (2013). *Accessibility Evaluation Tools*.
<http://webaim.org/articles/tools/>

WebAIM (2014). *Web Accessibility Evaluation Tool*.

<http://wave.webaim.org/>

WebPosible (2007). *Barreras y discapacidades en la web*.

http://www.webposible.com/articulos/barreras_discapacidades.html

Wendy Chisholm et al. - W3C (1999). *Web Content Accessibility Guidelines 1.0 (WCAG 1.0)*.

<http://www.w3.org/TR/WCAG10/>

Fuentes Bibliográficas

Aries Beltran (2013). *Getting Started with PhantomJS*. Birmingham, UK: Packt Publishing.

Claus Ibsen (2011). *Camel in Action*. Stamford (USA): Manning Publication Co.

Erich Gamma et al. (1994). *Design patterns*. United States of America: Addison-Wesley Professional.

Jim Thatcher (2006). *Web accessibility: Web standards and regulatory compliance*. United States of America: Friends of.

Kennet S. Rubin (2012). *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. United States of America: Addison-Wesley Professional; 1 edition.

Kristina Chodorow (2013). *MongoDB: The Definitive Guide*. United States of America: O'Reilly Media.

Luzardo Alliey A. (2009). *Diseño de la interfaz gráfica web en función de los dispositivos móviles*. Tesis de maestría. Universidad de Palermo.

Maia Naftali (2010). *Análisis e Integración de métricas para la Accesibilidad Web*. Tesis de grado. Universidad de Buenos Aires.

Pramod J. Sadalage and Martin Fowler (2012). *NoSQL Distilled: A Brief Guide to the Emerging World of Polyglot Persistence*. United States of America: Pearson Education, Inc.

Raoul-Gabriel Urma et al. (2014). *Java 8 in Action: Lambdas, Streams, and functional-style programming*. United States of America: Manning Publications; 1 edition.

Satya Avasarala (2014). *Selenium WebDriver Practical Guide*. United States of America: AdactIn Group Pty Ltd.

Tim Berglund (2013). *Gradle Beyond the Basics*. United States of America: O'Reilly Media.

Wilbert O. Galitz (1997). *The Essential Guide to User Interface Design: Introduction to GUI Design Principles and Techniques*. United States of America: Wiley.