

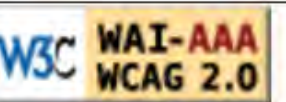

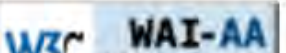



ACCESIBILIDAD UNIVERSAL EN LAS NUEVAS TECNOLOGÍAS

Accesibilidad Universal en las Tecnologías de la Información y la Comunicación

A	AA	AAA
 (png , gif , svg , eps)	 (png , gif , svg , eps)	 (png , gif , svg , eps)
		

Diseño y desarrollo Web accesible utilizando HTML5 y CSS3 con nivel de conformidad A, respecto a las pautas WCAG

Malqui, Diego David

*Proyecto Final de Carrera Ingeniería y Ciencias Hídricas.
Año 2015*

Universidad Nacional del Litoral, Argentina

Tutor: Minni Hugo. Profesor Universidad Nacional del Litoral

1. Introducción

Este Proyecto Final de Carrera (PFC) titulado "Diseño y Desarrollo Web Accesible Utilizando HTML5 y CSS3 con Nivel de Conformidad A, respecto a WCAG", está dividida en dos: una sección teórica y una sección práctica que se basa en el análisis teórico para desarrollar un caso de estudio.

La sección teórica comienza detallando conceptos generales de accesibilidad y luego brinda más específicamente nociones de accesibilidad Web, considerada como: un conjunto de estándares de diseño y de software pensados puntualmente para que personas con algún tipo de discapacidad ya sea física, sensorial, cognitiva o de entorno, puedan percibir, entender, navegar e interactuar con la Web. Luego se presentan las Pautas de Accesibilidad al Contenido Web, reguladas por un grupo de trabajo denominado Web Accessibility Initiative (WAI) que forman parte de World Wide Web Consortium (W3C) – el máximo organismo dentro de la jerarquía de Internet. Para concluir la parte teórica del Proyecto, se describe el estado del arte de las tecnologías Web enumerando las herramientas más recientes y actuales que nos permitirán concretar la segunda sección del Proyecto.

La sección de desarrollo también se ha separado en dos partes. La primer parte puntualizará brevemente los requerimientos con los que se debe completar el caso de estudio. La segunda parte, representa el núcleo del Proyecto y es la que analiza las posibilidades y alternativas para implementar una interface Web de forma accesible, señalando el motivo de las decisiones e implementándolas en el caso de estudio expuesto en <http://asetur2010.com.ar/PFC/>. A lo largo del capítulo se identifican las consideraciones necesarias para cumplir con los diferentes niveles de accesibilidad, se detallan decisiones de codificación y descuidos cometidos extensamente que afectan negativamente la accesibilidad en páginas externas a nuestro caso de estudio.

El Proyecto concluye con una revisión cautelosa de los criterios de conformidad del nivel planteado (Nivel A), detallando su cumplimiento en el caso de estudio. Para finalizar, el Proyecto plantea líneas futuras de investigación y culmina con una reflexión en cuanto a la aceptabilidad o resistencia de la accesibilidad Web de parte de desarrolladores y empresas.

1.1. Motivación

La World Wide Web (WWW) es uno de los servicios de INTERNET que más repercusiones ha tenido. Se trata de un conjunto de protocolos de comunicación que permiten, de forma sencilla, la consulta remota de archivos de hipertexto. Básicamente, usuarios finales pueden visualizar a través de navegadores información alojada en cualquier parte del mundo. Estos avances en la tecnología e internet, sin duda han acercado conocimientos y oportunidades a muchas personas; pero sin embargo y lamentablemente, hay personas que sin la adaptación necesaria de estas tecnologías a sus necesidades específicas, no podrían tener acceso a ellas.

Actualmente se invierte gran cantidad de tiempo en especificar pautas, normas, reglamentaciones y leyes para hacer que no sólo la Web, sino todo documento digital sea más accesible y pueda brindar a personas con algún tipo de limitación o discapacidad la posibilidad de consultar y comprender el documento de manera independiente.

1.2. Regulaciones actuales

El lenguaje HTML (HyperText Markup Language) es un estándar reconocido en todo el mundo, cuyas normas define el organismo sin ánimo de lucro W3C. Este, se creó en 1994 con el objetivo de asegurar el correcto crecimiento de la red a largo plazo y hoy se ha convertido en el máximo organismo dentro de la jerarquía de Internet. W3C además, trabaja conjuntamente en el desarrollo de estándares Web que sirven de guía a desarrolladores y diseñadores.

Dentro de W3C ha surgido un grupo de trabajo denominado WAI, que en 1999 publicó la versión 1.0 de sus Pautas de Accesibilidad al Contenido Web (WCAG 1.0). Con el paso del tiempo, WCAG 1.0 se han convertido en un referente internacionalmente aceptado y en diciembre del 2008, las pautas fueron actualizadas a una nueva versión WCAG 2.0, aprobadas como recomendación oficial.

Las Pautas de Accesibilidad al Contenido Web contienen además de las pautas en sí, una serie de puntos de verificación que ayudan a detectar posibles errores. Cada uno de estos puntos está asignado a un nivel de prioridad. Prioridad 1, prioridad 2 o prioridad 3. En función del cumplimiento de estos puntos de verificación se establecen los niveles de conformidad. Para obtener un nivel de conformidad A, se deben satisfacer todos los puntos de verificación de prioridad 1. Para obtener un nivel de conformidad AA, se deben satisfacer todos los puntos de verificación de prioridad 1 y prioridad 2. Para obtener un nivel de conformidad AAA, se deben satisfacer todos los puntos de verificación de prioridad 1, prioridad 2 y prioridad 3.

Las WCAG 2.0 son utilizadas como base para realizar documentos digitales más accesibles. Más allá del software utilizado para su redacción (Office Word, openOffice, Bloc de notas, etc.) los conceptos se extienden fácilmente. El presente documento expone las características de accesibilidad que se pueden desprender de las WCAG 2.0.

La WAI, grupo que se ha nombrado anteriormente, también, es la encargada de la redacción de las Pautas de Accesibilidad para Herramientas de Autor (ATAG) que muestran cómo hacer que los editores de HTML, Herramientas de Diseño y otras herramientas de autor, sean accesibles para personas con discapacidad. Así también, son encargados de la redacción de las Pautas de Accesibilidad para Agentes de Usuario (UAAG) las cuales muestran a los desarrolladores de navegadores Web, reproductores

multimedia, tecnologías asistidas y otros agentes de usuario¹ cómo hacerlos accesibles para personas con discapacidad [1].

1.3. Objetivos

A continuación se nombran los objetivos generales y específicos perseguidos:

1.3.1. Objetivo General

Analizar las posibilidades y alternativas de implementación de las diferentes secciones de una interface Web, utilizando las tecnologías y herramientas actuales, a fin de obtener un nivel de accesibilidad A de acuerdo a las Pautas de Accesibilidad al Contenido Web (WCAG 2.0).

1.3.2. Objetivos Específicos

- Determinar qué tecnologías son convenientes para el desarrollo de la interface de usuario de un sitio Web accesible.
- Implementar un caso de estudio, basado en los requerimientos reunidos en la Secretaria de Turismo de la Ciudad de Santa Fe, que responda a los criterios de accesibilidad en la Web WCAG 2.0.

1.3.3. Modificaciones en los objetivos iniciales

Los objetivos del anteproyecto fueron redactados nuevamente luego de lograr una mejor dimensión de la extensión de las WCAG 2.0. En el presente documento se expresan de forma tal de enfatizar que el propósito del Proyecto es concretar un análisis relativo a las opciones que brindan las tecnologías actuales plasmando la experiencia en un caso de estudio concreto para garantizar y corroborar su funcionalidad. Los objetivos brindados de esta forma ponen en evidencia que la temática concreta de la página no es relevante y que pueden ser utilizados de forma general.

¹ Un agente de usuario es un elemento de software que actúa en nombre de un usuario solicitando información al servidor [37].

1.4. Alcance

El caso de estudio tendrá como mínimo un nivel A, acorde a las Pautas de Accesibilidad al Contenido Web 2.0 y expondrá todos los requerimientos recopilados en el capítulo 4.

1.5. Herramientas Externas Utilizadas en la Implementación

Como la gran mayoría de desarrollos de software, este trabajo también se vale de herramientas que ayudan con determinadas funcionalidades, las cuales van más allá del alcance del Proyecto.

A lo largo del trabajo se utiliza el Sistema de Gestión de Contenido (CMS - Content Management System) Wordpress, como así también complementos brindados para este CMS. Se utilizó una plantilla llamada DevDmBootstrap3 y plugins que se detallarán más adelante.

Así también, para facilitar el diseño adaptativo el trabajo se vale de las utilidades del framework de CSS (cascading style sheets) Bootstrap. Este framework es el más utilizado por los desarrolladores que tienen como objetivo crear sitios Web adaptativos². En la sección dedicada al desarrollo se detalla si han sido utilizadas sus funcionalidades y si éstas tienen alguna relevancia en aspectos de accesibilidad.

2. Accesibilidad

La Real Academia Española describe accesibilidad como aquello con cualidad de accesible, es decir, algo de fácil comprensión e inteligible. Por lo tanto, se puede definir como características de accesibilidad a aquellas particularidades del urbanismo, de la edificación, del transporte o de los medios y sistemas de comunicación que le permiten a cualquier persona su utilización y disfrute de manera autónoma, independientemente de su condición física, psíquica o sensorial [47].

¹ Se toma como referencia cantidad de descargas y "estrellas" en <https://github.com>.

2.1. Accesibilidad Web

Especificando el término accesibilidad, se halla accesibilidad Web, la cual se define como la posibilidad efectiva de que el mayor número de personas puedan acceder de manera satisfactoria a la Web y hacer uso de sus contenidos, independientemente de cualquier limitación personal o limitación derivada del entorno [22].

En lo concreto, la accesibilidad Web refiere a un conjunto de estándares de diseño y de software pensados específicamente para que personas con algún tipo de discapacidad ya sea física, sensorial, cognitiva o de entorno, puedan percibir, entender, navegar e interactuar con la Web [1].

La Accesibilidad Web engloba muchos tipos de personas, incluyendo aquellas con limitaciones visuales, auditivas, físicas, cognitivas y neurológicas; así como también personas de edad avanzada o aquellas que sufren de dificultades derivadas del contexto de uso, ya sean técnicas o ambientales. Accesibilidad Web abarca también el propósito de superar dificultades causadas por insuficiencia de medios, restricciones en cuanto al dominio del idioma y poco o nulo conocimiento del uso de dispositivos electrónicos.

Los usuarios con discapacidad interactúan con las computadoras mediante las llamadas tecnologías asistivas o tecnologías de apoyo, las cuales están diseñadas para trabajar correctamente con páginas Web teniendo en cuenta criterios específicos.

Al ser un sistema abierto y englobar millones de documentos redactados por distintas personas y miles de funcionalidades desarrolladas alrededor del mundo, el comportamiento de la Web pasa a depender de la propia página.

Son las distintas estructuras, así como las incontables formas de presentación de los documentos, las que dificultan el trabajo de los sistemas de asistencia y generan "barreras" que interfieren con la accesibilidad. En otras palabras,

para la correcta navegación en Web mediante tecnologías asistivas es necesario contemplar las Pautas de Accesibilidad al Contenido Web para evitar “barreras”.

El respeto a los estándares de Accesibilidad Web procura extender al ámbito digital, el derecho de acceso igualitario a la cultura, al ocio y al tiempo libre establecido por nuestra constitución.

2.1.1. Tipos de Limitaciones

Los diferentes tipos de limitaciones y discapacidades marcan las necesidades a cubrir por las Pautas de Accesibilidad Web y todo avance en este campo. Las limitaciones de una persona pueden ser acumulativas, por lo que un mismo usuario puede padecer de varias limitaciones simultáneamente. Conocer las diferentes limitaciones y las tecnologías asistivas que suelen utilizar es fundamental para poder completar una página Web accesible para la mayor cantidad de usuarios posibles [21] [26].

Visual

• Ceguera Completa

En este contexto, se puede definir a las personas con ceguera completa a aquellas quienes deciden utilizar un lector de pantalla. Los lectores de pantalla consisten en un sintetizador de voz que “lee y explica” lo que se visualiza en la pantalla. Estos son uno de los productos de apoyo³ más utilizados y la idea fundamental del mismo es lograr que el usuario pueda comprender la información, no solo escucharla. Por lo tanto, los lectores de pantalla no hacen una lectura pasiva ni tampoco lee el documento línea por línea, sino que brinda los recursos para “oír” el documento. Con lectores de pantalla también es posible leer los títulos, enlaces, tablas y demás [23].

Claramente sería más sencillo definir

³Cualquier producto utilizado por o para personas con discapacidad destinado a facilitar la participación o sustituir funciones/estructuras corporales y actividades [31].

a la ceguera completa como una total pérdida de visión, pero existen muchos usuarios que deciden utilizar este tipo de tecnologías para acceder a la Web o a cualquier documento electrónico, ya sea por contar con poca visión o bien, personas con desordenes en el proceso de información y problemas de aprendizaje.

• Baja Visión

Tomando en cuenta el mismo criterio que antes, se definen a las personas con baja visión como aquellos usuarios que para una correcta navegación, requieren de la posibilidad de magnificar la pantalla, sin que se pierda información o se modifique el contenido. Los magnificadores de pantalla, son programas o dispositivos hardware (por ejemplo, lupas) que permiten visualizar la pantalla con un considerable aumento en su tamaño.

• Daltonismo

El daltonismo (también llamado deficiencia o ceguera de color), ocurre cuando los colores no pueden ser vistos de manera normal. Una persona daltónica no puede distinguir entre ciertos colores, generalmente entre verdes y rojos, y ocasionalmente azules. Suelen utilizar programas muy sencillos que describen el nombre del color y/o el valor RGB para que los usuarios puedan identificarlo.

Auditiva

• Sordera Total o Parcial

No todos conocen el hecho de que el aprendizaje de la lectura está muy ligado a la audición. Por lo tanto, las personas con sordera total o parcial, generalmente tienen mayores dificultades para interpretar el lenguaje escrito. El grado de dificultad varía, entre otras cosas, dependiendo de si han perdido la audición antes de haber aprendido a hablar y leer, o no. No suelen utilizar tecnologías asistivas para navegar en la Web.

Sordoceguera

La sordoceguera es una discapacidad que resulta de la combinación de las dos limitaciones sensoriales nombradas anteriormente: limitación visual y limitación auditiva. Las personas que sufren de sordoceguera, utilizan generalmente una herramienta denominada línea braille para la interacción con los ambientes digitales.

Física o Motriz

La deficiencia física o motriz es la que provoca en el individuo que la padece alguna disfunción en el aparato locomotor. Como consecuencia se pueden producir limitaciones posturales, de desplazamiento o de coordinación del movimiento, que desembocan en dificultades de coordinación, movimientos incontrolados, fuerza reducida, habla ininteligible o dificultad con la motricidad fina y gruesa. Existen una gran cantidad de enfermedades que provocan limitaciones físicas o motrices, alguna de ellas son: Parkinson, Distrofia muscular y Parálisis cerebral.

Para las limitaciones más severas existen recursos llamados licornios. El licornio es un dispositivo que se coloca en la cabeza y permite utilizar el teclado. También existe como opciones al ratón tradicional, el ratón de boca que se maneja con soplos o pulsadores de pie y mano que se pueden manejar con la cabeza.

Cognitiva o Neurológica

Existe una gran variedad de tipos de dificultades cognitivas o neurológicas que pueden afectar a una persona, dentro de ellas se distinguen:

- Discapacidad para comprender conceptos complejos.
- Trastorno de déficit de atención.
- Falta de memoria.
- Trastornos emocionales.
- Relacionadas con el aprendizaje.

- Dislexia.

Se puede interpretar fácilmente que este tipo de usuarios no utiliza tecnologías asistivas para ingresar a las páginas Web.

Personales

Hay limitaciones personales que no son dadas por una discapacidad, entre otras pueden ser:

- Incapacidad transitoria
- Derivadas de la edad.
- Inexperiencia tecnológica.
- Idioma, culto, nivel cultural, localización geográfica.

En el caso de las limitaciones personales tampoco se suelen utilizar tecnologías asistivas de parte de los usuarios

Tecnológicas

Entre las limitaciones tecnológicas que se pueden presentar están [27]:

- Hardware.
- Software.
- Ambientales.
- Red o conexión.

Tampoco se utilizan tecnologías asistivas específicas, solo se hace uso de mecanismos brindados por el hardware de ser necesario. Si bien en el presente Proyecto y particularmente en el caso de estudio se enfatiza sobre los perjuicios de una web inaccesible para los primeros 4 tipos de limitaciones nombradas anteriormente (visual, auditiva, sordoceguera, cognitiva o neurológica), se pueden identificar otro tipo de limitaciones que también se tienen en cuenta mayormente en niveles más avanzados de accesibilidad (AA o AAA).

2.2. Pautas de Accesibilidad al Contenido Web

En diciembre del 2008 las Pautas de Accesibilidad al Contenido Web 2.0 o WCAG 2.0 fueron aprobadas como

recomendación oficial. Estas pautas comparten el mismo espíritu que su antecesora WCAG 1.0 y tienen como principal objetivo continuar con la difusión del concepto del diseño accesible y universal. La experiencia que ha supuesto la aplicación de las WCAG 1.0 ha permitido establecer una serie de requisitos que sirven de base para esta nueva versión WCAG 2.0 [40], la cual abarca los siguientes temas [43]:

- **Independencia Tecnológica.** En la versión 2.0 de las WCAG se pretende conseguir que sean aplicables sobre diferentes tecnologías ya sean presentes o futuras. Para conseguirlo, las pautas deben ser lo suficientemente genéricas y con un mayor grado de abstracción que las anteriores respecto a cuestiones técnicas.
- **Claridad en los Requisitos.** La nueva versión persigue mayor claridad en cuanto a los requisitos a cumplir; es decir, busca establecer de la forma más precisa posible cuáles son los requisitos mínimos exigibles, y plantear metodologías para que puedan verificarse de forma inequívoca, ya sea de forma automatizada o mediante una evaluación manual según requiera el caso.
- **Aumento de la Audiencia.** Las WCAG 2.0 proyecta conseguir el mayor grado de difusión de la accesibilidad posible, identificando con claridad los beneficios y beneficiarios del diseño accesible, no solo estableciendo los requisitos a cumplir, sino explicando también cuál es la finalidad de esos requisitos.

2.2.1. Clasificación de las WCAG 2.0

En este Proyecto se presentan brevemente las WCAG 2.0 (Principios, Pautas, Criterios de Conformidad y Técnicas Suficientes y Recomendables), la documentación oficial completa está disponible en la página web de la w3c [48].

Cuatro Principios

En el nivel más alto se sitúan los cuatro principios que proporcionan los

fundamentos de la accesibilidad Web:

- **Perceptible.** El sitio Web debe poder ser visitado por personas con diferentes tipos de preferencias y capacidades, y también por robots de búsqueda⁴. La información e interface deben tener en cuenta personas con todo tipo de discapacidades y se deben ofrecer alternativas a los usuarios que no pueden utilizar algunos de sus sentidos.
- **Operable.** Los diseñadores y programadores deben proporcionar elementos de interacción y de navegación que puedan ser manejados por personas con diferentes discapacidades.
- **Comprensible.** El usuario debe entender de qué se le está hablando, y no sentirse perdido con la información o servicios brindados. El sitio Web debe ser diseñado incluyendo la información y la interface de usuario fácil de usar.
- **Robusto.** Este es el principio más dependiente de la tecnología ya que se basa en la capacidad de un sitio Web de ser transmitido e interpretado por los diferentes agentes de usuarios (navegadores, reproductores multimedia o plugins como flash, java), utilizando diferente productos de apoyo.

Doce Pautas

Las doce pautas proporcionan los objetivos básicos que los desarrolladores deben lograr con el fin de crear un contenido más accesible para los usuarios con distintas discapacidades y limitaciones. Estas pautas no son evaluables, pero proporcionan el marco y los objetivos generales que ayudan a los autores a comprender los criterios de conformidad y a implementar mejor las técnicas. El primer número de cada pauta hace referencia a cuál de los doce principios pertenece y el segundo al número de pauta dentro de ese principio. Las doce pautas son las

⁴ En un programa que recupera todos los enlaces que están referenciados en una web. De ello se alimentan los grandes motores de búsqueda.

siguientes:

- **Pauta 1.1: Alternativas Textuales.** Proporcionar alternativas textuales para todo el 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.** Proveer de alternativas al medio tempo-dependiente.
- **Pauta 1.3: Adaptable.** Crear el contenido para que pueda presentarse de diferentes formas (por ejemplo, una disposición más simple) sin perder información o estructura.
- **Pauta 1.4: Distinguible.** Hacer fácil para los usuarios ver y escuchar el contenido, incluyendo la separación entre el primer plano y el fondo.
- **Pauta 2.1: Accesible por teclado.** Hacer las funcionalidades accesibles por teclado.
- **Pauta 2.2: Tiempo suficiente.** Dar tiempo a los usuarios, el tiempo suficiente para leer y usar el contenido.
- **Pauta 2.3: Convulsiones**
No diseñar el contenido de tal forma que puedan causar convulsiones.
- **Pauta 2.4: Navegable.** Proveer modos para ayudar a los usuarios a navegar, encontrar contenidos, y determinar dónde están.
- **Pauta 3.1: Legible.** Proveer un contenido fácil de leer y de comprender.
- **Pauta 3.2: Predecible.** Hacer que las páginas Web tengan aspecto y se manejen de manera predecible.
- **Pauta 3.3: Asistencia en la introducción de datos.** Ayudar a los usuarios a evitar y corregir errores.
- **Pauta 4.1: Compatible.** Maximizar la compatibilidad con los actuales y futuros agentes de usuario, incluidos los productos de apoyo.

Sesenta y un Criterios de Conformidad

Para cada una de las 12 pautas se proporcionan los criterios de conformidad verificables que permiten emplear las WCAG 2.0. Como se especificó anteriormente, son 3 los niveles de conformidad: A (el más bajo), AA y AAA (el más alto), que se obtienen según el cumplimiento de estos criterios. Como se ya se comentó los dos primeros números del criterio de conformidad corresponden al principio y a la pauta respectivamente. Siguiendo esta lógica, el tercer número corresponde al criterio de conformidad.

- **Criterio de Conformidad 1.1.1:** Contenido no textual. Todo el contenido no textual que se presenta al usuario tiene una alternativa textual que cumple el mismo propósito, excepto los siguientes casos:

- Controles o mecanismos para que el usuario introduzca datos

- Presentaciones multimedia con desarrollo temporal.

- Pruebas o ejecuciones que no serían válidos si se presentara en forma de texto.

- Contenido cuyo objetivo principal es ofrecer una experiencia sensorial específica.

- CAPTCHA⁵.

- Contenido decorativo, contenido utilizado para dar formato a la página o que no se presenta a los usuarios.

- **Criterio de Conformidad 1.2.1: Sonido solo o video solo.** Proporcionar un documento alternativo dependiendo el tipo de sonido:

- **Sonido solo grabado:** Proporcionar un documento alternativo al medio tempo-dependiente que presenta información equivalente a su contenido.

⁵ Es una pregunta utilizada para determinar cuándo un usuario es o no humano

- Video solo grabado: Proporcionar un documento alternativo al medio tempo-dependiente o una pista sonora que representa información equivalente a su contenido.

• Criterio de conformidad 1.2.3: Audiodescripción o Medio Alternativo. Proporcionar un documento alternativo al medio tempo-dependiente o una audiodescripción de todo el contenido, excepto que el contenido multimedia sincronizado sea una alternativa al texto y esté claramente identificado como tal.

• Criterio de conformidad 1.2.2: Captions. Sonido grabado dentro de un contenido multimedia sincronizado: Proporcionar subtítulos para el contenido, excepto si el contenido multimedia sincronizado sea una alternativa al texto y esté claramente identificado como tal.

• Criterio de conformidad 1.3.1: Información y relaciones. La información, estructura y relaciones comunicadas a través de la presentación pueden ser determinadas por software o están disponibles como texto.

• Criterio de Conformidad 1.3.2: Secuencia significativa. Cuando la secuencia en la que se presenta el contenido afecta su significado, se puede determinar por software la secuencia correcta de lectura.

• Criterio de conformidad 1.3.3: Características sensoriales. Las instrucciones proporcionadas para entender y manejar el contenido no dependen exclusivamente de las características sensoriales de los componentes como su forma, tamaño, ubicación visual, orientación o sonido

• Criterio de conformidad 1.4.1: Uso de color. No utilizar el color como el único medio visual para transmitir la información, indicar una acción, solicitar una respuesta o distinguir un elemento visual.

• Criterio de conformidad 1.4.2: Audio Control. Si el sonido de una página web suena automáticamente durante más de tres segundos, proporcionar alguna de las siguientes opciones:

- Un mecanismo para pausar o detener el sonido.

- Un mecanismo para controlar el volumen del sonido que sea independiente del nivel de volumen global del sistema.

• Criterio de conformidad 2.1.1: Accesible por teclado. Es posible manejar toda la funcionalidad del contenido a través del teclado sin necesidad de alcanzar una determinada velocidad de pulsaciones de teclas, excepto cuando la función interna requiere que la introducción de datos dependa del camino trazado por el puntero.

• Criterio de conformidad 2.1.2: Sin trampas para el teclado. Si es posible mover el foco a un componente de la página usando el teclado, entonces el foco se puede quitar de ese componente usando solo el teclado y, si se requiere algo más que las teclas de dirección o de tabulación, informar al usuario el método apropiado para mover el foco.

• Criterio de conformidad 2.2.1: Tiempo ajustable. Para los límites de tiempo impuestos por el contenido, alguno de los siguientes casos es verdadero:

- El usuario puede eliminar el límite de tiempo antes de alcanzarlo.

- El usuario puede ajustar el límite de tiempo antes de alcanzar dicho límite en un rango amplio de tiempo (al menos 10 veces mayor al tiempo fijado originalmente).

- Advertir al usuario antes de que el tiempo expire y se le conceden al menos 10 oportunidades de 20 segundos para ampliar el límite con una acción simple (por ejemplo, presionar una barra espaciadora).

- Excepción tiempo real: el límite de tiempo es un requisito que forma parte de un evento en tiempo real (por ejemplo una subasta) y no resulta posible ofrecer una alternativa al límite de tiempo.
- Excepción esencial: el límite de tiempo es esencial, y si se expandiera, invalidaría la actividad.
- Excepción 20hs: el límite de tiempo es mayor a 20hs.
- Criterio de conformidad 2.2.2: Poner pausa, detener, ocultar. Para la información que se mueve, parpadea, se desplaza o se auto-actualiza, lo siguiente es verdadero:
 - Para cualquier información que se desplaza, parpadea o comienza automáticamente, dura más de 5 segundos y se presenta en paralelo con otro contenido, existe un mecanismo para que el usuario pueda pausar, detener u ocultar esto, excepto si es una parte esencial de una actividad.
 - Para cualquier información que se auto-actualiza, comienza automáticamente o se presenta en paralelo con otro contenido, existe un mecanismo para que el usuario pueda poner en pausa, detener u ocultar o controlar la frecuencia de actualización, excepto si la actualización automática es una parte esencial de una actividad.
- Criterio de conformidad 2.3.1: Umbral de tres destellos o menos. Las páginas no contienen nada que destelle más de tres veces en un segundo, o el destello está por debajo del umbral de destello general y de destello rojo.
- Criterio de conformidad 2.4.1: Evitar bloques. Habilitar un mecanismo que permita a los usuarios evitar bloques de contenido que se repiten en múltiples páginas web.
- Criterio de Conformidad 2.4.2: Titulado de página. Las páginas web tienen títulos que describen su temática o propósito.
- Criterio de conformidad 2.4.3 : Orden del foco. Si se puede navegar secuencialmente por una página web y la secuencia de navegación afecta a su significado o a su manejo, los componentes reciben el foco en un orden que preserva su significado y su operatividad
- Criterio de conformidad 2.4.4: Propósito del enlaces (en contexto). El propósito de cada enlace puede ser determinado con solo el texto del enlace o a través del texto del enlace sumado al contexto del enlace determinado por software, excepto cuando el propósito del enlace resulte ambiguo para los usuarios en general
- Criterio de conformidad 3.1.1: Idioma de la página. El idioma predeterminado de la página web puede ser detectado por software.
- Criterio de conformidad 3.2.1: En foco. Cuando cualquier componente recibe el foco, no se inicia ningún cambio en el contexto.
- Criterio de conformidad 3.2.2: En entradas. El cambio de estado en cualquier comportamiento de la interface de usuario no provoca automáticamente un cambio en el contexto a menos que el usuario haya sido advertido de ese comportamiento antes de usar el componente.
- Criterio de conformidad 3.3.1: Identificación de errores. Si se detecta automáticamente un error en la entrada de datos, identificar el elemento con el error y describir el error al usuario con un texto.
- Criterio de conformidad 3.3.2: Etiquetas o instrucciones. Proporcionar etiquetas o instrucciones cuando el contenido requiera la introducción de datos por parte del usuario.
- Criterio de conformidad 4.1.1: Procesamiento. En los contenidos implementados mediante el uso de lenguaje de marcas, los elementos tienen las etiquetas de apertura y cierre

completas; Los elementos están anidados de acuerdo a sus especificaciones, los elementos no contienen atributos duplicados y los identificadores son únicos.

Como se puede ver, de los 24 criterios de conformidad de Nivel A que busca alcanzar este proyecto:

- Dieciocho mejoran la navegación de usuarios con discapacidad visual⁶ (Los criterios dirigidos a la correcta funcionalidad de la página manejada desde el teclado mejoran también la navegación a usuarios con alguna limitación física o Motriz).
- Dos están dirigidas a enfermedades cognitivas o neurológica (trastorno de déficit de atención o falta de memoria)⁷.
- Dos mejoran la navegación de usuarios con discapacidad auditiva⁸.
- Una mejora la navegación de las personas que sufren de epilepsia⁹.
- Una mejora la comprensión de la página por parte de máquinas (motores de búsqueda por ejemplo)¹⁰.

Se hace evidente que el correcto funcionamiento de la página mediante el teclado, sumado a una correcta interpretación de parte del lector de pantalla, más la correspondiente información alternativa en caso de elementos multimedia son el pilar de una web accesible.

Más de Quinientas Setenta Técnicas Suficientes y Recomendables

Para cada una de los criterios de conformidad, se han documentado también una amplia variedad de técnicas. Las técnicas son informativas, ninguna técnica es obligatoria, ya que

⁶ Criterios de Conformidad 1.1.1, 1.2.3, 1.3.1, 1.3.2, 1.3.3, 1.4.1, 2.1.1, 2.1.2, 2.4.1, 2.4.2, 2.4.3, 2.4.4, 3.1.1, 3.2.1, 3.2.2, 3.3.1 y 3.3.2.

⁷ Criterios de Conformidad 2.2.1, 2.2.2.

⁸ Criterio de Conformidad 1.2.1 y 1.2.2.

⁹ Criterio de conformidad 2.3.1.

¹⁰ Criterios de Conformidad 4.1.1.

las WCAG 2.0 permiten crear mecanismos alternativos para superar el criterio. Hay 2 categorías de técnicas: las suficientes y las aconsejables. Las primeras son técnicas para que se consiga superar el criterio de conformidad, mientras que las segundas son recomendaciones para mejorar la accesibilidad y usabilidad del sitio Web.

A continuación se nombran algunas de las técnicas documentadas por la W3C [44] y que se han tenido en cuenta para la realización del proyecto:

- Técnicas suficientes para el criterio de conformidad 1.1.1

- Situación A - Si una descripción corta puede cumplir el mismo objetivo y presentar la misma información que el contenido no textual: Utilizar una de las técnicas de descripción corta (Utilizar el atributo "alt").

- Situación B - Si una descripción corta no puede cumplir el mismo objetivo y presentar la misma información que el contenido no textual (Ej. Gráfica o diagrama), utilizar una de las técnicas de descripción corta y una de descripción larga (Utilizar el atributo "longdesc" en imágenes).

- Situación C - Si el contenido no textual es un control o acepta una entrada por parte del usuario: Asociar una etiqueta al control del formulario con el elemento "label".

- Técnicas recomendables del criterio de conformidad 1.1.1.

Técnicas generales para informar el contenido no textual

- Identificar el contenido informativo no textual
- Hacer breve la descripción corta.
- Ofrecer el mismo texto que se incluye en las imágenes de texto.
- Enlazar a información en texto que ofrezca la misma información que el contenido no textual.

Técnicas generales para minimizar la barrera de los CAPTCHA

- Ofrecer más de dos modalidades de CAPTCHA o acceso directo al personal de atención al cliente.
- No pedir un CAPTCHA a usuarios que ya se han identificado.

Técnicas HTML

- Diseñar para navegadores que no soporten el elemento "frame".
- Ofrecer contenido alternativo para el elemento "iframe", pero no usar descripciones largas.

Técnicas CSS

- Formatear la página con las reglas de caja de CSS en lugar de con imágenes.
- Incluir imágenes decorativas con la propiedad "background".
- Mostrar las celdas vacías de las tablas.

Técnicas WAI-ARIA

- Identificar que elementos son meramente de presentación con el rol "presentation" de ARIA.
- Técnicas suficientes para el criterio de conformidad 1.2.1
 - Situación A: Si el contenido es sonoro solo grabado: Ofrecer una alternativa que presente la misma información.
 - Situación B: Si el contenido es video solo grabado: Ofrecer una alternativa que presente la misma información; u ofrecer un contenido sonoro que describa el contenido más importante del video.
- Técnicas recomendables del criterio de conformidad 1.2.1
 - Ofrecer una transcripción del contenido sonoro.
 - Enlazar a textos que ofrezcan la misma información.

- Técnicas suficientes para el criterio de conformidad 1.2.2

- Proveer de subtítulos para sordos de forma abierta¹¹.

- Técnicas recomendables del criterio de conformidad 1.2.2

- Si un video no tiene sonido avisar diciendo "video sin sonido".

- Técnicas suficientes para el criterio de conformidad 1.2.3

- Añadir una segunda pista de sonido, seleccionable por el usuario, que incluya la audiodescripción.

- Técnicas suficientes para el criterio de conformidad 1.3.1

Situación A: Si la tecnología ofrece una estructura semántica para comunicar la información y las relaciones:

- Formatear la estructura y el texto con elementos semánticos.

- Comunicar con texto la información que es transmitida por las variaciones en la presentación de texto.

- Separar la información y la estructura de la forma de presentación para permitir al usuario disponer de diferentes presentaciones, por ejemplo, utilizando sus propias hojas de estilo.

- Comunicar la información y sus relaciones mediante:

Respecto a la estructura de la página web, utilizar:

- Los elementos H1-H6 para identificar encabezados.

- Los elementos "ol", "ul" y "dl" para listas.

- El marcado semántico cuando se usan claves de color (por ejemplo rojo para error).

- Las funciones del DOM¹² para añadir

¹¹ Están incrustados en la propia imagen de video, por lo que no se puede alterar ni ocultar.

contenido a la página mediante programación.

En tablas de datos, utilizar:

- El elemento table para presentar información tabular.
- El elemento caption para titular la tabla de datos.
- El elemento summary para resumir la tabla de datos.
- El atributo scope para asociar celdas de encabezado con celdas de datos.
- Atributos id y header para asociar celdas de datos con celdas de encabezados.

En formularios, utilizar

- El elemento label para asociar etiquetas de texto con controles de formulario.
- El atributo title para identificar controles de formulario cuando no se puede usar label.
- Los elementos fieldsets y legend para describir grupos de controles de formulario.
- El atributo optgroup para agrupar elementos option dentro de un elemento select.

Situación B: Si la tecnología usada no ofrece una estructura semántica para comunicar la información y sus relaciones:

- Comunicar con texto la información que es transmitida por las variaciones en la presentación de texto.
- Hacer que la información y sus relaciones transmitidas mediante la presentación sean determinadas por software o estén disponibles en texto usando las convenciones estándares de formato de texto para párrafos, listas y

encabezados.

• Técnicas recomendables del criterio de conformidad 1.3.1

- Maquetar con CSS, y no con tablas.
- Ubicar las etiquetas para maximizar la predictibilidad de las relaciones entre estas y los controles a los que se representan.
- Ofrecer etiquetas para todos los controles de formulario que no tienen etiquetas implícitas.
- Organizar la página usando encabezados.

• Técnicas suficientes para cumplir el criterio de Conformidad 1.3.2

Ordenar todos los contenidos de la página web en un orden que tenga sentido y maquetarlos correctamente:

- Usar marcas de Unicode¹³ para mezclar texto "de izquierda a derecha" y "de derecha a izquierda".
- Añadir el atributo dir en elementos de línea para resolver problemas con diferentes direcciones anidadas.
- Ubicar el contenido según el marcado de estructuras.
- Controlar el espacio entre letras mediante CSS.

• Técnicas recomendables del criterio de conformidad 1.3.2

- Alinear el texto a la izquierda en idiomas que se escriban de izquierda a derecha; e inversamente.
- Ofrecer un enlace a una presentación lineal del contenido
- Ofrecer cambiar el estilo de la presentación de los elementos mediante hojas de estilo que cambien el orden de presentación.

• Técnicas suficientes para cumplir el criterio de conformidad 1.3.3

¹² Es una interface de programación para cambiar dinámicamente contenido estructurado en documentos.

¹³ Estándar de codificación de caracteres.

- Ofrecer identificación textual de los elementos que de otra manera solo dependerían de la información sensorial para ser comprendidos (por ejemplo, "pulsa el cuadro grande para continuar")

- Técnicas recomendables del criterio de conformidad 1.3.3

- Incluir los símbolos gráficos mediante imágenes con una alternativa textual y evitar los glifos de las fuentes Unicode.

- Técnicas suficientes para cumplir el criterio de conformidad 1.4.1

Situación A: Si se comunica información mediante el color en palabras, fondos u otros contenidos:

- Comprobar que la información presentada por colores está disponible en texto.

Incluir una señal de texto para etiquetas de controles de formulario coloreadas.

- Comprobar que hay señales visuales adicionales cuando las diferencias de color en el texto contienen información.

- Cuando el color indica que el foco se posa en un enlace o en un control, garantizar un ratio de contraste mínimo de 3:1 con el contenido adyacente y ofrecer señales visuales adicionales.

Situación B: si se comunica información mediante el color dentro de una imagen:

- Diferenciar los datos mediante colores y motivos de dibujo.

- Comprobar que la información comunicada por las diferencias de color esté también disponible en el texto.

- Técnicas recomendables del criterio de conformidad 1.4.1

- Destacar y redundar la información transmitida en texto con colores.

- Cambiar la presentación de los componentes de la interfaz de usuario cuando recibe el foco mediante CSS.

- Técnicas suficientes para cumplir el criterio de conformidad 1.4.2

- Si un sonido se reproduce automáticamente, apagarlo también automáticamente antes de los 3 segundos.

- Ofrecer control al usuario en el inicio de la página web para que pueda apagar los sonidos que suenan automáticamente.

- Reproducir los sonidos solo a petición del usuario.

- Técnicas recomendables del criterio de conformidad 1.3.2

- Ofrecer una opción que permita apagar el sonido de todo el sitio web.

- Técnicas suficientes para cumplir el criterio de conformidad 2.1.1

- Identificar todas las funcionalidades del contenido, controles de formulario y enlaces; y comprobar que pueden ser manejados solo con el teclado.

- Ofrecer manejadores de eventos de teclado funciones específicas de teclado y de otro dispositivo, haciendo que el evento on click en enlaces y botones sea accesible por teclado o haciendo redundante los eventos de ratón y teclado.

- Técnicas recomendables del criterio de conformidad 2.1.1

- Emplear los atributos role, state y value si quieres que los elementos estáticos de la interfaz de usuario se comporten como componentes interactivos en XHTML.

- Ofrecer teclas rápidas para acceder a los enlaces y los controles de formulario más importantes de la página.

- Comenzar cada elemento de la lista con una combinación de letras diferenciadoras.

- Al utilizar programación del lado cliente, elegir el manejador de eventos más abstracto posible, preferentemente el evento on active.

- Si se va a utilizar combinaciones de teclas para efectuar acciones, elegir combinaciones de teclas que no sean las específicas que utilizan los agentes de usuario para otros objetivos.

- Técnicas suficientes para cumplir el criterio de conformidad 2.1.2

- Comprobar que los usuarios no queden atrapados en el contenido

- Técnicas suficientes para cumplir el criterio de conformidad 2.2.1

Situación A: Si hay un límite de tiempo de sesión.

- Ofrecer un mecanismo en la primera página de un formulario que se distribuye en varias páginas que permita a los usuarios solicitar un mayor límite de tiempo o eliminarlo.

- Ofrecer un mecanismo que permita a los usuarios anular el límite de tiempo.

Situación B: Si el límite de tiempo es controlado por programación en la página.

- Ofrecer un mecanismo que permita a los usuarios anular el límite de tiempo.

- Ofrecer un medio para aumentar el límite de tiempo 10 veces al límite predefinido.

- Avisar al usuario cuando el límite de tiempo esté próximo a expirar; y permite que el usuario pueda aumentar el límite predefinido.

Situación C: si hay un límite de tiempo para leer los contenidos.

- Permitir a los usuarios pausar el contenido y volver a reproducirlo desde donde fue parado.

- Ofrecer una manera para que el usuario quite el límite de tiempo.

- Desplazar el contenido de forma automática mediante programación, y ofrecer un mecanismo para pausarlo.

- Ofrecer un mecanismo que permita a los usuarios mostrar un texto en desplazamiento o que se actualice, en una ventana estática o en un área de la página.

- Técnicas recomendables del criterio de conformidad 2.2.1

- Usar programación para preguntar al servidor si existe un límite de tiempo; y notifica al usuario en caso que no exista.

- Utilizar de sonido para enfocar la atención del usuario

- Técnicas suficientes para cumplir el criterio de conformidad 2.2.2

- Permitir al usuario pausar el contenido y recomenzar desde donde fue pausado

- Desplazar el contenido de forma automática mediante programación, y ofrecer un mecanismo para pausarlo.

- Crear contenido que parpadee menos de 5 segundos.

- Si usamos una tecnología para incluir contenido parpadeante, el usuario puede apagarlo a través del agente de usuario.

- Diseñar los gif animados¹⁴ para que paren de parpadear después de n ciclos reproducidos en 5 segundos.

- Controlar el parpadeo con programación y páralo en 5 segundos o menos.

- Ofrecer un mecanismo que detenga el movimiento, el parpadeo o el refresco del contenido.

- Ofrecer un mecanismo para recargar la página sin contenidos parpadeantes.

- Técnicas recomendables del criterio de

¹⁴Formato sin pérdida de calidad para imágenes.

conformidad 2.2.2

- Ofrecer un mecanismo para detener todos los contenidos que parpadeen o que estén en movimiento incluso si se detiene automáticamente dentro de 5 segundos.

- Técnicas suficientes para cumplir el criterio de conformidad 2.3.1

- Comprobar que ningún componente del contenido destella más de 3 veces por segundo.

- Hacer que el contenido destellante ocupe un espacio pequeño. El mayor tamaño aceptable es de 21 píxeles cuadrados en una pantalla de 1024 x 768.

- Asegurar que el contenido no viola el umbral del destello general o el umbral de destello rojo.

- Técnicas recomendables del criterio de conformidad 2.3.1

- Reducir el contraste de cualquier contenido destellante.

- Reducir el número de destellos incluso si estos no violan el umbral.

- Ofrecer un mecanismo para suprimir cualquier contenido destellante antes de que empiece.

- Reducir la velocidad de los videos en directo para evitar destellos rápidos.

Si se detectan 3 destellos dentro de un segundo:

- Congelar la imagen momentáneamente.

- Reducir el ratio de contraste.

- Técnicas suficientes para cumplir el criterio de conformidad 2.4.1

Evitar al usuario tener que pasar por bloques de contenido o elementos repetidos con una de estas opciones:

- Un enlace en el inicio de la página para ir directamente al contenido principal o a cada bloque de contenido.

- Un enlace en el inicio de cada bloque repetido para ir al final de ese bloque.

- Técnicas recomendables del criterio de conformidad 2.4.1

- Ofrecer acceso mediante teclado a los enlaces y los controles de formulario importantes.

- Ofrecer enlaces directos al contenido para mejorar la navegación en la página.

- Ofrecer teclas de acceso directo.

- Elegir técnicas que permitan a los agentes de usuario y productos de apoyo navegar por la estructura.

- Técnicas suficientes para cumplir el criterio de conformidad 2.4.2

- Dotar a todas las páginas de títulos que describan su temática o propósito.

- Técnicas recomendables del criterio de conformidad 2.4.2

- Ofrecer un título único y descriptivo para cada página web.

- Técnicas suficientes para cumplir el criterio de conformidad 2.4.3

- Situar los elementos inactivos en una secuencia que siga el orden y las relaciones dentro del contenido

- Situar el foco en los elementos en un orden secuencial. Elegir una de estas opciones: Crear un orden lógico de tabulación entre enlaces, controles de formulario y objetos o hacer que el orden del DOM sea igual que el orden visual.

- Si queremos que una página cambie dinámicamente, elegir una de estas opciones: Insertar contenido dinámico en el DOM inmediatamente después del elemento disparador o crear diálogos personalizados de forma independiente al dispositivo.

- Reordenar las secciones de la página con el DOM.

- Técnicas recomendables del criterio de conformidad 2.4.3

- Destacar claramente los enlaces o controles cuando reciban el foco por teclado.
- Si puede haber problemas para que los usuarios reconozcan la secuencia de navegación debido al diseño visual de la página, ofrecer al usuario un mecanismo para usar un orden de presentación alternativo más sencillo y evidente.

- Técnicas suficientes para cumplir el criterio de conformidad 2.4.4

Identificar el objetivo de un enlace usando:

- Un texto clarificador en el enlace.
- Un texto en el enlace combinado con el texto de oración que lo engloba; o
- Un texto de enlace combinado con su contexto de forma que sea determinado por software.
- Describir el objetivo de un enlace. Si se necesita ocultar texto del enlace con CSS, hacerlo.

- Técnicas recomendables del criterio de conformidad 1.3.2

- Mostrar tooltips¹⁵ cuando los elementos reciben el foco.
- Si hay una imagen y un texto adyacente que enlace al mismo sitio, combinarlos.
- Ofrecer etiquetas descriptivas de los controles de formulario con la propiedad "aria-describedby" de ARIA.

- Técnicas suficientes para cumplir el criterio de conformidad 3.1.1

- Identificar el idioma o los idiomas predeterminados usando el atributo lang en los elementos HTML.

- Técnicas recomendables del criterio de

conformidad 3.1.1

Especificar el idioma predeterminado de todas las páginas usando:

- En el atributo lang de la etiqueta "http".
- En el atributo content-language de la etiqueta meta.

- Técnicas suficientes para cumplir el criterio de conformidad 3.2.1

- Usar "Active" en lugar de focus como disparador de cambios de contexto. Observar que se habla de cambios de contexto, no de cambios de contenido.

- Técnicas recomendables del criterio de conformidad 3.2.1

- Evitar cambiar de estado o valores de forma permanente cuando un componente recibe el foco, o bien ofrecer un mecanismo para restablecer cualquier cambio.
- Abrir ventanas nuevas solo cuando sea la mejor alternativa y avisa siempre a los usuarios.

- Técnicas suficientes para cumplir el criterio de conformidad 3.2.2

- Para iniciar un cambio de contexto, ofrecer un botón que el usuario pueda usar.
- Si un cambio en un control de formulario causa un cambio de contexto, describirlo antes que suceda.

- Técnicas recomendables del criterio de conformidad 1.3.2

- Avisar que se va a abrir una nueva ventana.

- Técnicas suficientes para cumplir el criterio de conformidad 3.3.1

Situación A: Si el formulario contiene información obligatoria que el usuario no complete:

- Validar el formulario en el lado cliente.

¹⁵Una descripción emergente.

- Identificar los campos que no han sido completados.
- Avisar al usuario con un mensaje de texto explicando el error.

Situación B: Si la información que el usuario brinda debe seguir un formato correcto o debe estar en un rango de valores determinado:

- Validar el formulario en el lado cliente.
 - Añadir un mensaje de error con el DOM y;
 - Avisar al usuario con un mensaje de texto explicando el error.
- Técnicas recomendables del criterio de conformidad 3.3.1
 - Si hay varios errores, volver a mostrar el formulario con un resumen de los errores y ofrecer un mecanismo para que los usuarios puedan saltar de uno a otro.
 - Advertir de los errores en el momento que el usuario introduzca la información y enfatizar visualmente los errores donde sucedan.
 - Advertir de los errores con texto y con contenido no textual.
 - Validar el envío del formulario en el lado de cliente y también en el lado del servidor.
 - Tanto si el formulario ha sido enviado con éxito como si no, hacérselo saber al usuario.

- Técnicas suficientes para cumplir el criterio de conformidad 3.3.2

Ofrecer etiquetas label descriptivas y una de las siguientes opciones:

- Indicar el formato esperado de los datos y poner un ejemplo.
- Dar instrucciones al comienzo del formulario o de un grupo de campos.
- Asociar etiquetas de texto con controles de formulario y ubicarlas

para maximizar la predictibilidad de la relación.

- Identificar los campos obligatorios con texto.
- Técnicas recomendables del criterio de conformidad 3.3.2
 - Si hay cambios de contexto cuando el usuario cambia un control de formulario, avisar antes al usuario.
 - Usar ARIA para ofrecer una etiqueta descriptiva determinable por software.
 - Usar ARIA para identificar los campos del formulario.
 - Ofrecer un diseño lineal del formulario y agrupa elementos similares

- Técnicas suficientes para cumplir el criterio de conformidad 4.1.1

- Elegir el estándar que desees (HTML, XHTML, XML), pero seguir su especificación, poniendo atención al inicio y cierre de etiquetas y sus atributos.
- Comprobar que la página web puede ser procesada y que está bien formada.
- Comprobar que cada atributo id tiene un valor diferente en la página web.
- Comprobar que los elementos no tienen atributos repetidos.
- Validar la página web y verifica que cumplen completamente las especificaciones.

2.2.2. Independencia Tecnológica

Como ha sido mencionado previamente, en la nueva versión de las Pautas de Accesibilidad no hay ninguna mención acerca de cuáles tecnologías son accesibles y cuáles no, generando una independencia tecnológica.

Esta independencia tecnológica trajo dos nuevos conceptos: la "tecnología

posibilitante del uso accesible” y la “dependencia tecnológica” [43].

Tecnología Posibilitante de Uso Accesible

Una tecnología de contenido Web posibilita el uso accesible, cuando los usuarios de los productos de apoyo pueden trabajar con esa tecnología, y las características de accesibilidad de las tecnologías principales y mayoritarias funcionan con la misma.

La clave es cómo las tecnologías incluyen los mecanismos necesarios para que los agentes de usuario, incluyendo los productos de apoyo, puedan entenderlas y manejarlas. En otras palabras, una tecnología posibilitante de uso accesible brinda viabilidad para que los productos de apoyo puedan acceder al contenido, presentarlo al usuario, y éste, pueda percibirlo y manipularlo. Incluso, es posible que sea necesario manipular los agentes de usuario o los productos de apoyo para dar soporte a esa tecnología posibilitante de uso accesible (por ejemplo, descargando un plugin).

Dependencia Tecnológica

El concepto de dependencia tecnológica hace referencia a que cuando una página Web utiliza una tecnología determinada para mostrar el contenido, si el agente de usuario no reconoce o no soporta esa tecnología, o si esta desactivada, el contenido no se mostrará. Por ejemplo, una Web en flash no puede ser visualizada en un Iphone, pero sí en la mayoría de las pc de escritorio. Esto no es obstáculo para que flash pueda ser calificada como una tecnología posibilitante; pero genera una dependencia tecnológica.

2.2.3. Tecnologías Accesibles y no Accesibles

La neutralidad (No definen ninguna tecnología para concretar una página web accesible) expuesta en las WCAG 2.0, trae consigo la imposibilidad de afirmar que una tecnología es accesible o que no lo es. Esta neutralidad, genera algunos temas importantes que hay que saber⁴³:

- No es posible especificar hasta qué punto debe llegar una tecnología para calificarla de posibilitante de uso accesible.

No es posible cuantificar cuántos agentes de usuario, en cuantas versiones ni para cuáles productos de apoyo se debe respaldar una tecnología, ni tampoco en cuantos idiomas o dialectos debe estar traducido el agente de usuario para posibilitar una tecnología.

- Se puede utilizar una tecnología de una forma no accesible, pero ofreciendo una versión alternativa cumpliendo así con las pautas.

Es un recurso válido. Si no es posible, o es complejo brindar acceso a determinada información a todos los usuarios de la misma forma, es válido utilizar un mecanismo para determinados usuarios y otro mecanismo para otro grupo de usuarios. Por ejemplo, podemos utilizar Google maps¹⁶ (herramienta desarrollada en javascript) que propone funcionalidades inaccesibles para mostrar los bares en la ciudad de Santa Fe y a la vez ofrecer un enlace a una página con la misma información textual, listando los bares y describiendo la dirección en la que se encuentran. De este modo javascript sería la tecnología no accesible y con un enlace a otra página que muestra de otro modo la información estaríamos ofreciendo una versión alternativa para cumplir con las pautas.

- Se puede utilizar una tecnología accesible de manera inaccesible.

Es muy probable que desarrolladores que no conozcan de Accesibilidad Web, estén trabajando de esta manera. Aunque la tecnología proporcione mecanismos para ser accesibles, estos no estén poniendo en práctica.

2.2.4. Declaraciones de Conformidad

Cada desarrollo es responsable por sí mismo de verificar si se está cumpliendo

¹⁶ Google Maps es un servidor de aplicaciones de mapas en la web.

con las pautas de la WAI y a qué nivel [44]
 44. Para ello la W3C ofrece un juego de iconos que van unidos a una declaración de conformidad y exteriorizan el nivel alcanzado:




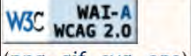


A	AA	AAA
 (png, gif, svg, eps)	 (png, gif, svg, eps)	 (png, gif, svg, eps)
 (png, gif, svg, eps)	 (png, gif, svg, eps)	 (png, gif, svg, eps)

Imagen 1. Juego de logos proporcionados por la w3c para el nivel de conformidad.

Ni los iconos, ni las declaraciones de conformidad son obligatorias sino recomendados; pero, si los iconos son utilizados, se debe incluir también la declaración de conformidad.

- Información Obligatoria en las Declaraciones de Conformidad
 - Fecha de la declaración de conformidad.
 - Pautas que se han seguido: título, versión y URL.
 - El nivel de conformidad alcanzado: A, AA, o AAA.
 - El ámbito al que se aplica la declaración, es decir, las páginas que cumplen la conformidad.
 - Las tecnologías Web usadas que generan dependencia.
- Información Opcional
 - Criterios de conformidad superiores al nivel declarado que se han cumplido.
 - Las tecnologías Web usadas que no generan dependencia.
 - Agentes de usuario y productos de apoyo con los que se ha probado la página.
 - Pasos dados (más allá de los indicados en las normas) para mejorar la accesibilidad.

2.2.5. Conformidad Parcial

La W3C ofrece dos posibles soluciones para páginas que van a ser actualizadas por muchas personas y no es posible controlar todo el contenido que se ingresan en el sistema. La primera opción dice que, si es posible monitorizar y reparar los errores del contenido externo del sitio Web en 2 días laborales, se puede usar la declaración de conformidad normal. La segunda opción dice que si no es posible monitorizar y reparar los errores, se puede utilizar una declaración de conformidad parcial de contenidos procedentes de fuentes no controladas.

La declaración parcial de contenidos se puede utilizar solamente en el caso que el contenido sea actualizado por muchas personas, y la forma de hacerlo es añadir a la declaración una lista detallada de los contenidos:

"Esta página no cumple el nivel [A, AA, AAA] de la WCAG 2.0, pero podría cumplirlo si se quitan los siguientes contenidos procedentes de fuentes no controladas..."

Otra figura que contempla la WCAG 2.0 es la declaración de conformidad parcial debido al idioma. Esta declaración se puede usar cuando la tecnología utilizada no posibilita el uso accesible con el idioma empleado en la página. Si se toma como ejemplo un sitio Web con versiones en español y chino, por limitaciones de la tecnología HTML, los atributos "alt" de las imágenes solo se mostrarían en español. Por lo tanto, los usuarios de la versión china no podrán comprender la descripción de la imagen. La forma de indicar esta conformidad parcial sería la siguiente:

"Esta página no cumple el nivel [A,AA,AAA] de las WCAG 2.0, pero podría cumplirlo si se hubiera posibilitado el uso accesible para el idioma".

2.3. Pautas de Accesibilidad para Agentes de Usuario

Los agentes de usuario son los programas

utilizados por la personas para interactuar con sus computadoras y los documentos técnicos de las Pautas de Accesibilidad para Agentes de Usuario (UAAG) están desarrollados por el grupo de trabajo de las Pautas de Accesibilidad para Agentes de Usuario. Estos documentos explican cómo crear agentes de usuario accesibles para personas con discapacidad y sirven fundamentalmente como guía a desarrolladores de navegadores¹⁷, media players¹⁸, asistentes tecnológicos y otros agentes de usuario, pero además puede tener lectores con otros intereses, como ser aquellos interesados en:

- Evaluar la accesibilidad de un agente de usuario.
- Fomentar la accesibilidad de un agente de usuario, exigiéndoles aspectos de accesibilidad en futuras versiones [29].

2.4. Pautas de Accesibilidad para las Herramientas de Autor

Las Pautas de Accesibilidad para las Herramientas de Autor (ATAG) herramientas útiles para desarrolladores, autores y diseñadores. Las ATAG contiene dos partes importantes: la parte A refiere a cómo crear herramientas de autor accesibles y la parte B refiere a cómo ayudar con las herramientas de autor a producir contenido accesible.

Para producir contenido Web, las pautas explican lo siguiente:

- Cómo hacer las herramientas de autor accesibles para que las personas con determinadas discapacidades puedan crear contenido Web.
- Cómo crear contenido más accesible; promover y fomentar la producción de contenidos conforme a las WCAG.

ATAG está especialmente dirigida a desarrolladores de herramientas de autor, incluidos:

¹⁷ Es un programa de software instalado en tu ordenador que nos permite visitar páginas Web y utilizar aplicaciones.

¹⁸ Es un programa dedicado a la reproducción de archivos multimedia.

- Herramientas para la creación de páginas Web, por ejemplo: editores HTML.
- Software para generar sitios Webs, por ejemplo: CMS¹⁹.
- Editores de textos convencionales que proporcionan la posibilidad de guardar el contenido como HTML, por Ejemplo: Word.
- Autores de herramientas multimedia.
- Páginas Webs que les permitan a los usuarios agregar contenidos, por ejemplo: blogs, wiki, sitios para compartir fotos.

3. Estado del Arte de las Tecnologías Web

A continuación se detallarán los aspectos más relevantes relacionados con la accesibilidad de las tecnologías con las cuales se aconseja trabajar a la hora de buscar el cumplimiento de las Pautas de Accesibilidad descritas en el capítulo anterior. Si bien existen innumerable cantidad de tecnologías involucradas en un proyecto Web completo, este proyecto se limita a las involucradas más habitualmente²⁰ en la interface del usuario.

3.1. HTML5

El lenguaje HTML actualmente es un estándar reconocido en todo el mundo y cuyas normas define el organismo W3C. Este consorcio, lanzó HTML5 como la nueva versión del lenguaje de marcado, y no sólo trata de incorporar nuevas etiquetas o eliminar otras de la versión anterior, sino que propone mejoras en áreas que hasta ahora quedaban fuera del lenguaje como lo relacionado con la incorporación de multimedia o validación de formularios y para las que se necesitaba utilizar otras tecnologías [2].

¹⁹ Un sistema de gestión de contenidos (Content Management System) es un programa que permite la administración de contenidos de forma sencilla.

²⁰ <http://code-trends.appspot.com/timetrend?tags=html5%2Cflash%2Csilverlight>.

HTML5 constituye un avance hacia el diseño de sitios Web que favorecen a la igualdad de condiciones para todos los usuarios de la red. El nuevo estándar aporta funcionalidades que quizás no estén disponibles actualmente para todos los navegadores, pero, se han dado los primeros pasos para incluirlas a la Web. Otras funcionalidades por lo contrario, ya están plenamente incorporadas por los navegadores.

A continuación se exponen las herramientas que ofrece HTML5 para crear páginas más accesibles, brindando nuevas etiquetas, otorgando mayor semántica y nuevas opciones para agregar elementos multimedia.

3.1.1. Semántica

El objetivo de la semántica es jerarquizar y clasificar con precisión los datos de un documento, no es un concepto exclusivo de un lenguaje, sino una característica con la que pueden contar algunas herramientas. HTML5 ha incorporado muchas etiquetas que sirven para indicar qué son y cuál es el significado de los elementos que contienen dentro, ayudando a evitar el abuso de elementos `<div>` que habitualmente se utilizan con el fin de delimitar las partes de un documento. Esta nueva característica es relevante en aspectos de accesibilidad porque puede interconectar a las personas con los datos de una forma más eficiente, definiendo la naturaleza del contenido que se está mostrando. De esta manera por ejemplo, los asistentes tecnológicos pueden señalarle al usuario que dato exactamente se espera que ingrese (números, letras o un email).

Las etiquetas que posibilitan la mejora en la semántica y por ende en la accesibilidad se detallan a continuación [4]:

`<header>`

Este elemento envuelve la cabecera de cada documento o sección y es utilizado como contenedor de la introducción al contenido.

`<nav>`

Se utiliza esta etiqueta para identificar semánticamente la sección o secciones de navegación del sitio [3]. Como ejemplo de utilidad en lo que hace a mejorar la accesibilidad, se puede mencionar que si el navegador sabe delimitar qué parte es la que le corresponde al menú de navegación, y llegado el caso que se entienda que este no es necesario que permanezca visible (por lo menos hasta que el usuario lo pida nuevamente), se puede ocultar y dejar ese espacio disponible para mostrar el contenido de forma más grande o mejor distribuido. Esto es muy habitual en los dispositivos móviles donde el espacio es más reducido.

`<h1>`, `<h2>`, `<h3>`, `<h4>`, `<h5>`, `<h6>`

Estos son elementos que ya existían en versiones anteriores de HTML; su importancia se reduce a medida que aumenta su número. Las recomendaciones generales de semántica indican que el título principal debe ser de tipo `<h1>`, los subtítulos del tipo `<h2>` y así sucesivamente. La novedad en la nueva versión es que permite tener varios `<h1>` declarados en la página, uno en cada sección o artículo (dos etiquetas que se desarrollaran en este capítulo), mientras que en las validaciones automáticas en versiones anteriores del lenguaje, esto sería objeto de error. La importancia en términos de accesibilidad radica en que los usuarios de lectores de pantalla generalmente listan estos títulos (`<h1>`, `<h2>`, `<h3>`, etc.) al ingresar a la página para tener un panorama amplio de su contenido. Si no se respeta una escalada coherente hacia el mayor detalle en la información, el usuario puede entender erróneamente el contenido de la página.

`<main>`

Mediante esta etiqueta se dice que el contenido que se envuelve es el principal del documento, y por esta razón solo se debe especificar una vez. El propósito de esta etiqueta es aumentar la accesibilidad de la página, dando la posibilidad de evitar los elementos complementarios

(menú, barras laterales, pie de página) y saltar específicamente al contenido esencial de la página. Su implementación es fundamental en una página que pretende cumplir con las pautas de accesibilidad. Identificar el contenido principal de cada página posibilita a los asistentes tecnológicos evitar los bloques repetidos en todas las páginas y dirigir al usuario directamente hacia el contenido principal de la página.

<article>

Esta etiqueta sirve para delimitar una unidad independiente de contenido; es decir, se podría envolver con esta etiqueta algo que tenga sentido por sí mismo tanto en la página en cuestión como fuera de ella. Por ejemplo: una noticia completa, una guía, un artículo, etc.

<section>

Es una etiqueta usada para representar un bloque de una página que tenga valor semántico y agrupar contenido relacionado. La etiqueta crea una sección de manera explícita en el outline del documento²¹, por lo que se le debe siempre asignar un encabezado, ya que si no, se le aplicará directamente un title (sin título, en español). Según la W3G:

“El elemento <section> presenta una sección genérica de un documento o una aplicación. Una sección en este contexto es un grupo o contenido temático. El tema de la sección debe ser identificado típicamente con un elemento título”.

<aside>

<aside> debe ser usado para todo aquel contenido tangencial al contenido principal [5]. Esta etiqueta es recomendada para sidebar o notas.

<footer>

La etiqueta <footer> representa el pie de un documento o sección. La información que se suele añadir en este bloque es el autor del documento, enlaces a contenido

²¹ No es más que la lista de secciones que forma la página Web.

relacionado, información de copyright, avisos legales o cualquier otra información necesaria legalmente.

<details> y <summary>

La funcionalidad de <details> y <summary> es de la menos vistas e implementadas de HTML5 por los navegadores. La misma presenta una novedad en términos de interactividad sin javascript, y brinda la posibilidad de mostrar o esconder contenido solamente usando las etiquetas de forma correcta. La accesibilidad de una página que requiere javascript es menor que la de una que no lo requiere, fundamentalmente porque mejora su robustez.

Un posible diagrama o estructura general de una página escrita en HTML5 podría ser la siguiente:



Imagen 2 - Posible diagrama semántico de la página [6].

3.1.2. Formularios

En lo que respecta a formularios, HTML5 incluye muchas nuevas características que hacen que los formularios Web sean mucho más fáciles de escribir, y mucho más potentes y consistentes en la Web. La nueva versión del lenguaje pretende estandarizar algunos de los controles más comunes, eliminar la necesidad de soluciones cargadas de scripts y evolucionar hacia una Web más inclusiva.

Entre otros cambios, HTML5 ha incorporado nuevos eventos relacionados con elementos de formularios, ha mejorado la semántica incluyendo diferentes tipos de campos de entrada, ha incorporado herramientas para brindar ejemplos o información al usuario para evitar errores a la hora de completar el formulario, ha agregado validación nativa ahorrando horas de trabajo a los desarrolladores y ha homogeneizando este aspecto de la Web.

3.1.2.1. Entrada de texto

HTML5 ha añadido nuevos valores que son permitidos en el atributo `type` por lo que ahora los elementos de entrada serán más específicos, aumentando así la semántica. A continuación se detallan algunos de los diferentes valores que puede tomar el atributo `type` de la etiqueta `input` [7]:

`type="tel"`

Cuando un usuario intenta completar un `input` de tipo `tel.`, un teléfono inteligente como el iPhone, convierte su teclado a números. También se puede integrar esto a otras fuentes de datos como por ejemplo los números de teléfono de los contactos. También es posible exigir que el texto introducido cumpla las especificaciones del formato. Ya hemos comentado sobre las ventajas de ahorrar espacio en el momento que sea posible, reducir las opciones (de alfanuméricos a numéricos) implica poder brindar al usuario botones más grandes y mejorar la accesibilidad de usuarios con párkinson por ejemplo.

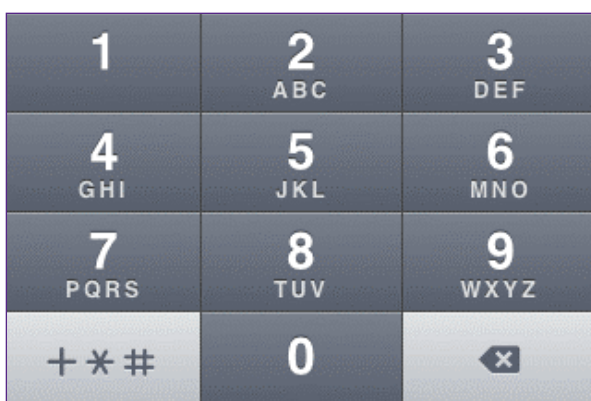


Imagen 3. Elemento `Input` de HTML5, con `type=phone`.

`type="email"`

En este caso, al especificar el tipo de entrada como `"email"`, se completa el teclado con la tecla `"@"`. Aquí también se puede integrar con otras fuentes de datos como por ejemplo los emails de los contactos. Siempre que las tecnologías de apoyo puedan interpretar el campo de entrada podrán ayudar a los usuarios a no cometer errores. Muchas veces los emails suelen ser complejos y contener caracteres tales como `"_"` o `"."`, las tecnologías de apoyo al saber que se trata de un email y no de un campo de texto habitual podrán interpretar que cuando un usuario está completando por voz un campo y dice `"guion bajo"` debe colocar `"_"` y no `"guion bajo"`.

Estos nuevos elementos vienen a sustituir al clásico valor `"text"` del atributo `type`. También están los tipos `"datetime-local"`, `"datetime"`, `"time"`, `"week"`, `"month"` y `"date"`, que pueden generar un significativo aumento en la accesibilidad, ya que se espera que los navegadores suministren herramientas homogéneas y sólidas para la selección de fecha y hora. Actualmente, lo más habitual es la utilización de JavaScript, en donde se suele tener deficiencia en cuanto al acceso por teclado y una variedad infinita de interfaces. Los tipos `"URL"`, `"email"` y `"tel"`, mejorarán ampliamente la usabilidad, dotando de opciones a los usuarios para completar los campos. La utilización en conjunto de los campos nombrados anteriormente disminuirá sin duda el tiempo de aprendizaje y mejorará la interacción entre el usuario y la página.

3.1.2.2. Marcador de Posición de Texto (placeholder)

Este atributo está ligado fuertemente a la accesibilidad [8]. Consiste en la colocación de un texto dentro del elemento de entrada y puede tener, por ejemplo, instrucciones relativas al completado del elemento. Actualmente, es posible de implementar un marcador de posición de texto mediante javascript, pero el uso generalizado del atributo

(en lugar de secuencias de comandos existentes) asegura que el texto del marcador de posición tendrá un aspecto y comportamiento uniforme a lo largo de la Web, incrementando así la accesibilidad y facilitando el aprendizaje de uso de los sitios Web. Su uso está incorporado en las versiones actuales de los navegadores, pero cada navegador lo implementa de forma distinta, al igual que los lectores de pantalla.



Ilustración 4 - Captura de pantalla formulario.

3.1.2.3. Campos Requeridos (required)

HTML5 brinda la posibilidad de definir que un campo es requerido añadiendo una característica directamente a la sintaxis [9], con la idea de que el navegador no sólo muestre los elementos requeridos de forma diferente, sino que proporcione automáticamente mensajes de error cuando alguien intenta enviar un formulario sin haberlo completado con todos los valores requeridos.

3.1.2.4. Patrones (pattern)

El atributo pattern de HTML5 incrementa las capacidades del atributo required anteriormente nombrado, al indicar que un campo de entrada particular debe tener un valor que coincida con un determinado patrón o cierta expresión regular definida por el desarrollador como valor del atributo pattern. Esto permite recibir los datos estrictamente con un formato predefinido o mostrar un mensaje

de error si se intenta enviar el formulario con datos en el campo con un formato incorrecto.

3.1.2.5. Enfoque Automático

El enfoque automático es un atributo booleano para los campos de un formulario. El mismo, permite especificar que un elemento debe recibir el foco tan pronto como se cargue la página, para que de esta manera, el usuario pueda empezar a escribir de inmediato. La capacidad puede ser obviamente dada también por javascript. Es muy probable que rápidamente se entienda el concepto ya que los buscadores utilizan la funcionalidad.

3.1.3. Multimedia

HTML5 introduce nuevos elementos relacionados con la multimedia, buscando simplificar y unificar criterios para la incorporación de este tipo de contenidos en la Web. Aunque se siguen encontrando algunos inconvenientes, estas incorporaciones han sido un gran avance.

Video

Hasta ahora, nunca ha habido un estándar para colocar videos en páginas Web, fundamentalmente porque los videos se convirtieron en un contenido de tipo esencial en Internet desde no hace mucho tiempo. Antes de HTML5, el método para incluir vídeo en un sitio Web requería un complemento (plugin) de terceros, dejando en manos de los desarrolladores la calidad, la interfaz de usuario y la accesibilidad del vídeo. La posibilidad de incorporar el elemento con la etiqueta estándar <video>, es sin duda una de las más relevantes incorporaciones de HTML5. Aunque actualmente no hay un acuerdo sólido sobre el Códec (codificador-decodificador) de video a utilizar, en teoría, el mercado lo ira definiendo.

Música

Además de <video>, el nuevo estándar de HTML, en lo que refiere a multimedia, ha incorporado la etiqueta <audio>. Similarmente a lo ocurrido con la etiqueta

<video>, también la etiqueta <audio> propone un estándar para incluir archivos multimedia en las páginas Web sin la necesidad de instalar ninguna otra aplicación externa. Esta etiqueta tiene prácticamente las mismas propiedades que <video>, y presenta también algunos inconvenientes en la definición del códec a utilizar.

Imágenes

HTML5 incorporó también el elemento figure <figure> que es utilizado para marcar el contenido que contiene imágenes, diagramas, ilustraciones, fotos y ejemplos de código. Está asociado opcionalmente con la etiqueta <figcaption>, la cual es útil para enmarcar un titular, dar información, agregar texto o una leyenda dentro de la etiqueta <figure>.

Esta asociación entre la figura y una leyenda visible no era posible en el estándar anterior de HTML. La incorporación es relevante en términos de accesibilidad ya que agrega una manera de compartir información sobre la imagen o diagrama tanto con usuarios que visualizan la imagen, como con los que no.

3.2. Hojas de Estilo en Cascada

3.2.1. Introducción

Hoja de estilo en cascada (CSS) es un lenguaje creado para controlar el aspecto o presentación de los documentos electrónicos definidos con HTML y XHTML. Utilizar hojas de estilos es la mejor forma de separar los contenidos y su presentación, y es imprescindible si se quiere mejorar la accesibilidad, reducir el costo de mantenimiento y dar la posibilidad de visualizar el mismo documento en diferentes dispositivos.

La versión actual de CSS, conocida como "CSS3", continúa en desarrollo, pero se han publicado como recomendaciones oficiales muchas herramientas pertenecientes a esta versión, y muchas otras se encuentran en candidatas a recomendación²².

Los navegadores Web tienen por defecto hojas de estilo integradas que definen la presentación de todos los elementos [17]. El término cascada se utiliza porque existe una jerarquía para definir qué estilo es el que se le aplica a cada elemento.

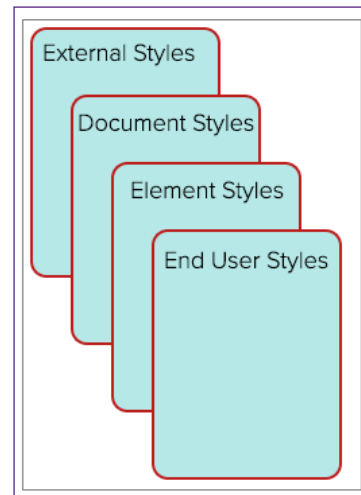


Imagen 5. Hojas de estilo.

Como regla general los estilos definidos en una hoja de estilos externa (External Style) "pisan" los definidos por el navegador. Los estilos definidos en el documento (Document styles) "pisan" los anteriores. Así si los estilos se definen en el elemento (Element style) estos son los que prevalecen. Y por último la decisión final cae en los estilos que ha definido el usuario final (End user Style).

Esta priorización es la más adecuada en términos de accesibilidad y usabilidad, ya que determinados usuarios pueden necesitar visualizar la página de una manera diferente a la que se le propone al iniciar una página. Por ejemplo, diferentes usuarios pueden requerir de un texto más grande, o de un mayor contraste de colores o de otro tipo de fuente.

3.2.2. CSS y accesibilidad

Tener el contenido en un archivo distinto al que define su presentación, es una técnica de desarrollo Web ampliamente aconsejada para mejorar la accesibilidad

²² Cuando el W3C tiene una propuesta como candidata a recomendación es muy probable que con pocos cambios pase a tener el estado de recomendada.

de la página. Separar el contenido (texto, imágenes y elementos del documento) de la presentación (forma de presentar el contenido) permite asegurar que si por algún motivo el archivo CSS está deshabilitado o se sobrescribe, el contenido seguirá siendo accesible. Además, si se define semánticamente la página, la visualización será agradable gracias a los estilos por defecto del navegador.

Mediante CSS, se puede modificar el orden visual de los elementos, más allá del orden del código. Los lectores de pantalla, se basan en el orden del código para leerles a los usuarios los elementos de la página; por lo tanto, si no se cargan los archivos CSS, los usuarios que visualizan la pantalla no podrán comprender la información que dependa del orden. CSS también es la herramienta con la cual se debe trabajar si se quiere ocultar determinado contenido. Más adelante se detallarán las opciones y recomendaciones del W3C para ocultar contenido correctamente.

Contrariamente a lo planteado en el párrafo anterior, en ocasiones CSS es utilizado para la presentación de contenido. A continuación se detallan las razones para hacerlo:

- Agregar una imagen para transmitir una idea
- Utilizar color para transmitir un mensaje.
- Modificar el flujo y cambiar el sentido original.
- Enfatizar contenido.

Generalmente estas acciones están ligadas a situaciones específicas, más adelante se detallan algunas.

3.2.2.1. Diseño Web Adaptativo

El diseño Web Adaptativo, es una técnica que persigue la correcta visualización de una página Web en distintas terminales. La misma fue incorporada en el año 2012 e incrementó su notoriedad trayendo otro enfoque a la resolución de problemas que

surgían a la hora de diseñar una Web para las distintas resoluciones de pantallas de los diferentes dispositivos.

Este tipo de diseños muestran el mismo código HTML a todos los dispositivos, más allá de sus dimensiones, usando solo CSS para cambiar la forma en que la página aparece. Google recomienda esta técnica de configuración [18] al tratar la optimización de sitios para móviles, por las siguientes razones:

- Si se usa una sola URL, a los usuarios les resulta más fácil interactuar con el contenido, compartirlo y enlazarlo. Así también, una sola URL, ayuda a los algoritmos de Google a asignar las propiedades de índice del contenido.
- No es necesario realizar un re-direccionamiento para que los usuarios lleguen a la vista optimizada, de modo que se reduce el tiempo de carga. El re-direccionamiento basado en el user-agent (una propiedad para identificar desde donde se está visitando el sitio derivándolos a distintos contenidos –misma URL-) es propenso a errores y puede degradar la experiencia de usuario
- Ahorra recursos tanto para el sitio como para los rastreadores de Google. En el caso de las páginas con diseño Web adaptativo cualquier user-agent del robot de Google rastrea la página una sola vez para recuperar el contenido, en vez de tener que rastrearlas varias veces con diferentes user-agents. Esta mejora en la eficiencia de rastreo puede ayudar de forma indirecta a Google a indexar una parte mayor del contenido del sitio y a mantenerlo adecuadamente actualizado

El diseño Web Adaptativo trae aparejado ciertas ventajas como ser:

- Reducción de costos: La creación de un sitio Web con un diseño adaptativo es más rápida y por ende más económica que la creación de un sitio destinado a computadoras de escritorio y otro destinado a teléfonos móviles.

- Actualizaciones: Una razón más que suficiente, no solo para emprender un desarrollo adaptativo en vez de varios desarrollos destinados a distintas plataformas, sino también para unificar desarrollos si es que están separados, es que en el tiempo se verán beneficios a la hora de las actualizaciones. Un cambio sencillo cuando se tienen varios desarrollos se puede tornar complejo, pero no así con el enfoque del diseño Web adaptativo.

3.3. WAI-ARIA

3.3.1. Introducción

Como se ha escrito anteriormente, el lenguaje HTML fue básicamente desarrollado para un modelo de cliente-servidor, en el que el usuario solicita al servidor y el servidor brinda el recurso. La evolución de la Web trajo entre otras cosas, las llamadas aplicaciones Web, las cuales están dotadas de funcionalidades brindadas por tecnologías como AJAX o javascript, que tornan a las páginas mucho más parecidas a los entornos de escritorio y con un funcionamiento más dinámico con respecto a los contenidos.

Las zonas que son actualizadas gracias a tecnologías estilo AJAX o javascript se denominan regiones activas. Estas regiones envían peticiones al servidor en un segundo plano, lo que hace que la página no se actualice por completo, pero que algún contenido específico se añada, elimine, o modifique.

Sin embargo, muchos productos de apoyo no reconocen estas regiones, es decir, no pueden reconocer que hay información nueva, y es precisamente por esto que surge la iniciativa del W3C, WAI-ARIA 1.0 o simplemente ARIA [19].

ARIA es una recomendación del W3C desde marzo del 2014 y la documentación fue escrita por el grupo de trabajo de protocolos y formatos para WAI. Esta tecnología tiene como principal objetivo aportar información acerca de las diferentes partes del contenido, especialmente del contenido dinámico, proveer de mayor sentido semántico y

dotar de mayor consistencia de soporte para acceso por teclado a los asistentes tecnológicos. Toda esta información será utilizada por los productos de apoyo para anunciar, acceder y saltar por los bloques relevantes de la página.

Las regiones que más suelen aparecer y más rápidamente pueden identificarse como "vivas", son aquellas que reflejen las últimas noticias o mensajes de un chat, mensajes de validación de un formulario, un banner que muestra un mensaje diferente cada un tiempo pre-establecido o un contador que indica el tiempo que resta para que empiece determinado evento. Cuando se produce la actualización de este tipo de contenido, el usuario puede tener el foco en cualquier otra parte de la página, de modo tal que si el usuario accedió con un producto de apoyo, la actualización le pasará desapercibida.

ARIA permite definir qué rol o función tiene un elemento e indicar su estado y sus propiedades, que a su vez se podrán modificar dinámicamente para que los productos de apoyo anuncien los cambios que se produzcan. ARIA permite también añadir información semántica, no solo sobre la estructura de la página, sino también sobre los elementos de la misma.

3.3.2. Roles

Es evidente que por definición, las etiquetas de HTML tienen un rol predefinido, el mismo que el propio elemento indica. Un rol de ARIA ayuda a concretar la funcionalidad de la etiqueta en la que va definida. Los roles no se deben modificar por acciones del usuario, sino que quedan fijos a lo largo del tiempo.

3.3.3. Estados y Propiedades

A diferencia de los roles, en el que sólo existe un atributo role y varios valores, los atributos de los estados y propiedades pueden ser varios, y cada uno puede aceptar varios valores. Los estados y propiedades, se definen tan sencillamente como los roles, es decir: se agrega el

atributo y el valor al elemento.

Tal como afirma la documentación brindada por la W3C [20] los términos “Estados” y “Propiedades” aportan características similares como ser que ambos proporcionan información específica acerca de un objeto. Si bien se mantienen conceptualmente distintos para marcar diferencias sutiles en su significado, la distinción entre estados y propiedades es de poca importancia para la mayoría de los desarrolladores y se suele definir a ambos términos simplemente como “atributos”. Como ejemplo de diferencia, aunque no es regla, se puede nombrar que los valores de las propiedades probablemente cambien menos a lo largo del tiempo en comparación a los valores de los estados que puede cambiar con frecuencia debido a la interacción del usuario.

Los elementos dinámicos cambian de estado, y el cambio se refleja teóricamente en el valor del atributo. El cambio se hace mediante javascript para que los productos de apoyo puedan transmitir el cambio al usuario.

3.3.4. Acceso Mediante Teclado

En HTML, sólo los enlaces y los elementos de entrada pueden recibir el foco del teclado. Esto significa que a medida que ‘tabulamos’ a través de una página, el navegador se detiene en determinados elementos. Los elementos de bloque como <div> no pueden captar el foco, por lo que una funcionalidad establecida ahí no podrá ser accedida mediante teclado, mientras que los elementos que reciben el foco pueden ser controlados a partir del atributo tabindex.

ARIA extiende las propiedades de tabindex [22] y permite implementar el atributo a cualquier elemento. Dependiendo del valor que tome, cambiará su actuación:

tabindex=“0”

Permite que un elemento que no toma el foco, por defecto pueda recibir el foco y se pueda acceder a el elemento con el tabulador. El orden dependerá de su

posición en el documento.

tabindex=“-1”:

Se diferencia del caso anterior es que no es posible tabular hasta el elemento, pero el valor “-1” en el atributo tabindex permite tomar el foco por programación.

tabindex=“X” (X siendo un numero positivo del 1 al 32768)

Especifica directamente dónde se coloca el elemento en el orden de tabulación. Estos elementos se colocan en el orden de tabulación antes de cualquier elemento con tabindex = “0” o que están naturalmente incluidos en el orden de tabulación (elementos de formulario y enlaces).

4. Implementación del Caso de Estudio

El presente capítulo detalla las posibilidades y alternativas, así como la implementación, de una página Web que cumple con los criterios de accesibilidad al contenido Web 2.0. Para cada sección se puntualizan los aspectos más relevantes y particulares de ese requerimiento. Los requerimientos son fundamentales como punto de partida, pero no dejan de ser un pretexto para lograr una implementación accesible de exigencias que generalmente se presentan en la mayoría de las páginas Web.

4.1. Relevamiento

En este apartado se describen brevemente los requisitos que fueron extraídos y refinados luego de varias reuniones con la Secretaria de Turismo de la Ciudad de Santa Fe y que se utilizaron para completar un caso de estudio concreto.

4.1.1. Visión

Relacionar a los prestadores de servicios turísticos con los visitantes de la ciudad de una forma rápida, intuitiva y accesible. Concretamente se persigue mejorar la promoción de los oferentes turísticos empleando los nuevos medios

y tecnologías disponibles, y propender al mayor consumo económico y cultural de los visitantes de la ciudad.

4.1.2. Actores

- Usuario (Casual): Usuarios sin registro que visita el sitio.
- Usuario Registrado (UR): Usuario registrado que visita el sitio.
- Usuario de Carga (UC): Usuario con accesos para carga de prestadores de servicios.
- Administrador (A): Usuario con total acceso.

4.1.3. Lista de Producto

1. Mapa con prestadores de servicios. Mostrar al usuario un mapa con los distintos prestadores de la ciudad de Santa Fe, el mismo, podrá tener la posibilidad de filtrar los prestadores según la categoría.

Los UC serán capaces de dar de alta un prestador de servicio, con los siguientes datos:

- Descripción.
- Categoría (hoteles, restaurantes, cajeros automáticos, café, otros).
- Localización en mapa de Google.
- Enlace al sitio oficial.

2. Comentarios. Darle la posibilidad a los UR de dejar un comentario general acerca de la ciudad de Santa Fe y su experiencia en la ciudad. El mensaje debe poder ser aprobado por un Administrador (A) antes de que sea visible para los demás visitantes.

3. Opciones de Temporada. Contar con una sección dinámica, en la que siempre estén presentes las actividades más apropiadas para esa temporada. No se deben borrar las que se cargan, sino que tiene que existir la posibilidad de ir agregando y seleccionando las deseadas. La información es acotada, pero no más de 5 párrafos.

4. Página de Contacto. Permitirle al

usuario hacer contacto rápidamente con turismo de la ciudad de Santa Fe.

5. Galería de imágenes. Contar con una galería de imágenes gestionable por un A.

6. Sitio Multilenguaje. Brindar la posibilidad de que todos los usuarios sean capaces de visualizar los comentarios en cualquier idioma (de igual forma pueden dejar el comentario en cualquier idioma), pero el sitio debe estar en el idioma definido por cada usuario. Sin embargo, los UC deben visualizar el sitio en español.

7. Video Institucional. Contar con un video institucional en el home del sitio.

8. Redes Sociales. Brindar la posibilidad de compartir y/o dar "likes" a comentarios y hacer login mediante redes sociales.

9. Compra de souvenir. Facilitar al usuario la posibilidad de comprar algún souvenir de Santa Fe.

10. Música de Santa Fe. Brindar la posibilidad de escuchar música autóctona.

11. Carga de Fotos. Brindar la posibilidad al usuario de cargar sus propias fotos.

4.2. Desarrollo Del Caso de Estudio

Más allá de cualquier finalidad, cada página Web cuenta con secciones generales, comunes a la mayoría de las páginas y secciones específicas relacionadas con la temática. El presente capítulo, además de la implementación de las secciones generales y específicas (de acuerdo a lo descrito en el relevamiento), detalla las posibilidades y alternativas de la interface Web, para lograr un nivel A de accesibilidad de acuerdo a las WCAG 2.0. No todos los criterios de conformidad se podrán aplicar al caso de estudio. Algunos de los criterios ya nombrados son tenidos en cuenta en proyectos determinados (por ejemplo: páginas de subastas) que no se dan en el caso de estudio de este proyecto.

4.2.1. Barra de Navegación

Como se vio anteriormente, HTML5 define una nueva etiqueta <nav>, para identificar

semánticamente la sección o secciones de navegación del sitio [8]. Un uso práctico específico de identificar la barra de navegación se da en el caso de dispositivos móviles, donde los agentes de usuario intentan constantemente ahorrar espacio. Llegado el caso que se entienda que el menú de navegación no es necesario que permanezca visible (por lo menos hasta que el usuario lo pida nuevamente), el mismo se puede ocultar y dejar ese espacio para agrandar o mejorar la distribución del contenido. Para esto, el navegador tiene que saber delimitar qué parte es la que le corresponde al menú de navegación.

Como vimos con ARIA se pueden especificar los roles de las diferentes zonas funcionales de un documento Web, haciendo que sea más semántico y accesible. Para contar con una estructura más robusta, se puede añadir el atributo role con el valor navigation a <nav> [32], quedando:

```
<nav role="navigation">
```

La estructura de la lista en el interior de <nav> puede tener muchas variantes, cada una con sus particularidades. No hay una gran diferencia estética entre cada uno de los casos, sino que la diferencia está en el código.

Alternativa Enlace

Una forma de asegurar la accesibilidad en una barra de navegación es recurrir al más antiguo de los recursos Web como es el enlace. Si se establece el menú principal como enlaces a páginas nuevas que contengan el submenú correspondiente en otras páginas y así sucesivamente, se puede asegurar su correcto funcionamiento.

A continuación se muestran dos imágenes que ejemplifican el procedimiento de este tipo de codificación.

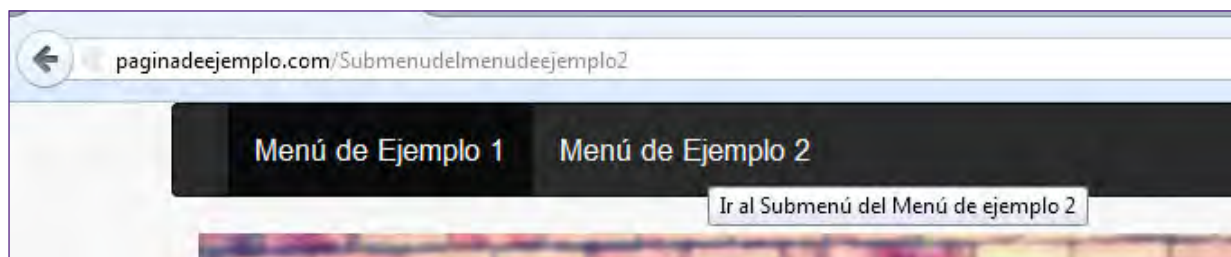


Imagen 6. Menú básico de ejemplo.

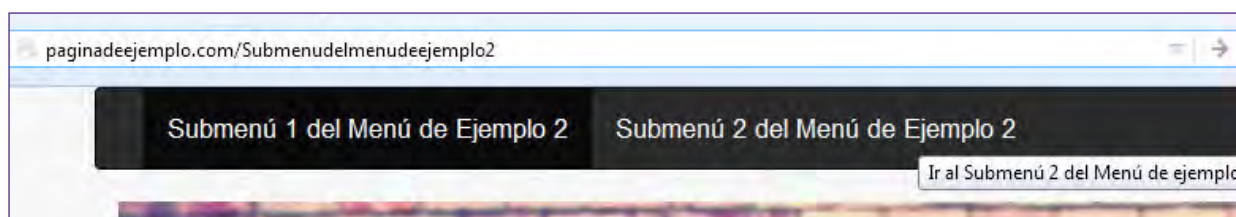


Imagen 7. Submenú básico de ejemplo.

Esta opción es poco vista dado que:

- El usuario estaría esperando la carga de una nueva página para cada submenú.
- El mantenimiento es complejo.
- Se utiliza gran cantidad de recursos del servidor.

A pesar de los puntos nombrados

anteriormente, brinda sin dudas la opción que tendrá mayor grado de posibilidad de ser ejecutada correctamente por los navegadores (al ser el enlace el recurso básico de internet).

Alternativa ARIA propuesta por la W3C

El documento de la W3C sobre cómo usar ARIA [33] aconseja hacer el menú

con los atributos y valores, `role="menu"`, `role="menuitem"`, `role="menubar"` y `ul` y `li` como elementos finales.

Esta opción es una buena alternativa y fundamentalmente es accesible según las WCAG 2.0, pero los elementos del menú no están propuestos como enlaces. Una de las funcionalidades más utilizadas por los usuarios de lectores de pantalla es la de listar los enlaces de la página que visitan. Si el menú es codificado de esta forma, cuando el usuario liste los enlaces de la página no podrá visualizar las opciones que brinda el menú. Además, de codificarse la estructura del menú de esta manera, no se podría navegar

directamente mediante el tabulador²³ (aunque se pueda reemplazar también este comportamiento con `tabindex`).

Alternativa Combinada

Establecer una estructura de tablas con enlaces dentro, puede solucionar la desventaja que presentó la alternativa ARIA expuesta en el punto anterior. Esta estructura, permite que el menú figure en la lista de enlaces y permite al usuario recorrerlo mediante el tabulador. En la siguiente imagen, puede verse como se listan los enlaces de la página (incluyendo el menú) para el caso de estudio planteado para el proyecto.

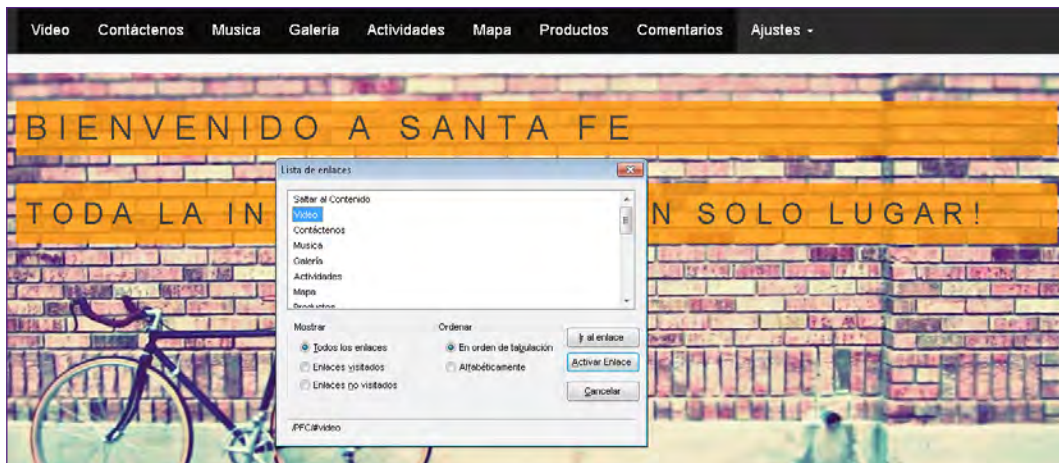


Imagen 8. Ventana con listas de enlaces.

4.2.1.1. Adaptabilidad

Como vimos anteriormente uno de los cuatro principios de las WCAG 2.0 es la robustez. La adaptabilidad es un aspecto fundamental de la robustez y es parte de las características que debe tener una página accesible. Mediante un diseño adaptable, es posible modificar el diseño del menú para que quepa en la pantalla de destino. El método más común y adoptado universalmente es el de cambiar el menú que se encuentra de forma horizontal (generalmente en el límite superior de la pantalla) por tres rayas en alguna esquina (generalmente en la superior derecha) que contenga el menú colapsado.

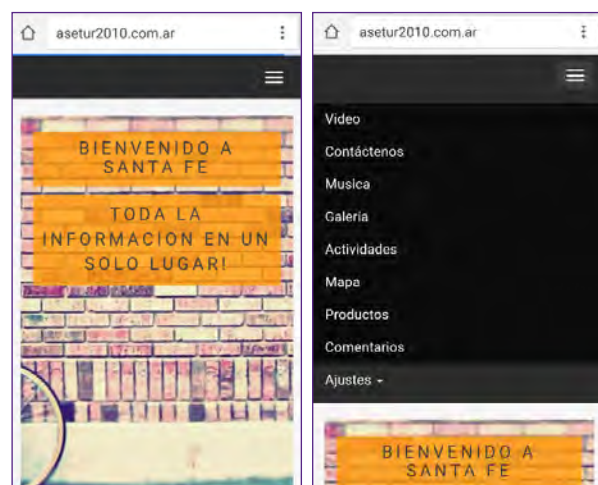


Imagen 9. Barra de Navegación Colapsada en un Smartphone. Imagen 10. Barra de Navegación Descolapsada en un Smartphone

²³ Los usuarios que utilizan lectores de pantalla suelen avanzar en la página web mediante esta tecla (\leftarrow).

Lógicamente que existen infinidad de opciones, pero la nombrada anteriormente es la aplicada frecuentemente. Al abrir el menú, el usuario se encuentra con un menú vertical. Entonces, se pasa de tener un menú lineal y visible, a un menú desplegable, por lo que es importante asegurarse de que la navegación siga siendo accesible para todos los usuarios. La opción seleccionada para este Proyecto fue la de "alternativas combinadas" como ya hemos comentado.

4.2.1.2. Saltear Menú

Podemos hacer cumplir el criterio de conformidad 2.4.1 al incluir después de la etiqueta <body> un enlace que permita saltarse el menú. Si el menú tiene varios elementos que se repiten constantemente, un usuario que acceda por teclado quizás quiera saltar al contenido principal y así acelerar su navegación yendo directamente al contenido.

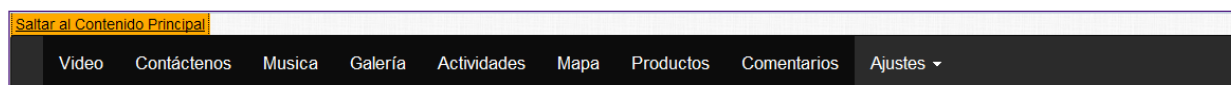


Imagen 11. Menú Descolapsado (Para dispositivos con resolución mayor a 780px de ancho).

Un caso sencillo útil como ejemplo para el entendimiento de esta funcionalidad es la página web de un diario, supongamos que el menú principal contiene los elementos: "Noticias más leídas", "último momento", "Deportes", "Espectáculo", "Ocio", "Viajes", "Servicios" y "Clasificados". Cuando el usuario selecciona deportes es de esperar que sea dirigido a otra página donde también esté el menú presente con las opciones mencionadas anteriormente. El usuario ya ha seleccionado que quiere leer las noticias relacionadas con el deporte pero el lector de pantalla lee secuencialmente la página y leerá el menú en primera instancia "Noticias más leídas", "último momento", "Deportes", "Espectáculo", "Ocio", "Viajes", "Servicios" y "Clasificados" en vez de las noticias de deporte. Si el usuario tiene la opción de "saltar al contenido principal" podrá hacer que el lector de pantalla no lea el bloque perteneciente al menú y la navegación será más dinámica.

Según una encuesta realizada por WebAim [34], aproximadamente el 40% de los usuarios de lectores de pantalla utilizan frecuentemente esta utilidad cuando está disponible.

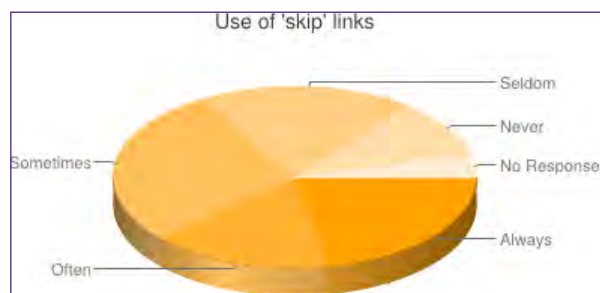


Ilustración 12 - Uso del enlace "saltar navegación"[34].

La representación tabular de la imagen anterior se presenta a continuación:

Respuesta	Porcentaje
Cuando está disponible	22%
Seguido	16%
A veces	28%
Raramente	19%
Nunca	10%
No responde	4%

Tabla 1. Uso el enlace "saltar al contenido" o "saltar navegación".

El framework bootstrap presenta clases muy específicas, sr-only y sr-only-focusable (sr por screen reader) las cuales solo muestran el contenido si se accede con lectores de pantalla o se tiene el foco puesto en el elemento. Esto ahorra mucho tiempo de codificación CSS y pruebas.

Al principio solo se implementó sr-only, pero la realidad es que agregar la clase sr-only-focusable al bootstrap 3.2.0 es una ventaja para usuarios que acceden mediante lectores de pantalla y también para usuarios que utilizan el teclado para navegar [12].

Más adelante se describirán varias técnicas que oculten el contenido a usuarios que pueden visualizar la pantalla y a la vez, que sean accesibles para usuarios que ingresan con tecnologías de apoyo.

Derivado de lo expuesto con anterioridad, para el caso de estudio, con el objetivo de cumplir con un nivel A de las WCAG 2.0, se presenta una codificación como la planteada en “Alternativas Combinadas” con un aspecto adaptativo.



Imagen 13. Menú colapsado (Para dispositivos con resolución menor a 780px de ancho).

Además se antepone el enlace necesario para que el usuario cuente con la opción de saltar el menú y que no sea leído por los lectores de pantalla página tras página.

4.2.1.3. Casos externos que presentan aspectos de inaccesibilidad en la barra de navegación

Como vimos, es posible que en un elemento como la barra de navegación no se cumpla con varios de los criterios de conformidad expuestos en las WCAG 2.0. Estos incumplimientos pueden estar dados en aspectos generales (como el uso del color, el tamaño de la fuente o la falta de un enlace para evitar la lectura completa de la barra de navegación) comúnmente encontrados en desarrollos en los que la accesibilidad no se tiene en cuenta, pero, los errores o equivocaciones²⁴ más usualmente cometidos en desarrollos que si contemplan aspectos de accesibilidad están dados en la codificación del elemento y en el intento por que este sea adaptable.

²⁴ Fallo de sistema en un requerimiento

Si navegamos la página de Starbucks²⁵ podemos identificar rápidamente claros indicios de que la accesibilidad es un aspecto que se ha tenido en cuenta para el desarrollo (han utilizado ARIA por ejemplo). Pero si bien la navegación por la barra es óptima cuando el menú esta descolapsado, al colapsarlo no es posible hacer foco para desplegarlo, ninguna persona que ingrese mediante el teclado podrá ingresar al menú. Esto se debe a que el elemento que contiene los enlaces del menú es un <div> sin el atributo tabindex. En nuestro caso de estudio al tratarse de un botón no se tiene este problema.

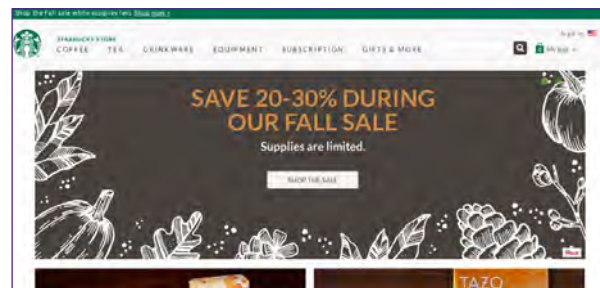


Imagen 14. Página de starbucks con menú descolapsado

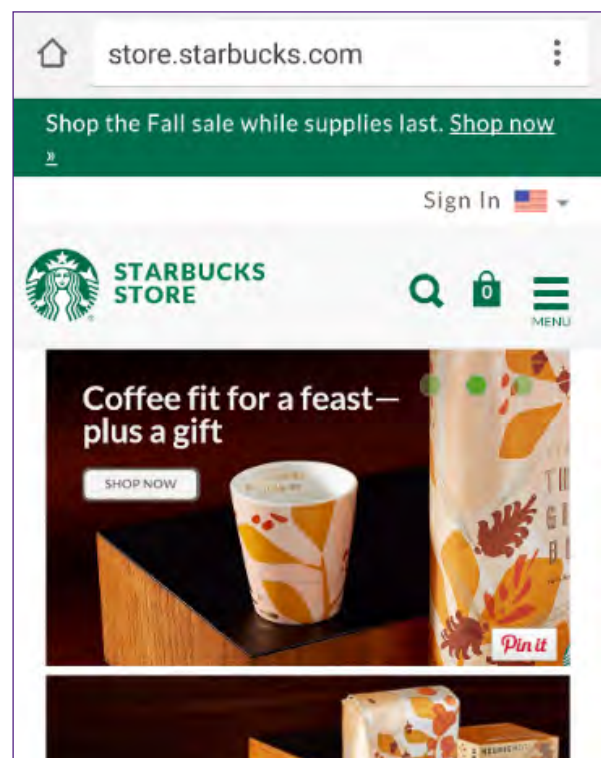


Imagen 15. Página de starbucks con menú colapsado

²⁵ <http://store.starbucks.com/>. Cadena internacional de café fundada en Seattle, Washington.

La barra de navegación presentada en el sitio de avianca²⁶ exhibe un error más sencillo pero que refleja el incumplimiento de las WCAG 2.0. Simplemente no propone un enlace para saltar al contenido principal (criterio de conformidad 2.4.1: Evitar Bloques). De esta forma un usuario que utiliza el teclado para navegar si requiere información relacionada con las cancelaciones o reembolsos, primero el lector de pantalla deberá pasar por 40 enlaces y luego por 80 (el total) para terminar yendo al contenido principal con la información que el usuario está necesitando.

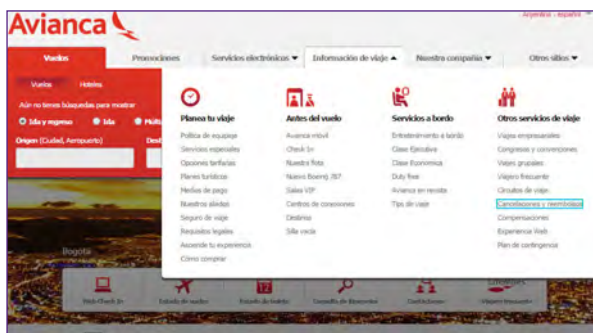


Imagen 16. Página de Avianca.

4.2.2. Cabecera

La cabecera en las páginas Web es la primera impresión que tiene el usuario del sitio en el que se encuentra, por lo que se debería poder visualizarla completamente sin la necesidad de ninguna acción. En las páginas tradicionales la cabecera cuenta con un logo, algunos títulos y no ocupa más del 20% del largo. La tendencia actual es mostrar grandes imágenes fijas o cambiando en determinado tiempo (slider) y una frase representativa de la filosofía de la empresa.

Dentro de los nuevos elementos que incorporó HTML5 esta <header>. Este elemento que envuelve la cabecera de cada documento o sección, es utilizado como contenedor de la introducción al contenido como ya se ha descrito.

También se ha comentado que en las especificaciones de HTML5 se permite que la página cuente con varios títulos de

etiqueta h1, uno en cada sección, e iniciar nuevamente la cadena comentada de h2,...,h6. En este caso de estudio se optó por dejar un único h1 en la cabecera de la página, comenzar las secciones con h2 e ir subiendo hasta h6, para no asumir un riesgo innecesario.

Es posible complementar la semántica con ARIA agregando el rol banner, tal como definen los documentos de recomendación [13]. Se completa así:

```
<header role="banner">
```

4.2.2.1. Imagen Decorativa

Una de las condiciones más conocidas de la accesibilidad es la de completar el atributo alt de la etiqueta , la cual es una situación claramente sencilla de resolver técnicamente, pero requiere conocimiento del tema.

En este caso en particular, la imagen es solo decorativa, no contiene información y no le sería útil al usuario que se le comente de qué imagen se trata. Es por esta razón que se ha agregado desde la hoja de estilos, siguiendo la recomendación de la W3C respecto a qué hacer cuando no hace falta un contenido textual alternativo.

En el capítulo dedicado a la configuración correcta de una galería de imágenes, se describe más en detalle cómo seleccionar un correcto texto alternativo para las imágenes.

Para el caso de estudio se ha decidido codificar la cabecera envuelta en la etiqueta:

```
<header role="banner">
```

Conteniendo una imagen adaptativa junto con el único título <h1> de la página.

²⁶ Principal aerolínea de Colombia.



Imagen 17. Cabecera.

4.2.2.2. Caso externo que presenta aspectos de inaccesibilidad en la cabecera

Podemos ver en el sitio oficial de Gobierno de la Provincia de Santa Fe²⁷ el incumplimiento del criterio de conformidad 1.3.1: Información y Relaciones, como hemos nombrado anteriormente una de las técnicas suficientes es "Usar h1-h6 para identificar

los títulos". Este criterio no se cumple ya que no mantiene una estructura de encabezados coherente, la página no tiene en su estructura un elemento H1 pero si varios elementos H2. El incumplimiento del criterio puede afectar la "visualización global" de una página en braille por ejemplo [50].

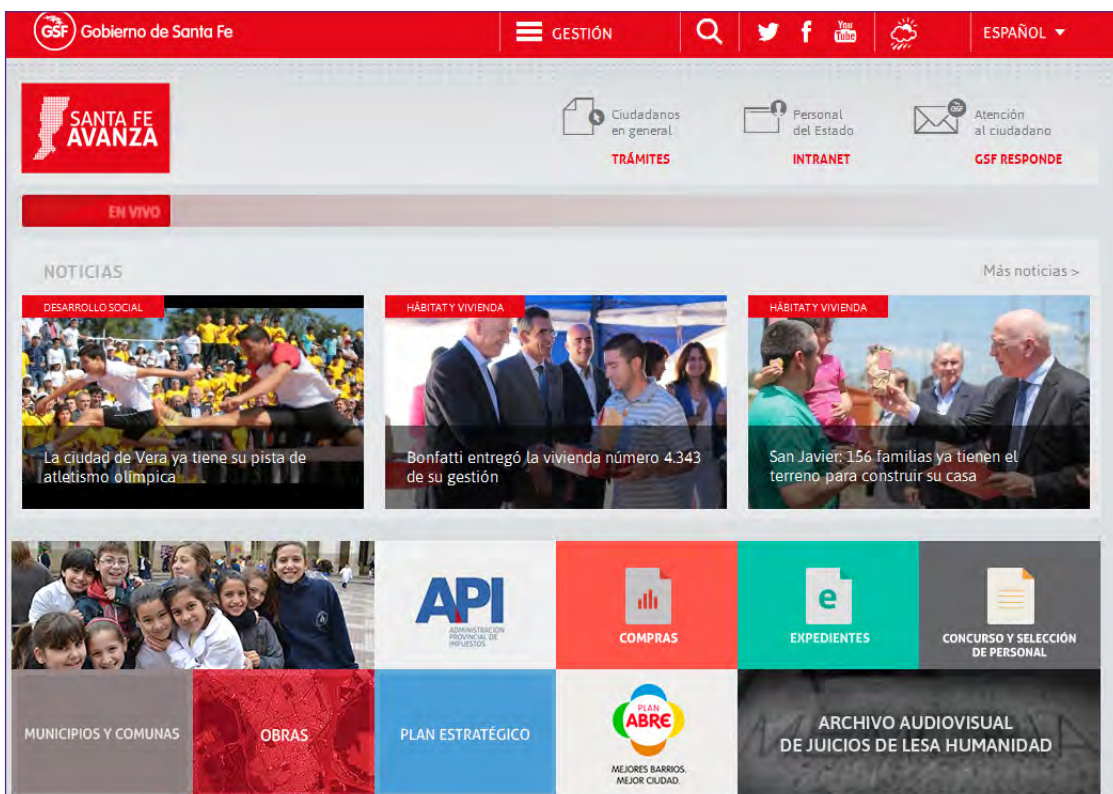


Imagen 18. Página del Gobierno de la Provincia de Santa Fe.

²⁷ <http://store.starbucks.com/>.

4.2.3. Contenido Principal

Mediante la etiqueta <main> se envuelven las opciones que ofrece la Ciudad de Santa Fe. En la sección dedicada a la descripción teórica de las etiquetas incorporadas a HTML se identificará el motivo.

Para dotar de mayor robustez en lo que respecta a la estructura, ARIA define también un bloque dedicado al contenido principal del documento, para así poder definir el rol main como valor del atributo role de la etiqueta <main> [14], quedando codificado:

```
<main role="main">
```

Hemos comentado la importancia que puede tener este elemento para evitar al usuario pasar por bloques de contenido que ya ha "leído". Si se identifica la sección principal de la página el mismo lector de pantalla puede ofrecer ir al contenido principal (ya que sabrá donde está) sin que el desarrollador hubiera incorporado un enlace con este propósito.

4.2.4. Secciones

De las etiquetas nuevas de HTML5 que se definen para la semántica, existe la ya descrita <section>. En el desarrollo,

cada sección de la página se divide mediante este elemento, con su título e ID correspondiente.

Como se ha visto en otras oportunidades, es posible sumar accesibilidad mediante las especificaciones de ARIA, ya que en este caso HTML5 permite complementar la etiqueta role con el valor "región" [15].

4.2.5. Footer

La etiqueta footer ya ha sido presentada previamente y representa el pie de un documento o sección. La W3C propone la utilización del rol ARIA contentinfo en el atributo role, que se suele establecer para una región perceptible y grande, que contiene información sobre el documento principal. Dentro de cada role definido como documento o aplicación, se debería marcar no más de un elemento con el valor de role contentinfo.

Para el caso estudio, tal como se ha especificado en el párrafo anterior, el pie de la página se envuelve en la etiqueta:

```
<footer role="contentinfo">
```

Conteniendo en su interior enlaces relacionados y el alcance de accesibilidad alcanzado en la página.



Imagen 19. Pie de página del caso de estudio (Footer)

4.2.2.1. Imagen Decorativa

Así como en casi cualquier sección de una página web los errores posibles de cometer son infinitos. Los más típicos suelen estar relacionados con el segundo principio: Operable. Se puede ver este principio incumplido en el sitio web del instituto provincial para la inclusión de las personas con discapacidad del chaco (IPRODICH)²⁸, específicamente el criterio de conformidad 2.4.3: Orden del foco.

Aunque aparentemente nada parece sugerir algún tipo de inconveniente, al recorrer el sitio mediante el teclado se puede ver una incoherencia en el orden del foco. Los elementos del pie de página reciben el foco de la siguiente manera: "Informes", "Asesoría Legal", "Proyectos Especiales", "Prestadores", "205", "212", "214", "207" y así con las siguientes columnas. Una de las técnicas suficientes para cumplir el criterio es: Situar el foco en los elementos en un orden secuencial. Claramente para el caso de un listado

²⁸ <http://iprodoch.gob.ar/index.php?r=norma/index>.

de teléfonos el orden del foco debería ser: "Informes - 205". "Asesoría Legal

- 212", "Proyectos Especiales - 214" y "Prestadores - 207".

0362 - 4453101 INTERNOS POR ÁREAS				ENLACES DE INTERÉS	
Informes	205	Secretaría Privada	203	Planificación	211
Asesoría Legal	212	Secretaría General	218	Administración	216
Proyectos Especiales	214	Mesa de Entradas	206	Compras	209
Juntas de Personas y Prestadores	207	Personal	210	Rendición y Patrimonio	215

ENLACES DE INTERÉS	
-	CONADIS
-	Servicio Nacional de Rehabilitación - SNR
-	Instituto de Vivienda - IPDUV
-	ANSES

IPRODICH | PRESTACIONES Y PRESTADORES | PROGRAMAS DE APOYO | INSTITUCIONES

CHACO
GESTION CAPITANICH

Imagen 20. Pie de página del sitio web de IPRODICH.

4.2.6. Video

Las especificaciones de ARIA no definen ningún complemento para esta etiqueta, por lo que solo <video> es suficiente semánticamente para describir su contenido.

Navegación por Teclado

Uno de los problemas actuales con los complementos agregados para visualizar videos, es el pobre acceso por teclado que tiene el usuario para interactuar con el video. Aún peor es el hecho de que incorporar cualquier funcionalidad de teclado al video puede llegar a ser muy laborioso y hasta imposible. Vale recordar que para el cumplimiento de criterio de conformidad 2.1.1 (Accesible por teclado) es necesario que el usuario pueda interactuar con el video mediante el teclado. En este aspecto, la inserción de video mediante la etiqueta <video> de HTML5 es un gran avance ya que permite una correcta navegación mediante el teclado.

Como los controles (play, pausar, stop y volumen) están definidos por el propio navegador esto está sujeto a cada implementación, aunque se puede personalizar mediante javascript.

Subtítulos

HTML5 proporciona una herramienta mediante una etiqueta denominada track dentro de <video> que permite la incorporación de subtítulos o captions²⁹ a los videos a una página, sin la necesidad

de plugins. Hay que tener en cuenta que en borradores del lenguaje HTML5 anteriores al 2010, ésta opción no estaba disponible, y aun existe mucha bibliografía afirmando que no es posible introducir subtítulos de forma nativa [16]. La incorporación de los captions es necesaria para el cumplimiento de las WCAG 2.0 ya que el criterio de conformidad 1.2.2 así lo establece.

Audiodescripción

Proporcionar una audiodescripción para el video es requerido por los estándares de accesibilidad WCAG 2.0 (criterio de conformidad 1.2.5) para cumplir con un nivel AA y es una de las recomendaciones para mejorar la usabilidad. Pero proveerla puede ser algo complejo por las siguientes razones:

- Es caro de producir.
- No es una tarea sencilla, ya que las descripciones deben estar en medio de los espacios vacíos de audio y la audiodescripción va entre medio del audio original.

La tarea de incorporar audiodescripción a un video, no tiene comparación con la opción del subtítulo que es nativa, fácil de implementar y es más reconocible por

²⁹ Subtítulos generales (subtitles): acompañan los diálogos de los personajes, voces off y elementos que se traducen.

Subtítulos para sordos (captions): subtítulos generales que incluyen información sobre los sonidos que suceden ("murmullos de fondo") o la anotación "gritando".

el usuario. La alternativa de los subtítulos cumple con los criterios para completar el nivel A de accesibilidad, y no requiere de javascript para su implementación.

Para el caso de estudio se ha incorporado el video según el estándar HTML5, por las razones expuestas anteriormente, conteniendo este, todo lo necesario para cumplir los criterios de accesibilidad correspondientes:



Imagen 21. Interface de Video del caso de estudio.

4.2.6.1. Casos externos que presentan aspectos de inaccesibilidad en videos

Hay una gran cantidad de sitios Web que presentan inaccesibilidad en los

elementos de video que incorporan en sus sitios. Fundamentalmente el criterio de conformidad por el cual se puede concluir rápidamente que no se cumplen con las WCAG 2.0 es el criterio de conformidad 1.2.2: Captions, pero como en todos los componentes hay varios motivos por los cuales se puede tener un video inaccesible según las WCAG 2.0 y perjudicar directamente a personas sordas pero también a los usuarios en general.

Podemos ver como en el Banco Patagonia por ejemplo se brinda información sensible como la forma en la que un cliente del banco debe generar su clave mediante un video completamente inaccesible.

El video se abre en una ventana emergente sin previo aviso, es necesaria la instalación de un plugin para funcionar, no es posible manejar los comandos (poner en pausa, subir volumen, etc.) mediante el teclado y no se agregaron los captions correspondientes.

Claramente las personas sordas no tendrán acceso a toda la información.



Imagen 22. Página web del Banco Patagonia.

Para sumar otro ejemplo podemos ver el sitio web de la casa rosada³⁰. El sitio incorpora los videos de una forma más razonable que el ejemplo anterior. Los videos están codificados mediante HTML5 por lo cual sin mucho esfuerzo como ya

hemos comentado se podría agregar captios de forma sencilla. Pero este es uno de los puntos por los cuales este sitio no cumple con las WCAG 2.0.

³⁰ <http://www.casarosada.gob.ar/>.



Imagen 23. Sitio web de la Casa de Gobierno.

4.2.7. Música

Si bien se ha descrito anteriormente en términos generales qué es lo que se gana con la llegada de las nuevas etiquetas de multimedia, esta porción del Proyecto especificará cómo y por qué el audio que se ofrece cumple con los criterios de accesibilidad del nivel A. Al igual que la etiqueta <video>, la etiqueta <audio> es suficiente semánticamente para describir su contenido y ARIA no especifica ninguna recomendación para aportar a su interpretación.

Se puede alcanzar la accesibilidad planteada por WCAG 2.0 con Nivel A (también nivel AA y AAA de así diseñarlo) exponiendo una transcripción textual del contenido auditivo (criterio de conformidad 1.2.1), ya que se trata de solo audio y grabado. Al ser canciones el contenido específico planteado en este caso de estudio, es necesario (no para el nivel A en realidad), agregar el atributo lang, ya que el texto se va a mantener en español más allá del idioma del documento. La letra de las canciones se coloca dentro de un elemento desarrollado por bootstrap.

El framework ofrece una manera simple de ocultar el contenido, colocando un elemento llamado collapse, el cual ocultará el contenido hasta que el usuario presione el enlace (el título). Esta opción

es atractiva ya que la letra de la canción puede no ser interesante para todos los usuarios, por lo tanto se mostrará a los que manifiesten interés. Si el usuario se decide a abrir el contenido, el contenido es mostrado mediante un efecto de deslizamiento y el mismo incorpora algunos roles y estados de ARIA para mayor accesibilidad.

Si bien puede ser mejorado agregando más opciones de teclado para acceder a la información, o sea para ver el contenido o para leerlo es completamente funcional, accesible y usable.

La misma funcionalidad de ocultar contenido hasta no ser solicitado por parte del usuario explícitamente la ofrece HTML5 en su nuevo estándar con los elementos <details> y <summary>, con las mismas ventajas comentadas anteriormente respecto a la no dependencia de javascript y a la homogeneidad. Contrariamente a otros casos en los que como conclusión se consideró HTML5 como el mejor camino a implementar, en este caso resultó ser que las etiquetas <details> y <summary> no están del todo aceptadas aún.

Para el caso de estudio desarrollado, se ha incorporado la música envuelta en la etiqueta audio anteriormente descrita, con la siguiente interface:

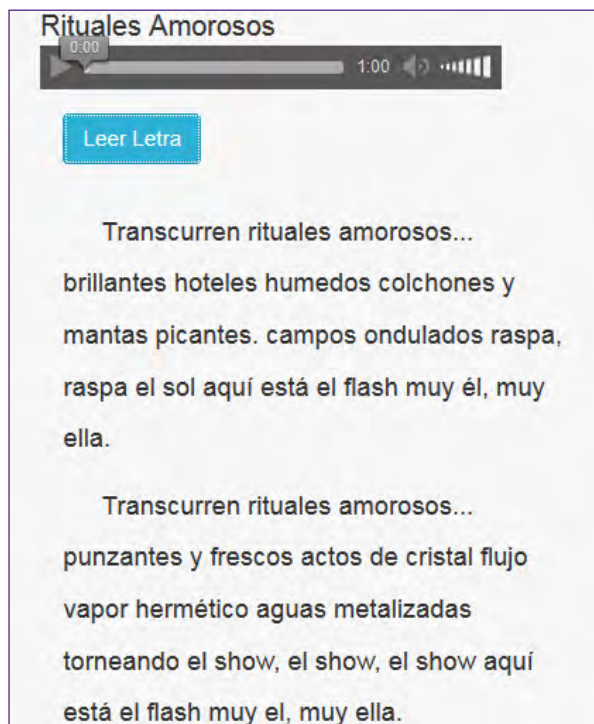


Imagen 24. Elemento de Música agregado en el Caso de Estudio.

La imagen anterior representa la forma más adecuada respecto a criterios de accesibilidad de incorporar un archivo de audio en una página Web.

4.2.7.1. Casos externos que presentan aspectos de inaccesibilidad en audios

Al igual que para la incorporación de video, el audio suele presentar aspectos inaccesibles en una gran cantidad de sitios. Los incumplimientos de las WCAG 2.0 a la hora de sumar este recurso en un sitio generalmente están relacionados con la posibilidad de manejar su reproducción mediante el teclado o con no proporcionar un documento alternativo que presente información equivalente a su contenido. Directamente afectando a las personas sordas.

Podemos identificar que en el diario Veloz³¹ que los audios son incorporados mediante un plugin (No incorporado por defecto por Chrome por ejemplo) que no permite ser controlado con el teclado. Además de eso no agrega la transcripción

³¹ <http://www.diarioveloz.com/notas/145660-audio-las-reveladoras-escuchas-julio-grondona-arbitrajes-dedo-y-evasion-impuestos>.

correspondiente para cumplir con las WCAG 2.0, específicamente el criterio de conformidad 1.2.1: Sonido solo o video solo.



Imagen 25. Captura de pantalla de la página del diario Veloz.

Otro diario como Infobae³² si bien agrega el audio tal como es aconsejado por este Proyecto no proporciona un documento alternativo que presente información equivalente a su contenido incumpliendo el criterio de conformidad 1.2.1 tal cual el ejemplo anterior.



Imagen 26. Captura de pantalla de la página de Infobae.

4.2.8. Galería de Imágenes

Las galerías de imágenes están presentes en todas partes se pueden ver en redes sociales, tiendas online o en páginas personales. Utilizar una galería es una forma difundida de incorporar una gran cantidad de imágenes en relativamente poco espacio, complementando los contenidos y aportando material visual

³² <http://www.infobae.com/2014/11/26/1671651-escuchas-causa-nisman-b-1009-2014-11-26-210738-2>.

para generar más interés.

Para este caso de estudio en particular en el que se apunta a un desarrollo utilizando las últimas tecnologías posibles, se codificó según la semántica propuesta por el estándar HTML5 y recomendada en el documento titulado: "Técnicas para proveer un texto alternativo útil" [35].

Así también, siguiendo con la norma del Proyecto hasta ahora, el estudio se vale de ARIA para dotar de mayor robustez y accesibilidad al código anterior, y para esto se agregó un rol que une todo lo envuelto por el elemento figure. Siendo el role group (role="group"), el elemento quedaría de la siguiente forma:

```
<figure role="group">
<div>

</div>
<figcaption>
Leyenda Correspondiente
</figcaption>
</figure>
```

Una de las características menos conocidas de wordpress es la capacidad de añadir galerías de fotos en casi cualquier lugar, de forma nativa y sin necesidad de instalar ningún componente adicional. El área de administración es muy completa e intuitiva, posibilita dotar a las imágenes de textos alternativos, y ofrece los siguientes campos:

- Título: Título de la imagen (no aparece en el código impreso por defecto).
- Leyenda: Utilizada en HTML5 para completar la etiqueta figcaption.
- Texto alternativo: Completa el atributo de la imagen alt.
- Descripción: Descripción de la imagen (no aparece en el código impreso por defecto).

En primera instancia, la galería que propone wordpress es relativamente accesible, es decir, le da la oportunidad al administrador de que las imágenes sean completadas correctamente de forma que se pueda comprender cuál es el objetivo de la imagen aunque no se las pueda ver.

Los dos inconveniente principales de la galería que wordpress brinda por defecto son que no se adapta a distintas resoluciones y que codifica estilos en línea (completamente desaconsejado [49]). Esto podría ser un problema menor si se podría revertir la situación mediante algunos cambios en los estilos pero no es posible de manera sencilla.

Por último, dado que la incorporación de galerías de forma nativa es una característica reciente de wordpress, pocos temas tienen definidos estilos para su correcta visualización. El tema utilizado en este Proyecto (DevDmBootstrap3) tampoco trae por defecto ninguna referencia a la galería, por lo que el cambio de estructura permite fácilmente agregar en la plantilla de estilos lo necesario para tener un diseño adaptable.

4.2.8.1. Textos Alternativos

Aunque el requerimiento más conocido relacionado con la accesibilidad sea el de proporcionar un texto alternativo para las imágenes (Pauta 1.1: Alternativas Textuales), es fácil encontrar textos alternativos inadecuados porque no se ha tenido en cuenta el tipo de imagen, su función y el contexto en el que se inserta o porque se ha perdido la objetividad [35].

Los usuarios que se benefician del texto alternativo de las imágenes son:

- Usuarios con conexiones lentas y navegadores con las imágenes deshabilitadas.
- Usuarios con poca o nula visión que utilizan lectores de pantalla.
- Usuarios con deterioro cognitivo que utilizan lectores de pantalla.
- Usuarios que utilizan navegadores de solo texto.

- Usuario que prefieren no bajar las imágenes para mejorar la velocidad.
- Usuarios que naveguen páginas con problemas para cargar las imágenes o códigos con errores.

Para entender cuál es el texto alternativo correcto es importante entender por qué está la imagen en nuestra página Web y cuál es el propósito, lo que llevará a entender qué se busca con la imagen e implementar un texto alternativo adecuado.

4.2.8.1.1. Métodos para Proveer Textos Alternativos

El principal método para proveer textos alternativos es el atributo alt en etiqueta img.

- Ventajas:
 - Soporte robusto para navegadores gráficos, no gráficos y para asistente tecnológicos.
 - Está explícitamente asociado con la imagen.
 - El atributo alt en la imagen no implica un impacto visual.
- Desventajas
 - No está disponible para algunos usuarios que podrían encontrarlo útil.
 - Si es muy largo puede causar problemas en algunos navegadores.
 - Si es muy largo puede causar problemas en algunos asistentes tecnológicos.
 - No se le puede agregar estructura semántica.

Si bien no hay exactamente una definición clara de lo largo que debería ser el texto alternativo ni de la cantidad de caracteres correcta, se puede establecer que entre 75 y 100 caracteres (1 o 2 sentencias) es aceptable para completarlo tanto en el atributo alt como en el elemento figcaption.

4.2.8.1.2. Textos Alternativos Útiles

Las imágenes agregadas a la galería son con la finalidad de que el usuario pueda ver experiencias vividas por personas que han visitado la ciudad y puedan reconocer los lugares a los que han ido. Las imágenes no tienen texto y no están contenidas por un elemento <a>, ni forman parte de un formulario. Tampoco son gráficos, mapas de imagen, ni están en la página solo como decorado. Este y otros criterios llevan a establecer los textos alternativos de cada una de las imágenes para el caso de estudio, fundamentalmente describiendo la imagen en el atributo alt y los detalles que no son posibles observar en el figcaption.

El caso de estudio expone la siguiente interface para la galería de imágenes:



Imagen 27. Galería de Imágenes aleatorias (solo usadas como ejemplo).

Con la siguiente codificación

```
<figure role="group">
```

```
<div>
```

```

```

```
</div>
```

```
<figcaption>
```

```
Leyenda Correspondiente
```

</figcaption>

</figure>

Por las razones que se ha comentado anteriormente.

4.2.8.2. Casos externos que presentan aspectos de inaccesibilidad en las imágenes

Ya hemos expuesto la relevancia de cumplir con uno de los criterios más conocidos de las WCAG 2.0: Todo el contenido no textual que se presenta

al usuario tiene una alternativa textual que cumple el mismo propósito (criterio de conformidad 1.1.1) Mientras más relevancia tenga la información que se presenta en imágenes y no se ofrece una alternativa textual, mayor es el grado de frustración que padecerá el usuario.

En la página oficial destinada a brindar información relativa a cómo obtener el nuevo DNI se ofrece una imagen que contiene una gran cantidad de información pero el texto alternativo es nulo.



Imagen 28. Página oficial con información para tramitar el nuevo DNI.

Una situación similar se da en la página <http://www.tramites.gov.ar> donde se ofrece información para obtener un certificado único de discapacidad pero

los usuarios a los que no les es posible visualizar la imagen no podrán contar con esa información ya que el texto alternativo es nulo.



Imagen 29. Captura de pantalla página interna³⁴ de <http://www.tramites.gov.ar>

³³ <http://www.nuevodni.gov.ar/inicio/>.

³⁴ http://www.tramites.gov.ar/multimedia/infografias/obtencion-certificado-unico-discapacidad-cud-territorio-nacional_t552

opción (cruz) para cerrar la ventana, esto no permitiría cumplir con un nivel A en las WCAG 2.0 dado el criterio de conformidad 1.4.1: Uso de color. Por esta razón en el Proyecto, se reescribieron los estilos para modificar los propuestos por defecto.

4.2.9.1. Técnicas para Ocultar Contenido

Otro aspecto de esta parte de la implementación, es la utilización del texto en el botón que abre la ventana emergente o modal. Textos como "Mas Info" o "Click Aquí" impiden lograr el cumplimiento de las WCAG 2.0 dado el criterio de conformidad 2.4.4: Propósito de los enlaces.

Agregar un botón con un mensaje tal como "Más Información" en el widget sería cometer el tipo de error nombrado anteriormente. A continuación se detalla una lista de técnicas para ocultar contenido y poder evitar este tipo de error, o sea, se puede mostrar el texto "Más información" pero en realidad tener el tener el texto "Más información de la fiesta de la cerveza", permitiendo también continuar minimalistas para usuarios que visualizan la pantalla:

Display: none.

Es de uso habitual, básicamente no ocupa lugar en el flujo de contenidos, y todo se acomoda como si no existiese el elemento ni sus descendientes. Varios artículos lo desaconsejan [36], mayoritariamente porque casi todos los lectores de pantalla en combinación con los navegadores dejan de leerlo. Por lo que no solo lo se está ocultando a los usuarios que visualizan la pantalla sino que a los que utilizan tecnologías de apoyo también.

Visibility:hidden

Es un caso similar al anterior display:none, los lectores de pantalla no leerán al usuario los elementos ni sus descendencias con esta propiedad. Por lo que si el objetivo es ayudar a los usuarios que ingresan con tecnologías de apoyo pero mantener más minimalista la pantalla este no es un recurso valido.

Text-indent:-999em

Text-indent:-999em envía el texto fuera de la pantalla y todos los lectores leerán el contenido. Esta técnica es la aconsejada por (entre otros) Katie Cunningham en el libro Accessibility Handbook, cuando escribe a cerca de ocultar texto para el usuario que ingresa sin lector de pantalla [27]. Sin embargo, esta técnica también presenta algunos inconvenientes que no son nombrados por Katie Cunningham:

- El usuario que visualiza la pantalla pierde el foco si este texto lo toma. Podría estar tabulando varias veces hasta encontrar donde está el foco ya que el mismo estaba fuera de la pantalla.
- En términos generales, Google penaliza las páginas que ocultan texto y si bien ocultar este tipo de texto (plano) no esté explicito, podría generar un problema.

Clip

La propiedad de CSS clip define qué porción de un elemento es visible y se aplica solamente sobre elementos con position:absolute. Clip no tiene la robustez de la técnica de enviar el texto fuera de la pantalla, pero la mayoría de los lectores de pantalla leen el contenido y consta con la ventaja de no traer ningún problema de SEO. Además, de querer si se quiere hacer visible el contenido para usuarios que usan el tabulador en elementos que reciben el foco, se puede hacer por ejemplo mediante las siguientes reglas:

```
button{
    position: absolute;
    clip: rect(0px,0px,0px,0px);
}
button:focus{
    position: static;
    clip: auto;
}
```

La sección de destacados cuenta con la siguiente interface:



Imagen 31. Destacados.

El enlace de “Más Información” contiene un texto oculto con el atributo de CSS clip comentado anteriormente, con lo que se logra cumplir con los criterios de accesibilidad.

4.2.9.2. Casos externos que presentan aspectos de inaccesibilidad en enlaces

Varios de los framework más utilizados en la actualidad, incorporan una utilidad mediante la cual es posible muy fácilmente limitar el contenido de un artículo a una cierta cantidad de palabras y ofrecer al

usuario un enlace a otra página con la información completa del artículo. Por esa razón gran cantidad de páginas web incumplen el criterio de conformidad 2.4.4: Propósito del enlace (en contexto) ya que no modifican esta utilidad que el framework por defecto ofrece.

A continuación se puede ver como en la página web de la comisión nacional asesora para la integración de las personas con discapacidad (CONADIS)³⁷ se comete este incumplimiento.



Imagen 32. Página web de CONADIS.

³⁷ www.conadis.gov.ar.

4.2.10. Formularios

Cuando se habla de la accesibilidad de los formularios, se refiere generalmente a que sean accesibles para las personas que utilizan lectores de pantalla. Las personas con otros tipos de discapacidad normalmente resultan menos afectadas si los desarrolladores no contemplan criterios de accesibilidad en estos elementos. La mayoría de los formularios que incumplen las WCAG 2.0 es específicamente en los criterio de conformidad 3.3.1 y 3.3.2. De todas formas, siempre es conveniente tener en cuenta otro tipo de discapacidad a la hora de crear componentes esenciales como el principal mecanismo de interacción con nuestros usuarios. Un ejemplo de esto sería tener en cuenta que generalmente las personas con problemas de audición, no pueden contestar al teléfono. En realidad para que una persona con problemas auditivos atienda el teléfono, la persona del otro lado de la línea debe saber y tomar medidas. Por lo tanto, se puede resumir que se deben establecer alternativas por si el usuario prefiere otro método de contacto que no sea el telefónico.

Varias de las nuevas incorporaciones de HTML5 para definir más precisamente los campos correspondientes a un formulario han sido comentadas anteriormente, como así también se ha hecho referencia a herramientas definidas por ARIA. La extensión de las características semánticas en formulario es extensa y variada, por lo que a lo largo de esta sección se enlistaran las ventajas y desventajas de cada una de las opciones.

Algunos aspectos genéricos a tener en cuenta son [39]:

- Garantizar que los Formularios sean Accesibles por Teclado, sean Lógicos y Fáciles de Usar: Cualquier elemento que queda fuera del alcance de algún usuario puede impedir que el mensaje llegue o que llegue correctamente. Por lo tanto, es necesario evitar pedir información innecesaria o redundante.
- Crear Etiquetas para los Elementos

de Formulario Utilizando el Elemento <label>: La etiqueta debe aparecer cerca del elemento del formulario. Cuando un lector de pantalla accede a un elemento del formulario que tiene un código <label> asociado a ella, se leerá el texto en el código <label> e indicará el tipo de elemento de formulario que es, por ejemplo: "Nombre. Cuadro de texto" o "Recibir Notificaciones. Casilla de verificación". Las etiquetas son necesarias para todos los elementos del formulario a excepción de los botones, ya que el lector de pantalla lee el texto que se encuentra en el botón, como por ejemplo: "Botón. Enviar".

ARIA cuenta con el elemento aria-describedby, un atributo que puede aumentar la comprensión. Otra ventaja de usar las etiquetas es que el usuario puede presionar en la etiqueta para establecer el foco en el elemento del formulario, lo cual es útil para algunas personas con discapacidad motora especialmente para la selección de casillas de verificación y botones de radio pequeños.

- Agrupar Elementos de Formulario mediante el Uso del Elemento <fieldset>: Cuando haya varios elementos de formulario asociados, se debería agrupar en el elemento fieldset. Cada fieldset debe tener una leyenda definida en la etiqueta legend que describe el grupo de asociados.

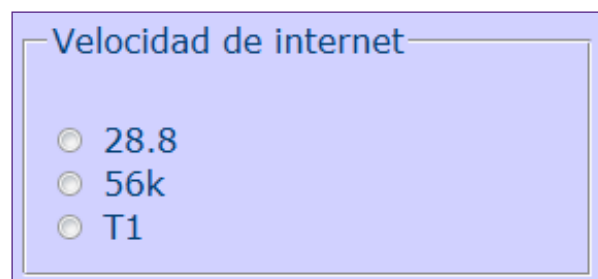


Imagen 33. Captura de pantalla del 3 elementos checkbox con su fieldset.

En la imagen anterior, la etiqueta de la primera casilla en el fieldset "Velocidad de Internet" es 28,8. Sin no se hubiera establecido la etiqueta fieldset, un lector de pantalla leería algo así: "Veintiocho

punto ocho. No se controla". Con el fieldset en su lugar, el lector pantalla leería "Velocidad de Internet: Veintiocho punto ocho. No se controla". Los fieldset se deben utilizar cuando hay grupos de casillas de verificación o botones de radio.

- Un Formulario debe tener un Único Propósito. Para no confundir al usuario, es importante que quede claro lo que debe esperar después de enviar el formulario.
- Evitar el Evento onchange. El evento onChange en los selects se activa cuando el usuario suelta el botón del ratón después de seleccionar la opción en el menú desplegable que ellos quieren. Sin embargo, si el usuario no está utilizando un ratón, deben utilizar las flechas del teclado para navegar por el menú desplegable. Cada vez que se mueven hacia arriba o hacia abajo por los elementos del menú, el código JavaScript detecta un evento onChange, que dirige el navegador a una página nueva. En su lugar, se debería permitir al usuario seleccionar el elemento del menú, y a continuación, que seleccione un botón de formulario adyacente para ir a la página según lo que ha seleccionado.

4.2.10.1. Complemento para Formulario

Hay una enorme cantidad de complementos que se pueden utilizar para implementar rápidamente un formulario en wordpress algunos gratuitos y otros pagos. La ventaja fundamental de su utilización es el ahorro en tiempo, ya que en general no solo completan la codificación del formulario sino que también algunos gestionan el envío de email y permiten modificaciones por parte del administrador del sitio. Sin embargo, la gran mayoría de complementos que se utilizan para implementar rápidamente un formulario en wordpress asume un compromiso parcial con el cumplimiento de los criterios de accesibilidad.

Contact form 7

Uno de los complementos con la finalidad

de facilitar la implementación de un formulario más populares y sencillos de usar, es Contact form 7. De hecho, es el segundo más descargado con este propósito, con más de 18 millones de descargas. Este complemento puede manejar múltiples formularios de contacto y además permitir al administrador personalizar el formulario y el contenido de notificación de correo de manera sencilla. En este caso de estudio se utiliza este componente para crear un formulario simple que contenga los campos email, teléfono y comentarios. Únicamente el email se comporta como requerido, el teléfono tiene un patrón a seguir y el comentario es un campo libre.

Accesibilidad

- Anuncio de Campos. Como se ha comentado anteriormente, una de las mayores ventajas de los complementos es que mayoritariamente ya constan con un panel de administración desarrollado para que el administrador pueda hacer cambios. Contact Form 7 a diferencia de otros complementos, no agrega un asterisco a la etiqueta label, sino que agrega textualmente 'requerido' en los formularios creados por defecto. Hacerlo de esta manera es apropiado, ya que no aclarar que el campo es requerido sería un error y no se estaría cumpliendo con las pautas WCAG 2.0.

Además de la palabra literal en el campo label, el plugin agregó el atributo aria-required útil tanto para los selects como para los inputs. Aria-required="false" se utiliza si el campo no es requerido, mientras que este mismo atributo con el valor true hace que los lectores de pantalla lo identifiquen como obligatorio y se lo haga saber a los usuarios.

Un campo sin este atributo establecido con el valor true sería anunciado de la siguiente forma (dependiendo el navegador): "Edita texto. Entrada de correo electrónico."

Un campo con este atributo establecido con el valor true: "Edita texto. Entrada

de correo electrónico obligatorio.”

Más adelante se detalla la opción `required` de HTML5 como atributo de una entrada de texto para obligar al usuario a completar el campo, aunque generalmente esto no es anunciado por los lectores de pantalla.

Respecto al tipo de campo, solo Chrome con Chromevox anuncia el tipo de campo con el que se debe completar la entrada de texto. Por ejemplo, si se trata de un teléfono, se anuncia que la entrada es de solo números. La misma situación se da con el placeholder, que por ahora solo lo lee Chrome con Chromevox.

- **Alertas.** Si bien el componente instalado trabaja con atributos de la nueva versión de HTML (como el placeholder), el mismo no propone la validación de esta misma forma, sino que por el contrario utiliza javascript para validar los campos. Los campos que se marcan como obligatorios y no están completos al momento del envío, así como los campos que fueron completados con un formato incorrecto, desencadenan una serie de mensajes al usuario con información que hace referencia al tema. Se desata un mensaje arriba del formulario comentando que ha ocurrido un error y se especifica abajo del campo cual ha sido el error (puede ser el formato o que no fue completado un campo requerido).

Todos los mensajes de error contienen el atributo `role` de ARIA con el valor `"alert"`, quedando por ejemplo en el componente `<div>` que contendría un mensaje en el formulario como `<div role="alert">`, `role=alert` es soportado por la mayoría de los navegadores modernos y por casi todas las tecnologías asistivas. Cuando un elemento que tiene un `role=alert` se muestra en pantalla, desencadena un evento de alerta en la API de accesibilidad del navegador. La alerta es escuchada por los asistentes de tecnología y es anunciado, generalmente nombrando la palabra `"alert"` antes o después del mensaje.

La validación nativa que se evalúa en el próximo punto, alerta al usuario de que no está completando un campo requerido, remarcando el campo, haciendo visible un Tooltips y llevando el foco al campo de entrada en cuestión.

- **Validación.** Usando HTML5 se puede identificar un campo como requerido, simplemente agregando la palabra como atributo de la entrada de texto. Un simple ejemplo quedaría:

```
<label for="address">Address:</label>  
  
<input type="text" name="address"  
id="address" required>
```

Una de las mayores ventajas de esta funcionalidad es que no se requiere de javascript para validar el formulario. Si el atributo está en la etiqueta y el navegador lo puede interpretar correctamente, al intentar enviar el campo requerido vacío, se remarcará el campo, se mostrará un mensaje de error y se volverá el foco al elemento de entrada en cuestión.

Verificación

- **Anuncio de Campos.** Especificar que el campo es requerido en el label del elemento es el mejor mecanismo para que un usuario que está visualizando la pantalla pueda interpretarlo y es la opción escogida para el caso de estudio, poner solo un asterisco es un error.

Agregar el atributo de WAI-ARIA `required=true` es una buena opción para que esté identificado semánticamente como requerido por los lectores de pantalla, si ambos están presentes los lectores de pantalla lo anunciarán 2 veces (el label y el Aria). Si además se incluye el atributo `required` de HTML5 el campo podría llegar a ser anunciado inclusive 3 veces, según el navegador y la combinación con el lector de pantalla. En la práctica, básicamente en todos los navegadores se está anunciando 2 veces.

La opción del label solo no conforma, dado que no contiene la información semántica, y el atributo ARIA solo

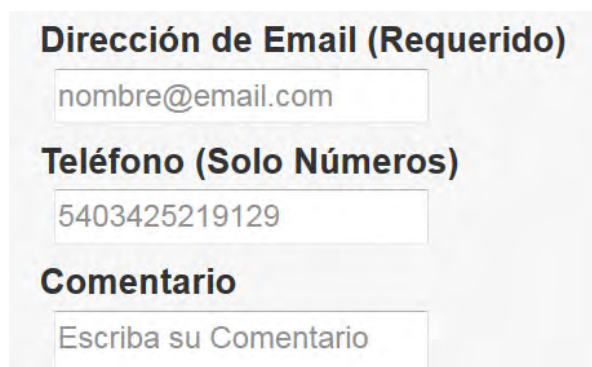
tampoco podría completar la codificación ya que no contiene la información para usuarios que visualizan la pantalla. En un futuro cuando los lectores de pantalla identifiquen el requerido y lo anuncien correctamente se podrá dejar de lado el atributo de ARIA.

El caso de estudio se completó con las 3 opciones, dado que el compromiso en la accesibilidad se puede ver afectado si no se anuncia el campo como requerido, pero no si se repite la información.

Agregar el atributo placeholder es una buena opción, pero aunque mayoritariamente los navegadores nuevos pueden interpretarlo, los lectores de pantalla no lo anuncian. Por lo tanto, no hay que agregar por este medio información ineludible y sigue siendo más robusto incluirlo como un texto de ejemplo bajo del campo, asociado al campo con aria-describedby. En el caso de estudio se optó por utilizar el placeholder pero sin que tenga información relevante.

Con el tipo de dato se tiene una situación semejante a la del placeholder; como se mencionó anteriormente, solo Chrome con chromevox anuncia de qué tipo es el nuevo campo, completando la lectura de un campo por ejemplo con "Edita texto. Entrada de correo electrónico obligatorios" o "Edita texto. Entrada numérica". Si el navegador no soporta el nuevo tipo de input, no tiene por qué suponer un problema, sino que lo interpretará como un type="text".

La interface del formulario sería:



The image shows a form with three input fields. The first field is labeled 'Dirección de Email (Requerido)' and contains the placeholder text 'nombre@email.com'. The second field is labeled 'Teléfono (Solo Números)' and contains the placeholder text '5403425219129'. The third field is labeled 'Comentario' and contains the placeholder text 'Escriba su Comentario'.

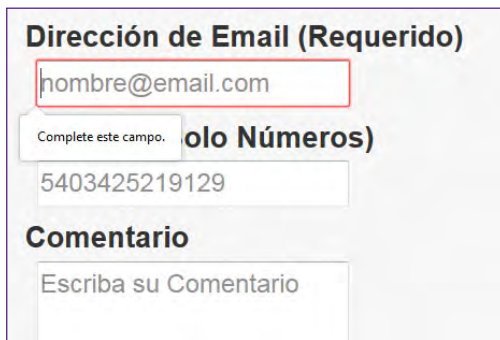
Imagen 34. Formulario.

- **Alertas.** Como se comentó anteriormente, para alertar a los usuarios con un mensaje, el complemento que se implementó en este caso utiliza el rol alert de ARIA. Dada la naturaleza del valor alert en el atributo role, no se mueve el foco en el navegador, sino que solo se anuncia lo que está dentro del elemento. Esto es conveniente cuando el objetivo es transmitir a los usuarios un mensaje sin interrumpir el flujo de navegación de la página, generalmente cuando el mensaje es positivo, es decir, cuando el mensaje es de éxito y se supone que el usuario querrá seguir avanzando por la página. Cuando lo que se quiere es transmitir un error o hacerle saber al usuario que su intento fue fallido, se considera más apropiado cambiar el foco hacia donde se ha cometido un error si es que el usuario tiene que rehacer alguna tarea, para así corregir el error. El funcionamiento del formulario en el caso de estudio se basó en este análisis.

El atributo required de HTML5 propone otro tipo de alerta: hacer visible un tooltips con un mensaje haciendo alusión al error cometido por el usuario y llevando el foco al campo de entrada en cuestión. Esto resulta muy conveniente, pero si bien el atributo esta soportado por los navegadores más utilizados desde hace ya varias versiones, no es así para los lectores de pantalla.

Contrariamente al atributo pattern y al placeholder, los mensajes son bien anunciados por JAWS en Firefox e Internet Explorer, pero no son anunciados en Chrome ni con JAWS ni con chromevox (el required ni siquiera es mostrado si chromevox está activado).

Por lo nombrado con anterioridad, en el caso de estudio se completó el formulario con la validación de HTML5. Más allá de los inconvenientes encontrados, se considera lo suficientemente sólido para ser implementado. La interface cuando se encuentre un campo obligatorio incompleta sería:



The image shows a web form with three sections. The first section is titled "Dirección de Email (Requerido)" and contains an input field with the text "nombre@email.com". A red border highlights the input field, and a red error message "Por favor, complete el campo requerido." is visible below it. The second section is titled "Número(s)" and contains an input field with the text "5403425219129". The third section is titled "Comentario" and contains an input field with the text "Escriba su Comentario".

Imagen 35. Alerta de error en formulario.

Y el mensaje de éxito quedaría:



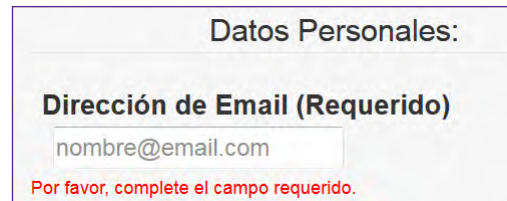
The image shows a web form with a success message at the top: "Su mensaje se ha enviado con éxito. Muchas gracias." Below the message, the form has a section titled "Datos Personales:" and a sub-section titled "Dirección de Email (Requerido)" with an input field containing "nombre@email.com".

Imagen 36. Alerta de éxito en formulario.

- Validación. Siempre está la opción de validar el formulario del lado cliente mediante javascript, y por ahora, es la opción más utilizada. Es usual aplicar librerías y tener una validación completa en poco tiempo. En el caso de no contar el navegador con javascript o tenerlo deshabilitado, no sería un inconveniente significativo ya que del lado del servidor el formulario también debería ser validado. Es muy adaptable, pero a mayor complejidad, más cantidad de código se escribe, por lo que si la validación es compleja, es la mejor opción.

La validación por HTML5 como se comentó funciona actualmente en todos los navegadores, pero no es posible decir que no tiene fallas. El prescindir de javascript y evitar destinar el tiempo al desarrollo una y otra vez, es una de los mayores beneficios de esta utilidad, mientras que la poca homogeneidad de los mensajes y formatos entre los navegadores, sumado a la no posibilidad de personalizarlos son las mayores desventajas. Según la misma W3C, la mejor manera de fomentar su uso y mejorar, será implementándola.

El caso de estudio implementa la validación de formularios nativa de HTML5, pero por ahora no resulta lo suficientemente fiable como para dejar de implementar la validación javascript por si HTML5 no puede ser interpretada por los navegadores. Si la validación se hace mediante javascript, el mensaje sería:



The image shows a web form with a section titled "Datos Personales:" and a sub-section titled "Dirección de Email (Requerido)" with an input field containing "nombre@email.com". A red error message "Por favor, complete el campo requerido." is visible below the input field.

Imagen 37. Validación de formulario por javascript.

4.2.10.2. Casos externos que presentan aspectos de inaccesibilidad en formularios

A continuación se hará mención a 2 sitios Web los cuales no cumplen con las WCAG 2.0.

La página web del Instituto Nacional de la Administración Pública³⁸ (INAP) presenta a primera vista un formulario accesible bajo los criterios de la WCAG 2.0. Pero está muy lejos de eso. Como primera medida el texto que se encuentra en la parte superior del input no es su label correspondiente por lo que no es leído por el lector de pantalla. Además de eso el select tampoco puede interpretarse correctamente ya que tampoco contiene su label correspondiente. Esto hace que el criterio de conformidad 3.3.2: Etiquetas o instrucciones, este incumplido.

³⁸ https://capacitacion.inap.gov.ar/?page_id=3218.



Imagen 38. Página web de INAP.

La página web del Instituto Nacional contra la Discriminación, la Xenofobia y el Racismo (INADI)³⁹ presenta un formulario superador al ejemplo anterior pero con errores de accesibilidad evitables. Hemos nombrado los riesgos de marcar al usuario mediante asteriscos la información obligatoria que debe cumplir y la necesidad de tener en cuenta la robustez en los desarrollos, no obstante esto, la presentación del formulario esté bien estructurada y es posible completar los datos. Según lo visto anteriormente a lo largo del Proyecto se podrían modificar algunos aspectos para hacer más accesible el formulario y cumplir con el criterio de conformidad 3.3.1: Identificación de errores.

Luego de presionar sobre el botón para enviar el formulario un usuario que no visualiza la pantalla no puede saber si el formulario se ha enviado o no, ya que esto no se informa correctamente. Que el reclamo no se ha enviado solo se podrán enterar los usuarios que vuelvan a recorrer la pantalla. En el caso de que lo hagan, el mensaje "Hubo un problema con los datos completados. Se resaltan los errores a continuación" los pondrá al tanto de esto.

³⁹ <http://inadi.gob.ar/>.

Imagen 39. Formulario con errores resaltados de la página web del INADI.

5. Revisión y Conclusión

Se escogió estructurar el proyecto de la manera más apropiada para presentar una documentación concisa, con descripciones prácticas de las implementaciones escogidas. El caso de estudio expone la mejor alternativa para el correcto desarrollo de cada requisito logrando conseguir un nivel A según las WCAG 2.0.

La W3C propone varias herramientas para hacer la validación de accesibilidad⁴⁰ y una oficial para hacer validación de código⁴¹. Se puede decir que los validadores automáticos son una herramienta útil y práctica a la hora de verificar que se está cumpliendo con los criterios que impone cada nivel, pero ninguno da la información completa, por lo que, la mejor manera de verificar que la accesibilidad se está cumpliendo, es ir punto por punto comprobando que la página en cuestión cumpla con todo lo necesario para alcanzar el nivel deseado.

5.1. Verificación Punto por Punto

A continuación se especifican los criterios de conformidad de nivel A y se detalla en qué situación del caso de estudio se debió tener en cuenta cada criterio. Todos los criterios de Nivel A se cumplen en el caso de estudio realizado para este proyecto.

- Criterio de Conformidad 1.1.1: Contenido no Textual. En el caso de estudio presentado se refiere a las imágenes y el mapa con los prestadores de servicio. En la sección "galería de imágenes" se identifican los criterios generales seguidos para establecer alternativas a este contenido no textual de forma correcta, y en la sección "mapa" se establece una de las alternativas para el contenido no textual que brinda Google.
- Criterio de Conformidad 1.2.1: Sonido Solo o Video Solo. En la sección "Música" se describe cómo cumplir con el criterio y se detallan además otros puntos a tener en cuenta para obtener un mayor nivel

de accesibilidad.

- Criterio de Conformidad 1.2.2: Captions. En la sección "Video" se describe cómo cumplir con el criterio. Para este caso de estudio se optó por la incorporación de herramientas nativas.
- Criterio de Conformidad 1.2.3: Audiodescripción o Medio Alternativo. En la sección "Video" se describe cómo cumplir con el criterio. Para este caso de estudio se optó por la incorporación de herramientas nativas.
- Criterio de Conformidad 1.3.1: Información y Relaciones. Es un criterio de alcance global, que involucra todas las secciones de una página. A lo largo del Proyecto se detallaron las opciones semánticas con la que se completó la implementación.
- Criterio de Conformidad 1.3.2: Secuencia Significativa. Es un criterio de alcance global e involucra cada una de las secciones de una página. Cada una de las etiquetas propuestas se justificó a lo largo del documento.
- Criterio de Conformidad 1.3.3: Características Sensoriales. Es un criterio de alcance global y se debe tener en cuenta a la hora de diseñar los componentes, así como se tuvo en cuenta en este caso de estudio.
- Criterio de Conformidad 1.4.1: Uso de Color. Es un criterio de alcance global y se debe tener en cuenta a la hora de diseñar los componentes. Para este criterio los validadores automáticos son de gran utilidad.
- Criterio de conformidad 1.4.2: Audio Control. Este criterio no aplica a nuestro caso de estudio, ya que la página no cuenta con sonido que se reproduzca automáticamente.
- Criterio de conformidad 2.1.1: Accesible por Teclado. Es un criterio de alcance global y se debe controlar de forma correcta en cada sección. En apartados "Barra de navegación" y "Multimedia" se hace referencia a su cumplimiento.

⁴⁰ <http://www.w3.org/WAI/ER/tools/>.

⁴¹ <http://validator.w3.org/>.

- Criterio de conformidad 2.1.2: Sin Trampas para el Teclado. Es un criterio de alcance global y se debe controlar de forma correcta en cada sección. Herramientas para cumplir con este criterio se ven en el apartado "Acceso mediante teclado".
- Criterio de Conformidad 2.2.1: Tiempo Ajustable. Este criterio no aplica a nuestro caso de estudio. Nuestra página Web no cuenta con contenido que requiera la interacción de un usuario en tiempo finito.
- Criterio de conformidad 2.2.2: Poner Pausa, Detener, Ocultar. Este criterio no aplica a nuestro caso de estudio. La página web desarrollada para el caso de estudio no contiene información que se mueva, parpadee, se desplace o se auto-actualice.
- Criterio de Conformidad 2.3.1: Umbral de Tres Destellos o Menos. Este criterio no aplica a nuestro caso de estudio. La página Web desarrollada no contiene ningún elemento que destelle.
- Criterio de Conformidad 2.4.1: Evitar Bloques. Se detalló una forma de hacerlo con el bloque de navegación en la sección "Barra de navegación", incorporando un enlace al contenido.
- Criterio de Conformidad 2.4.2: Título de Página. Es un criterio de alcance global que está muy relacionado con el SEO, razón por la que se lo debe tener en cuenta. El CMS propone una manera casi transparente de hacerlo.
- Criterio de Conformidad 2.4.3: Orden del Foco. Es un criterio de alcance global, la secuencia en la que se va recibiendo el foco se trata en el apartado "acceso mediante teclado".
- Criterio de Conformidad 2.4.4: Propósito del Enlace (en contexto). En la sección "destacados" se describió cómo modificar los enlaces para cumplir con este criterio (y criterios de mayor nivel).
- Criterio de Conformidad 3.1.1: Idioma de la Página. Es un criterio de alcance

global que está muy relacionado con el SEO, razón por la que se lo debe tener en cuenta. El CMS propone una manera casi transparente de hacerlo, aunque no propone cómo de cumplir con criterios similares de mayor nivel.

- Criterio de Conformidad 3.2.1: En Foco. Es un criterio de alcance global que está muy relacionado con el diseño. Nuestro desarrollo lo cumple ya que cualquier componente que recibe el foco, no inicia ningún cambio en el contexto.
- Criterio de conformidad 3.2.2: En Entradas. Es un criterio muy relacionado con los formularios. En la sección "formularios" se describe como un aspecto general a tener en cuenta.
- Criterio de Conformidad 3.3.1: Identificación de Errores. Es un criterio muy relacionado con los formularios. En la sección "formularios" se detallan opciones y se brinda una conclusión específica de este punto.
- Criterio de Conformidad 3.3.2: Etiquetas o Instrucciones. Es un criterio muy relacionado con los formularios. En la sección "formularios" se detallan opciones y se brinda una conclusión específica de este punto.
- Criterio de Conformidad 4.1.1: Procesamiento. Es un criterio de alcance global que no solo debe cumplirse para completar la accesibilidad, sino también para evitar errores de validación con consecuencias en posicionamiento y en estructura de la página.

5.2. Tendencia Hacia la Accesibilidad

La redacción de este proyecto aporta a los avances relacionados a la accesibilidad Web [51]. Gobiernos de todo el mundo, incluyendo el Argentino exigen el seguimiento de pautas concretas y en algunos países hasta aplican severas sanciones económicas. Gran parte de los complementos investigados en este Proyecto para su incorporación en el caso de estudio presentado han incorporado aspectos de accesibilidad recientemente.

La W3C como organismo regulador está haciendo lo necesario para permitir que los desarrolladores tengan las herramientas necesarias para poder brindar a los usuarios la accesibilidad que requieren. El lenguaje HTML5 puede considerarse como un claro ejemplo y más aun WAI-ARIA que está en su primera versión y fue ideado y concebido exclusivamente para mejorar la accesibilidad de las páginas.

Sin embargo y lamentablemente, también se percibió a lo largo de la investigación para el Proyecto, mucha ignorancia en el tema. Desarrolladores con experiencia no solo no aplican las Pautas de Accesibilidad para el Contenido Web, sino que no saben a qué se refiere el término. Aunque las regulaciones estén y en Argentina las empresas subsidiadas total o parcialmente por el estado puedan ser sancionadas por el incumplimiento de la Ley N° 26.653 de Accesibilidad de la Información en las Páginas Web⁴² que les exige cumplir con las WCAG 2.0 estas no se cumplen. Páginas de relevancia no cumplen con las pautas de accesibilidad al contenido Web, como :

- <http://www.argentina.gob.ar>.
- <http://www.mininterior.gov.ar>.
- <http://www.desarrollosocial.gov.ar>.
- <http://www.trabajo.gob.ar/>.
- <http://www.msal.gov.ar/>.
- <http://www.afip.gov.ar/>.
- CONADIS: Comisión Nacional Asesora para la Integración de las Personas con Discapacidad.
- IPRODICH: Instituto Provincial para la Inclusión de las Personas con Discapacidad.
- IPRODI: El Instituto Provincial de Discapacidad (Entre Ríos).
- INADI: Instituto Nacional contra la Discriminación, la Xenofobia y el Racismo.

⁴² <http://www1.hcdn.gov.ar/dependencias/dsecretaria/Periodo2009/PDF2009/>.

Incorporar conceptos de accesibilidad en las páginas Web tiene su costo, más aun si no se han tenido en cuenta en el inicio del proyecto. Resulta desalentador, pero tanto en la etapa de desarrollo como fundamentalmente en la de testeo sumar estos criterios a un sitio Web agrega un tiempo valioso que seguramente actualmente no se podrá cobrar.

Según informes recientes⁴³ el aumento de empleo de personas con algún tipo de discapacidad va en aumento exponencial. Esto hace que el consumo de este sector reflote, tanto en bienes de consumo como en servicios (como turismo) por lo cual las expectativas para contar con sitios Web más accesibles aunque lejanas son relativamente buenas ya que sería posible cobrar por estos.

El proyecto brinda una noción básica de las limitación con las que pueden contar los usuarios de un sitio web, además, especifica los recursos ofrecidos por los lenguajes de programación para dotar de accesibilidad a las páginas, propone detalles respecto al desarrollo accesible de muchos de los componentes que generalmente podemos ver en la mayoría de los sitios web y entrega un caso de estudio que logra un nivel A de las WCAG 2.0. Aunque el PFC deja muchos conceptos útiles a la hora de buscar un nivel mayor.

6. Líneas Futuras de Investigación

El análisis de determinados requisitos habituales en muchas de las implementaciones, pueden requerir un tiempo con el cual los desarrollos Web generalmente no cuentan. Aunque en el caso de estudio queda plasmado un desarrollo que cumple las pautas de accesibilidad, se entiende como insuficiente la teoría recopilada como para ser plasmada en el cuerpo de este PFC. Las siguientes tres secciones reflejan ejemplos:

6.1. Comentarios

Tener la alternativa de permitirles a

⁴³ http://www.fundacionadecco.es/_data/SalaPrensa/SalaPrensa/Pdf/714.pdf.

los usuarios dejar sus comentarios es cada vez más común en las páginas de Internet. Entre otras cosas, esto puede proporcionar una buena noción sobre el tipo de visitantes que leen el artículo. Wordpress incorpora un sistema de gestión de comentarios propio permitiendo insertar comentarios en cualquier entrada. El sistema de comentarios que viene por defecto, cumple su función, pero no es lo suficientemente potente como lo que la mayoría busca para su sitio Web.

Existen varias alternativas para modificar el funcionamiento que se propone por defecto en el CMS, y una de ellas es Disqus. Disqus es un servicio en línea que ofrece una plataforma informática centralizada para discusiones en blogs y sitios Web. Actualmente es uno de los más descargados en su tipo, con más de 1.500.000 descargas, tiene buena documentación (incluso tiene un apartado de accesibilidad) [42], brinda soporte técnico, provee al administrador la habilidad para modificar las autorizaciones de comentarios y una buena cantidad de alternativas para acceder mediante redes sociales.

Disqus funciona de la siguiente forma: si un usuario entra a un sitio Web y remite un comentario en un sitio que utilice este sistema, el comentario es enviado a disqus,

que verifica si la persona que realizó el comentario tiene una cuenta allí, y rastrea todos los comentarios que haya hecho en todos los sitios Web que utilicen Disqus. Disqus permite dejar comentarios en cualquier sitio Web, ingresando con las credenciales de Facebook, Twitter, Google, OpenID o Yahoo, lo que facilita a todos los usuarios a opinar sobre cualquier tema de manera fácil y cómoda.

Según una auditoria de accesibilidad llevada a cabo en el año 2010 por Nomensa⁴⁴, basada en las Pautas de Accesibilidad WCAG 2.0, los temas a tratar para mejorar la accesibilidad eran 8. Para el año 2012 disqus renovó en gran parte su interface y corrigió estos puntos.

Actualmente no se han encontrado inconvenientes con el nivel de accesibilidad A del plugin; cumple con las Pautas de Accesibilidad plantadas por la WAI y coincide con lo expuesto de parte de la empresa desarrolladora donde afirma una correcta navegación con teclado y determinados lectores de pantalla⁴⁵.

Para el caso de estudio se ha agregado el plugin mencionado. Como se comentó anteriormente no compromete el nivel de accesibilidad del sitio. La interface del elemento sería:

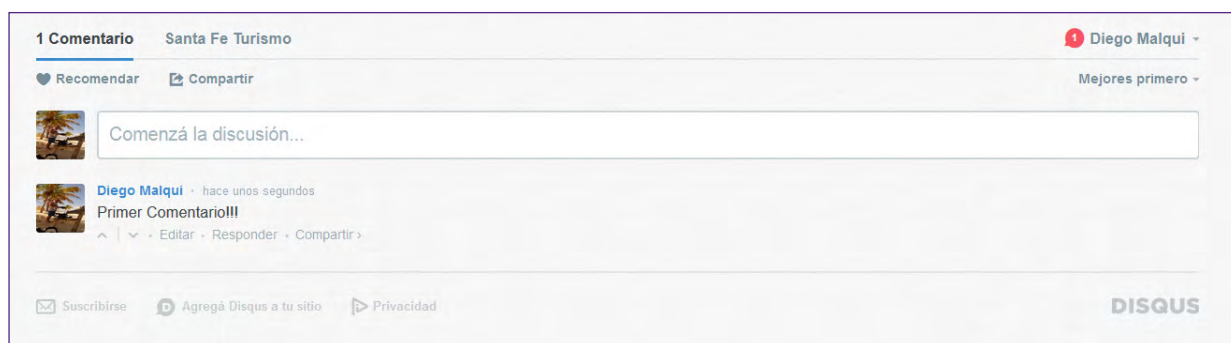


Imagen 40. Interface de Disqus.

6.2. Venta de Productos

Para ejemplificar con números el crecimiento de compras por Internet en nuestro país, se puede usar el denominado

CyberMonday, la mayor campaña nacional de ventas por Internet con promociones y descuentos, llevado a cabo el pasado 10 y 11 de noviembre del 2014. Según la

⁴⁴ <http://accessibility.nomensa.com/>.

⁴⁵ <https://help.disqus.com/customer/portal/articles/1424209-accessibility-on-disqus>

afirma la Cámara Argentina de Comercio Electrónico⁴⁶ (CACE), la campaña generó un 245% de aumento en las ventas y 290% de crecimiento en la cantidad de productos vendidos, comparado con el CyberMonday Argentina 2013. Además la cantidad de empresas que participaron creció casi un 400% con respecto a la edición anterior.

Dado el crecimiento de los usuarios que utilizan el comercio en línea (e-commerce) para comprar algún producto, han surgido utilidades para los desarrolladores, como ser PayPal. PayPal es un sistema que permite a cualquier usuario con una dirección de correo electrónico enviar y recibir pagos de forma segura por Internet usando su tarjeta de crédito, tarjeta de débito o cuenta bancaria. Actualmente tiene más de 100 millones de usuarios, y brinda soporte técnico para integrar sus funcionalidades a los más conocidos gestores de contenido, fácilmente.

Según la Organización Mundial de la Salud (en el 2003), se estima que en el año 2013 había en todo el mundo entre 40 y 45 millones de ciegos y 135 millones de personas con baja visión⁴⁷. Si tan solo el 1% de estos usuarios usan Internet para comprar productos, ya es un número significativo por el cual valdría la pena tener una página accesible [48]. Un sitio Web que contenga un acceso sencillo a la compra y sea accesible, contará con un grupo mayor y leal de compradores.

Más allá del aspecto moral el cual debería regir en todo el mundo, en países donde las pautas WCAG 2.0 son ley, no cumplirlas puede traer aparejado condenas económicas. El ejemplo más resonante ha sido cuando se condenó a target.com a pagar 6 millones de dólares a la federación nacional de ciegos por no codificar de forma accesible su sitio Web [46].

Cualquier sitio Web que presente una sección de comercio electrónico, es importante que cuente con los recursos

necesarios para hacer que personas con algún tipo de discapacidad puedan acceder a realizar una compra. Para muchos tipos de sitios si un potencial cliente no puede realizar la compra de un producto, carece de importancia lo que haya hecho en el resto del sitio por la accesibilidad, ya que todo se resumía en la compra de un producto.

Tanto la galería de productos, en la cual el usuario visualiza los productos, como el proceso de compra, que se inicia al dejar la galería de productos, y es donde los usuarios pueden revisar, eliminar o actualizar sus productos, se deberían cumplir determinados criterios para satisfacer las Pautas de Accesibilidad.

Galería de Productos

Los errores comunes que suelen cometerse en el desarrollo de cualquier página, son generalmente los mismos que se comenten en galerías de productos, la diferencia está en que en los últimos casos no solo se priva de información a los usuarios, sino que además se pierden potenciales clientes.

Los errores más comunes que se pueden ver en las galerías son:

- Escoger tipografías para títulos, ofertas o destacados, que resultan difíciles de leer a personas con algún tipo de deficiencia visual y utilizar el color como único medio de comunicación.
- Utilizar textos alternativos incorrectos. Este aspecto es importante porque si usuarios con posibilidad de visualizar los productos en fotos, a veces sienten desconfianza antes de comprar, más desconfianza sienten aquellos que no pueden ni siquiera ver la foto.
- Categorizar de manera incorrecta los productos. La categorización adecuada de productos debe de hacerse de manera tal que el usuario encuentre los productos rápido. Si bien en este caso de estudio no afecta porque son pocos los productos ofrecidos, es un error importante a obviar.

Si en términos generales es difícil

⁴⁶ <http://www.cace.org.ar/>.

⁴⁷ <http://www.who.int/mediacentre/news/releases/2003/pr73/es/>.

encontrar complementos que brinden soluciones accesibles de desarrollo, más difícil aún es encontrar soluciones accesibles para el proceso de compra. Generalmente, las páginas no solo no cumplen las pautas establecidas por la WAI, sino no intentan hacerlo. El error más común en las galerías es que la mayoría de los plugins utiliza Ajax para actualizar el carro de compras y no informan a las tecnologías de apoyo de forma adecuada si la actualización se llevó a cabo correctamente o no.

El caso de estudio que se desarrolló para este PFC integra utilidades que brinda PayPal [50], fundamentalmente por estar muy bien documentada y porque las otras opciones no tenían ninguna aproximación al cumplimiento de las Pautas de Accesibilidad para el Contenido Web. Siguiendo con esta línea, en el caso de estudio se integró el botón "Agregar al carro" o "add to cart" mediante un código que brinda la documentación oficial para ser escrito directamente en la página.

La Interface lograda es:

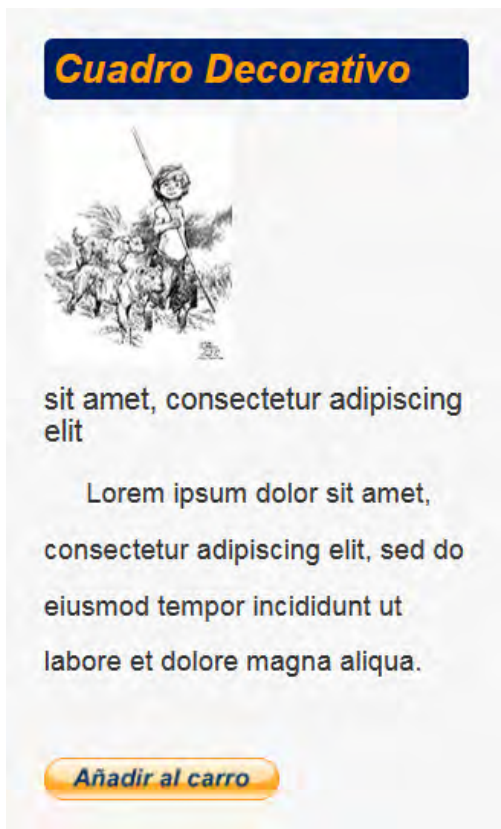


Imagen 41. Galería de productos.

Sin embargo, y a pesar de la elección de PayPal, algunas utilidades de PayPal no han sido aplicadas al caso por no coincidir con los parámetros de accesibilidad. Por ejemplo, la empresa brinda como código por defecto lo que aparece a continuación, cuando según la W3C [35] para una imagen mediante la cual se va llevar a cabo una acción, el texto alternativo debe representar la acción. En este caso, el texto alternativo que propone PayPal es incorrecto ya que es una especie de eslogan y no representa la acción. Un usuario que ingresa con lector de pantalla podría por error, interpretar el eslogan como una marca o un anuncio.

```
<input type="image" src="https://www.paypalobjects.com/en_US/i/btn/btn_cart_LG.gif" border="0" name="submit" alt="PayPal - The safer, easier way to pay online!">
```

Además, para una galería más grande de productos, esta implementación sería incorrecta, ya que faltaría una categorización de los productos, utilización de filtros y la posibilidad de ordenar los productos según determinados parámetros. A una galería más grande también se le debe agregar un paginado para no extender mucho la galería de forma vertical.

Proceso de Compra

El proceso de compra es conocido también por su nombre en inglés "checkout" que comienza cuando el usuario deja la galería de productos seleccionando uno o varios productos. Las Pautas de Accesibilidad al Contenido Web requieren que este paso intermedio entre la galería de productos y el proceso final de compra cumplan con 4 funciones específicas:

1. Brindar al usuario la posibilidad de ver un resumen de lo que está por comprar.
2. Brindar al usuario la habilidad de actualizar (borrar o cambiar de cantidad) los ítems que esta por comprar.
3. Brindar al usuario la posibilidad de volver a visualizar los productos antes de continuar.

4. Brindar al usuario la posibilidad de continuar con el proceso de compra.

Las opciones se brindan en una tabla bien definida, tal como se estableció previamente, pero no se coincide en algunas cuestiones simples cuando los productos son muchos. Si un usuario pretende repasar qué es lo que compró, solo con el nombre posiblemente no pueda hacerlo; puede que lea la

descripción pero que otra sea similar, o pudo haber leído varias descripciones y no recordar el producto según el nombre. Una opción para arreglar este inconveniente es agregar la descripción o mínimamente una imagen en miniatura.

La interface con la que se maneja paypal por defecto y la implementada para el caso de estudio es:

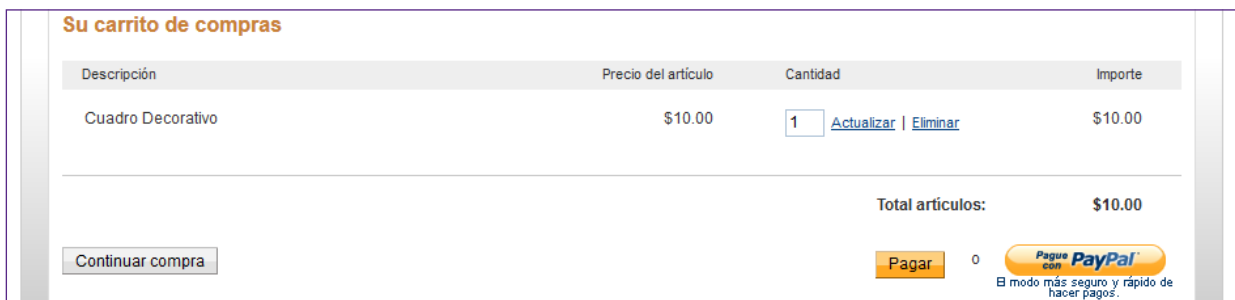


Imagen 42. Proceso de compra - Resumen.

Pago

El proceso de pagos para una persona con algún tipo de inconveniente en sus habilidades, es aún más complejo que el proceso de compra. Si bien como se comentó, este caso de estudio implementó PayPal, también PayPal sufre errores.

PayPal contiene dos bloques en la página de pago: uno para abrir una cuenta y otro si ya se es usuarios del sistema. En ambos casos los formularios trabajan de la misma forma, donde visualmente es posible identificar que se trata de un error, pero se entiende que los textos son por lo menos mejorables. En ningún caso el usuario entenderá que completó mal el formulario; lo que genera que el usuario no sepa que la información que introdujo es incorrecta y debe modificarlo, sino que crea que ocurrió un error externo a él y simplemente se proponga volver a completar información (información errónea).



Imagen 43. Proceso de compra - pago.

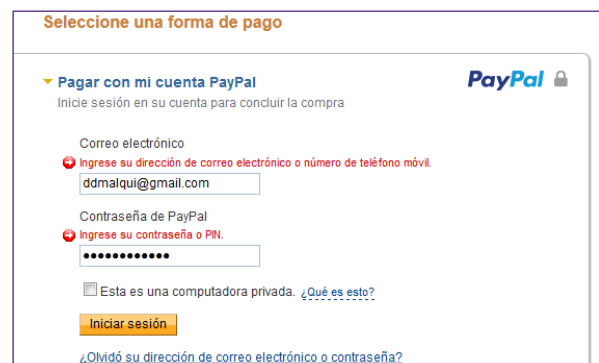


Imagen 44. Proceso de compra - pago - formulario incompleto

6.2.1. Proceso completo

La implementación de un proceso de compra completo con registros en sesión, validación de tarjetas mediante cada API, creación de interface y testeo entre otras cuestiones, ocupa un tiempo considerable y generalmente no es desarrollado desde el inicio en proyectos chicos.

Si bien hay muchos complementos capaces, flexibles, seguros y administrables, no se logró encontrar ninguno con características de accesibilidad sólidas. Siendo más específico, no se encontró ningún complemento que detalle en su documentación, las características de accesibilidad con las que cuenta.

La opción de PayPal es la más atractiva y la que mejor se ajusta al caso de estudio presentado en este PFC, porque brinda la información necesaria para su correcta implementación. Además, es uno de los sistemas de pago más difundidos y utiliza las tarjetas de créditos más extendidas por el mundo; es una empresa comprometida para con la accesibilidad y consta con pocas deficiencias. PayPal es una empresa con infinidad de recursos, y aquellos puntos en los que este trabajo discierna con PayPal porque se entiende pueden ser modificados sin perder ninguna funcionalidad, han sido justificados.

6.3. Mapa

Google Maps es un servidor de aplicaciones de mapas que pertenece a Google y ofrece imágenes de mapas desplazables, así como fotografías por

satélite del mundo. Inclusive, con Google Street View se pueden ver imágenes de las calles del mundo, incluso se incorporó no hace muchos años a la Argentinas.

Los mapas presentes en las páginas Web generalmente utilizan esta herramienta o pueden presentar alguna semejante, y siempre presentan barreras para los usuarios con algún tipo de impedimento.

Google map basa su funcionamiento en la interacción con el mouse. Modificar esto no solo es un trabajo de desarrollo complejo en javascript, sino que probablemente no se tenga acceso a todas las opciones y si la API de google cambiara la implementación que se desarrolló posiblemente la misma quede obsoleta.

La opción más extendida para este tipo de situaciones es proveer al usuario de contenido textual. De esta manera se cumplen las Pautas de Accesibilidad y se brinda un fácil acceso a la información, tanto a los usuarios imposibilitados de utilizar algún sentido, como al resto de los usuarios.

Para nuestro caso de estudio se brindó a los usuarios enlaces a diferentes páginas sobre cada categoría. Dentro de estas páginas, los usuarios pueden encontrar detalles de cada negocio brindados en tabla o servicios de esa categoría. Es sumamente necesario asegurarse que la opción accesible seleccionada para ofrecer al usuario, sea realmente accesible. La siguiente imagen refleja la interface de sección correspondiente al mapa:

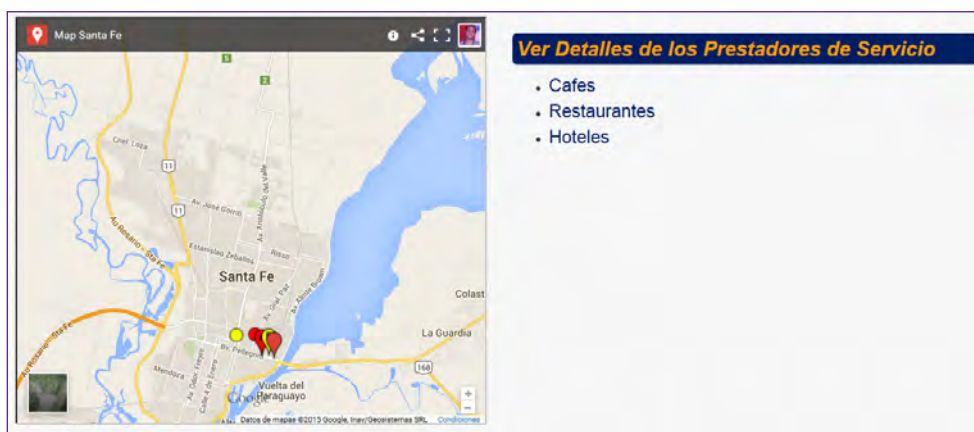


Imagen 45. Mapa.

Bibliografía

- [1] Mike Paciello, Octubre del 2000. Introducing web Accessibility, Web Accessibility for People with Disabilities (pag. 3-23), New York: CMPBooks.
- [2] Mark Pilgrim, Dive Into HTML5, disponible en <http://diveintohtml5.info/> al 10/12/2014
- [3] Matthew MacDonald, Diciembre Del 2013. HTML5: The Missing Manual, O'Reill.
- [4] Jennifer Niederst Robbins, Agosto Del 2013. HTML5 Pocket Reference: O'Reill.
- [5] HTML: The Markup Language (an HTML language reference), disponible en <http://www.w3.org/TR/html-markup/aside.html> al 05/12/2014.
- [6] Introduction: towards more simplicity, new structuring elements, disponible en <http://classroom.w3devcampus.com/mod/book/view.php?id=1225> al 20/07/2015.
- [7] Jon Duckett, Noviembre del 2011. Form, HTML and CSS: Design and Build Websites (pag. 144-176), Indianapolis Indiana: Paperback.
- [8] Aaron Andersen, Mayo del 2010. Future Web Accessibility: HTML5 <input> Extensions, disponible en <http://Webaim.org/blog/future-Web-accessibility-html5-input-extensions/> al 05/12/2014
- [9] Roger Hudson, Mayo del 2013. Accessible Forms 2: Required Fields and Extra Information, disponible en <http://usability.com.au/2013/05/accessible-forms-2-required-fields-and-extra-information/> al 05/12/2014
- [10] The Ogg container format, disponible en xiph.org/ogg/ al 07/12/2014
- [11] Octubre del 2014. video coding standards, Digital Video Concepts, Methods, and Metrics: Quality, Compression, Performance, and Power Trade-off Analysis (pag 125-155), New York: Apress Open.
- [12] About WebM, disponible en <http://www.webmproject.org/about/> al 19/07/2015
- [13] Mark Pilgrim, Agosto del 2010. Detecting HTML5 Features, Detecting HTML5 Features (pag. 15-30). Estados Unidos: O'Reill.
- [14] MP3, disponible en en.wikipedia.org/wiki/MP3 al 07/12/2014
- [15] Matthew MacDonald, Diciembre Del 2013. Audio and Video, HTML5: The Missing Manual (pag. 143-169), Estados Unidos: O'Reill.
- [16] Jean-Yves Perrier, Mayo del 2014, Usando aplicaciones offline disponible en https://developer.mozilla.org/es/Apps/Using_apps_offline al 17/12/2014
- [17] Agosto del 2013, Accessible CSS, disponible en <http://Webaim.org/techniques/css/> al 05/12/2014
- [18] Yi Min Shum, Abril del 2015, Responsive design, el factor crucial para el nuevo algoritmo de Google, disponible en <https://platzi.com/blog/responsive-design-seo/>
- [19] James Craig and Michael Cooper, marzo del 2014, Accessible Rich Internet Applications (WAI-ARIA) 1.0, disponible en <http://www.w3.org/TR/wai-aria/> al 15/08/2014
- [20] Supported States and Properties, disponible en http://www.w3.org/TR/wai-aria/states_and_properties al 05/11/2014
- [21] Joshue O Connor, Mayo del 2012. Pro HTML5 Accessibility. Estados Unidos: Apress.
- [22] Heydon Pickering, Julio del 2006, Apps For All: Coding Accessible Web Applications, Alemania: Smashing Magazine GmbH.
- [23] Noviembre 2014, Designing for Screen Reader Compatibility Disponible en <http://webaim.org/techniques/screenreader/> al 19/07/2015
- [24] Introducing ChromeVox, disponible en <http://www.chromevox.com/> al 11/10/2014
- [25] Welcome to Orca, disponible en

<https://help.gnome.org/users/orca/stable/introduction.html> al 11/10/2014.

[26] Shadi Abou-Zahra, Agosto del 2012, Diversity of Web Users, disponible en <http://www.w3.org/WAI/intro/people-use-web/diversity> al 16/07/2015

[27] Katie Cunningham, Septiembre 2012, Accessibility Handbook, Estados Unidos: O'Reill

[28] Michael Cooper, Andrew Kirkpatrick and Joshue O'Connor, Febrero 2015, Understanding Conformance, Understanding WCAG 2.0: A guide to understanding and implementing WCAG 2.0, disponible en <http://www.w3.org/TR/UNDERSTANDING-WCAG20/conformance.html> al 05/05/2014.

[29] Shawn Lawton Henry and Matt May, Julio del 2005, User Agent Accessibility Guidelines (UAAG) Overview, disponible en <http://www.w3.org/WAI/intro/uaag.php> al 15/10/2014.

[30] Shawn Lawton Henry, Julio del 2005, Authoring Tool Accessibility Guidelines (ATAG) Overview, disponible en <http://www.w3.org/WAI/intro/atag.php> al 15/10/2014.

[31] Margarita Sebastián Herranz, Febrero del 2011, "Guía de orientación en la práctica profesional de la valoración reglamentaria de la situación de dependencia: Productos de Apoyo para la Autonomía Personal" disponible en <http://www.ceapat.es/InterPresent1/groups/imsero/documents/binario/guadeproductosdeapoyo.pdf> al 20/07/2015.

[32] Steve Faulkner, Febrero del 2013, Using WAI-ARIA Landmarks, disponible en <http://www.paciellogroup.com/blog/2013/02/using-wai-aria-landmarks-2013/> al 17/10/2014.

[33] Joseph Scheuhammer and Michael Cooper, Marzo del 2013, "WAI-ARIA 1.0 Authoring Practices", disponible en <http://www.w3.org/TR/wai-aria-practices/#menu> al 15/10/2014.

[34] Mayo del 2012, "Survey of Preferences of Screen Readers Users", disponible en Webaim.org/projects/screenreadersurvey/#skipnav al 10/10/2014.

[35] Steve Faulkner, Octubre del 2012, HTML5: Techniques for providing useful text alternatives, disponible en <http://dev.w3.org/html5/alt-techniques/> al 10/10/2014.

[36] Peter Gasston, Noviembre del 2014, The Book of CSS3: A Developer's Guide to the Future of Web Design, Estados Unidos: no starch press.

[37] Cory Janssen, User Agent (UA), disponible en <http://www.techopedia.com/definition/1614/user-agent-ua> al 20/07/2015.

[38] Malqui Diego, Noviembre 2014, <https://github.com/twbs/bootstrap/issues/15113>.

[39] Agosto 2013, Creating Accessible Forms, disponible en <http://Webaim.org/techniques/forms/> al 12/10/2014.

[40] Olga Carreras Montoto, Responsive Design y accesibilidad, disponible en <http://olgacarreras.blogspot.com.es/2007/02/wcag-20.html#000documentos> al 05/8/2014.

[41] Laura, Introduction to Google My Maps, disponible en <https://support.google.com/mymaps/answer/3024396?hl=en> al 12/10/1014

[42] Accessibility on Disqus, disponible en <https://help.disqus.com/customer/portal/articles/1424209-accessibility-on-disqus> al 9/12/2014.

[43] Olga Revilla Muñoz, Octubre del 2010, Web Content Accessibility Guidelines 2.0 made easy, España: Itakora Press.

[44] Octubre del 2013, W3C Web Content Accessibility Guidelines 2.0 Conformance Logos, disponible en <http://www.w3.org/WAI/WCAG2-Conformance> al 10/12/2014.

[45] Accessibility on Disqus, disponible en <https://help.disqus.com/customer/portal/articles/1424209-accessibility-on-disqus> al

10/11/2014.

[46] Matthew Magain, Agosto del 2009, Target Settles Accessibility Lawsuit for \$6 Million, disponible en <http://www.sitepoint.com/target-settles-accessibility-lawsuit-for-6-million/> al 8/08/2014.

[47] Jesús Hernández Galán, Junio de 2011, Accesibilidad Universal y Diseño para Todos. Arquitectura y Urbanismo, Artes Gráficas Palermo.

[48] Michael Cooper, Diciembre 2008, Web Content Accessibility Guidelines (WCAG) 2.0, disponible en <http://www.w3.org/TR/WCAG20/> al 01/09/2015.

[49] Jennifer Kyrnin, Avoid Inline Styles for CSS, disponible en <http://webdesign.about.com/od/css/a/aa073106.htm> al 02/09/2015

[50] Roger, Mayo del 2009, Refreshable Braille and the Web, disponible en <http://www.dingoaccess.com/accessibility/refreshable-braille-and-the-web/>.

[51] Loïc Martínez Normand, Junio 2015, Novática N° 232, Accesibilidad Web: Tendencias de futuro, Pag. 5. Disponible en: <http://www.ati.es/novatica/2015/232/Nv232-Presentacion.pdf>.