



Accesibilidad Web

WCAG 2.1 de forma sencilla

- › Aplica las **Pautas del W3C**
- › Evalúa con **WCAG-EM**
- › Crea documentos **PDF** accesibles
- › Implementa **ARIA**

Olga Revilla Muñoz
Olga Carreras Montoto

Prólogo de
Emmanuelle Gutiérrez y Restrepo

Accesibilidad Web

WCAG 2.1 de forma sencilla

Olga Revilla Muñoz

Olga Carreras Montoto

Prólogo de Emmanuelle Gutiérrez y Restrepo



Itákora Press

Accesibilidad Web. WCAG 2.1 de forma sencilla

Autoras: Olga Revilla Muñoz y Olga Carreras Montoto

Copyright © 2018 Itákora Press. Todos los derechos reservados.

Impreso en Madrid, España

Edición: Septiembre 2018

ISBN: 978-84-09-04489-4

Depósito legal: M-30483-2018

Este libro traduce, resume y cita contenidos de las *Web Content Accessibility Guidelines 2.1*. Copyright © [2018] *World Wide Web Consortium* (*Massachusetts Institute of Technology, European Research Consortium for Informatics and Mathematics, Keio University*), así como de la traducción de las WCAG 2.0 candidata a ser la oficial al español publicada el 15 de diciembre de 2009 coordinada por la Fundación Sidar - Acceso Universal.

Editora: Olga Revilla Muñoz – Itákora

Editora ejecutiva: Olga C. Santos Martín

Diseño editorial: Ana Matellanes García

Prólogo: Emmanuelle Gutiérrez y Restrepo

Imagen de mosaico: Patrón 3534214 de Geralt en Pixabay

Iconos de “Idea” por Ralf Schmitzer; “Video” de Alfa Design; y “Stop” de Márcio Duarte; todos ellos de *The Noun Project*

Logo de cafetera: Olga Revilla Muñoz

Revilla Muñoz, Olga; Carreras Montoto, Olga

Accesibilidad Web. WCAG 2.1 de forma sencilla. / Revilla Muñoz, Olga; Carreras Montoto, Olga — Madrid: Itákora Press, 2018. — 297 p.: 14 il.; 23 cm. — ISBN 978-84-09-04553-2

Todos los derechos reservados. Ninguna parte de este libro puede ser reproducida o transmitida por ningún medio o en ninguna forma (electrónica, mecánica, fotocopia, registro o cualquier otra), sin el permiso previo por escrito de la editora. Para solicitar información sobre derechos de reproducción o transmisión, contacte con info@itakora.com

Para mi cómplice, testigo, víctima y coartada de todo cuanto hago.

Olga Revilla

*Para Jesús, Silvia y Samuel,
al otro lado de “Usable y Accesible”*

Olga Carreras

Gracias 2.1

Olga Revilla
Itakora

El trabajo de miles de personas creando internet ha cambiado radicalmente la forma en la que accedemos a la información, contratamos bienes y servicios, nos relacionamos con otras personas, o realizamos trámites con las instituciones. Sin embargo, este cambio de paradigma conlleva nuevos retos —como la seguridad de los datos, la privacidad de las personas, la veracidad de la información o la igualdad de oportunidades—, que los profesionales que creamos internet debemos conocer y ser capaces de darles respuesta.

En concreto, la igualdad de oportunidades en el acceso a los recursos de internet es un reto que afecta a millones de personas. Características como velocidad de conexión, dispositivo de acceso, cobertura de red, nivel educativo, edad, origen social, ingresos, habilidades digitales, o discapacidad son sólo algunas de las variables que pueden llevar a una persona a la exclusión digital, y, por ende, al empobrecimiento personal y profesional.

Como profesionales, en nuestra mano está poner nuestro granito de arena para evitarlo. Una de las formas de hacerlo es la Accesibilidad Web, es decir, creando sitios web que no impidan o dificulten el acceso de las personas por alguna de las anteriores variables. La forma más popular de conseguirlo es seguir las Pautas de Accesibilidad para el Contenido Web (en inglés, *Web Content Accessibility Guidelines* – WCAG). Si las has leído, sabrás que son largas y complejas. Por ello, y por su necesidad de ser divulgadas e implementadas, decidí revisarlas y presentarlas de una manera más sencilla en forma de libro.

Este no es el primer libro de accesibilidad web que publico. En 2008 se publicaron las WCAG 2.0, y escribí “WCAG 2.0 de forma sencilla”. Diez años más tarde se publica una revisión y ampliación de las Pautas, respondiendo a un mayor uso del móvil, y a colectivos que no habían sido suficientemente considerados antes. De este modo, las WCAG 2.1 han crecido en tamaño y en complejidad, por lo que era necesario revisar mi anterior libro.

Esta vez, no he querido enfrentarme a este reto yo sola, y para ello he contado con la colaboración de Olga Carreras, la persona que, desde su blog *Usable y Accesible*, más trabaja por la divulgación de la accesibilidad web en lengua española. Sin su colaboración, conocimiento y experiencia, esta edición sin duda sería mucho más limitada.

Este libro es mucho más que una mera reedición con las nuevas pautas. Entre las novedades, un capítulo introductorio explicando la accesibilidad y responsabilidades de las diferentes profesiones digitales. También se ha sustituido el capítulo de las técnicas Flash y Silverlight por uno monográfico de documentos PDF, más extenso y con más ejemplos. Igualmente hemos dedicado un nuevo capítulo a la comprobación de accesibilidad siguiendo la metodología WCAG-EM, y otro dedicado a ARIA, con ejemplos prácticos de cómo aplicarlo. Y por supuesto, hemos incluido los nuevos criterios, técnicas y ejemplos de aplicación de la versión 2.1 de las Pautas.

Muchas gracias a todos los que habéis ayudado a que, tanto este libro como el anterior, vieran la luz, y muchas gracias a todos los que adquiristeis una copia de este, lo que ha permitido financiar esta segunda autoedición. Pero, sobre todo, muchas gracias a todos los que os dedicáis a mejorar la accesibilidad de cualquier producto o servicio, digital o físico: con vuestro trabajo, cada vez somos más las personas que podemos disfrutar de la información, aprovechar las oportunidades y sortear los retos que nos presenta la vida diaria.

Prefacio

Olga Carreras
Usable y Accesible

Hace 10 años, la transición de las WCAG 1.0 a las WCAG 2.0 fue complicada porque la nueva versión del estándar era muy diferente tanto en su planteamiento como en su estructura.

El blog de Olga Revilla, y posteriormente su libro, fue referente y tabla de salvación para muchos diseñadores, programadores o editores de contenido que debían aplicar las nuevas pautas. El libro les ayudó a comprender y digerir los nuevos conceptos y criterios, así como toda la documentación asociada a los mismos.

El paso de las WCAG 2.0 a las WCAG 2.1 será más sencillo, ya que los criterios que ya conocemos no cambian ni en orden ni en contenido. Eso sí, se añaden 17 criterios nuevos que en esta revisión del libro ayudamos a comprender y a evaluar.

En las auditorías que realizo desde *Usable y Accesible* detecto tres grandes problemas recurrentes que abordamos en nuevos capítulos específicos del libro.

En primer lugar, a pesar de que las WCAG ya no obligan a proporcionar alternativas accesibles para los documentos electrónicos que se adjuntan a un sitio web, sino que basta con hacerlos accesibles de forma nativa, vemos que hoy en día sigue siendo habitual excluirlos de las declaraciones de conformidad. Esperamos que el capítulo *Documentos PDF accesibles* ayude a concienciar y revertir esta situación.

En segundo lugar, uno de los principales problemas en los portales con elementos dinámicos (ventanas modales, mensajes de alerta, acordeones, menús desplegables, carruseles, etc.) es decir, la mayoría de los portales hoy en día, es el desconocimiento o la mala aplicación del estándar WAI-ARIA. El nuevo capítulo *ARIA el aliado (casi) desconocido* intenta introducir el estándar de manera sencilla, hacer hincapié en que su correcta aplicación es imprescindible para implementar portales accesibles para los usuarios de lector de pantalla.

En tercer lugar, es habitual que el compromiso y el esfuerzo empleado en publicar un producto accesible se diluya con el tiempo y su accesibilidad se vaya degradando poco a poco. Por eso es fundamental implicar a toda la organización, a todos los perfiles profesionales y establecer mecanismos para mantener la accesibilidad en el tiempo, implantando una política de Gestión de la Accesibilidad. Abordamos este tema en el nuevo capítulo *Introducción a la Accesibilidad Web*.

Por último, quiero agradecerle a Olga Revilla que me haya invitado a colaborar en su libro, ha sido un placer trabajar con ella en este proyecto que me entusiasmó desde el primer día, y que espero que os sirva de guía y de apoyo para que juntos consigamos que internet sea de verdad para todos.

Prólogo

Emmanuelle Gutiérrez y Restrepo
Directora de la Fundación Sidar – Acceso Universal

Al escribir el prólogo para “*WCAG 2.0 de forma sencilla*” sabía, y así lo predije, que no sería el único que habría de escribir para su autora, confirmando el merecido éxito y utilidad de ese trabajo pionero, tanto en su género como por su originalidad. Ocho años más tarde Olga Revilla se ha aliado con Olga Carreras para revisar esta obra con la última versión de las WCAG que acaba de publicarse, las ahora WCAG 2.1.

Nuevamente soy consciente de que los lectores de este libro probablemente querrán saltarse el prólogo e ir directamente al corazón de este, puesto que su objetivo es eminentemente práctico: resolver dudas y adquirir buenas prácticas. Sin duda los lectores lo conseguirán a través de las siguientes páginas.

Este libro resultará extremadamente útil a un amplio rango de “usuarios”, ya que sus contenidos cubrirán las expectativas y necesidades de personas dedicadas a muy diversos cometidos dentro del mundo de la creación de páginas y contenidos web.

En “*Accesibilidad Web. WCAG 2.1 de forma sencilla*” sus autoras describen y detallan no sólo los conceptos básicos para comprender las Pautas de Accesibilidad para el Contenido Web, sino que explican y recomiendan algunas prácticas que son vitales para desarrolladores, diseñadores y responsables de contenido.

Al igual que en la versión anterior de este libro, no se trata de una mera transcripción o traducción de las WCAG 2.1, ni pretende ser un sustituto de su lectura, la cual evidentemente se recomienda. Es más un resumen elaborado de técnicas que pueden ser aplicadas a cada pauta y criterio de éxito. Está escrito en un estilo y lenguaje que pueden ser comprendidos fácilmente por desarrolladores, arquitectos de información, diseñadores de interacción, y todos aquellos profesionales y especialistas que han emergido alrededor de la creación de contenidos y aplicaciones web.

Las autoras, Olga Revilla y Olga Carreras, no necesitan presentación para el público hispanohablante interesado en el diseño de interacción y en la accesibilidad. Con sus blogs “Itákora” y “Usable y accesible”, se han hecho un nombre en la consultoría de la Experiencia de Usuario, de la Usabilidad y de la Accesibilidad.

Las dos autoras son miembros destacados del Seminario SIDAR y por lo tanto estoy orgullosa de presentar y apoyar esta publicación, como un signo más del progreso y proyección de sus miembros. En ese espíritu del Seminario, en el de ayudarnos los unos a los otros, creo que está fundada la creación de esta obra y eso honra a sus autoras.

Estoy segura de que esta obra tendrá una amplia acogida y me congratula que así sea. No sólo porque les deseo el mayor de los éxitos a sus autoras sino porque será signo de que la accesibilidad ha calado en las filas de los implicados en la creación de contenidos, páginas y aplicaciones web en España y el mundo de habla hispana en general. Son ya más de veinte años los que llevo impulsando el concepto de accesibilidad en la llamada “Sociedad de la Información y del Conocimiento” y sé que el éxito de este libro será a su vez signo también de que van surgiendo frutos.

Contenidos

Introducción a la Accesibilidad Web	11
Las pautas WCAG 2.1.....	27
Requisitos de Conformidad.....	37
Evaluar la accesibilidad con WCAG-EM.....	45
Principio 1. Perceptible	53
Principio 2. Operable	104
Principio 3. Comprensible	143
Principio 4. Robusto.....	167
Documentos PDF accesibles	174
ARIA, el aliado (casi) desconocido.....	197
Recursos y herramientas	222
Resúmenes y esquemas	238
Glosario	258
Índice Global.....	261

Introducción a la Accesibilidad Web



Qué es la Accesibilidad Web

Toda persona que trabaje en el ámbito digital acabará teniendo un requisito por defecto en sus proyectos: “que sea accesible”, es decir, que las personas con discapacidad puedan percibir, comprender y operar con el sistema. Esta forma de entender la accesibilidad, como norma impuesta desde fuera, resta fuerza a su verdadero potencial.

En su nacimiento, las normativas de accesibilidad web ayudaban a que las personas con discapacidad pudieran acceder a los contenidos que se publicaban en internet. Estas primeras normas eran bastante duras (por ejemplo, no permitían usar JavaScript, ni Flash, ni imágenes en movimiento), por lo que generaron cierto rechazo en un incipiente medio de comunicación con ansias de libertad y creatividad sin límites.

Éramos muy creativos, pero nos mirábamos el ombligo pisoteando los derechos de las personas con discapacidad. Con el tiempo, habíamos levantado barreras que hacían que muchas personas, con o sin discapacidad, encontraran dificultades para interactuar con nuestros productos.

Afortunadamente, nos hemos dado cuenta de que estamos dejando pasar increíbles oportunidades al diseñar solo para personas que tienen nuestras mismas necesidades, que usan los mismos dispositivos y que se encuentran en el mismo entorno de uso que nosotros. Hemos sido capaces de levantar la vista, mirar a nuestro alrededor, y darnos cuenta de que podemos derribar las mismas barreras que hemos creado.

Fíjate en la Tabla 1. En ella se recogen problemas comunes de accesibilidad que crean situaciones discapacitantes¹. Estos problemas afectan a algunas personas de forma permanente o temporal, pero además los podemos sufrir todas las personas en situaciones corrientes. Por ejemplo, los problemas que experimenta una persona sorda para entender un vídeo en internet son los mismos problemas que experimenta un oyente con el mismo vídeo en un entorno ruidoso; o las dificultades que tiene una persona tetrapléjica para apuntar con un puntero en la pantalla son parecidas a las que tienen los ancianos al mover un ratón; o la interacción por voz que permite a las personas ciegas usar el móvil no es muy diferente a la que hacen los conductores de vehículos.

La discapacidad se ha revelado como una oportunidad de explorar cómo la tecnología crea y al mismo tiempo es capaz de resolver situaciones limitantes. Las soluciones a las que tienen derecho las personas con discapacidad (permanente o temporal) son las mismas que resuelven el problema situacional de muchas personas sin discapacidad. No obstante, el primer argumento debería ser más que suficiente para tener en cuenta la Accesibilidad Web.

¹ Recomendamos también el Toolkit de Microsoft sobre diseño inclusivo y situaciones discapacitantes <https://www.microsoft.com/design/inclusive/>

Tabla 1 La diversidad como fuente de inspiración

Problema	Permanente	Temporal	Situacional	Solución común
Pérdida de Visión	Ceguera	Pupila dilatada por un fondo de ojos, o cataratas	No mirar a la pantalla, como al conducir un coche	Facilitar interacción por voz
Baja visión	Glaucoma, o alta miopía	Conjuntivitis	Información legal en letra pequeña	Permitir hacer zoom en la pantalla
Ceguera al color	Daltonismo	Deslumbramiento por luces fuertes	Pantalla de móvil oscurecida al tener poca batería, o al darle luz excesiva	Contraste suficiente
Pérdida de Audición	Sordera	Otitis, o no entender bien el idioma	Entorno ruidoso, como un bar; o un entorno silencioso, como una biblioteca	Subtitulado
Baja Movilidad	Parálisis o pérdida de brazo, Parkinson, o artrosis	Esguince, tendinitis o túnel carpiano	Usar el móvil con una sola mano mientras la otra se usa para agarrarse al autobús	Puntos de interacción de tamaño suficiente
Dificultad para fijar y mantener la atención y concentración	Autismo, trastorno por déficit de atención con hiperactividad	Estados de ansiedad o depresión, dolor de cabeza	Realizar otra actividad, como cuidar niños mientras se mira el móvil	Simplicidad en la organización de los contenidos
Comprensión lectora	Dislexia, discapacidad cognitiva, o sordera de nacimiento	Bajo nivel educativo, bajo dominio del idioma	Entorno complejo, como una calle bulliciosa	Sencillez de los textos

Utilizando técnicas de diseño accesible logramos que un rango amplísimo de personas en situaciones dispares sea capaz de disfrutar de experiencias que van más allá de “leer” la web, como poder llenar una solicitud de empleo, reírse de un chiste que se cuenta en un video, o hacer una videoconferencia con la familia. Y todo ello sin importar si la persona que accede a nuestra experiencia tiene una discapacidad, es poco hábil, está en un entorno complicado, o se conecta a través de un dispositivo con una funcionalidad limitada.

Por eso, la accesibilidad es mucho más que un mero requisito, es la oportunidad de generar un producto de calidad que facilita la vida de todas las personas que lo usan. Piensa que un problema de accesibilidad no sólo es bloqueante para una persona con discapacidad, sino que también es una molestia para muchos usuarios, por lo que, si evitamos esa situación con técnicas de accesibilidad web, mejoraremos la experiencia para todos.

De esta manera, la accesibilidad web se puede definir como **el conjunto de características que debe incorporar un sitio web para que el mayor número posible de personas en el mayor número posible de circunstancias pueda acceder a él y usarlo**.

Estas características deben capacitar a cualquier usuario para:

1. Descubrir la existencia de la interfaz y de sus elementos.
2. Percibir y comprender los estados, las propiedades y los *affordances* (formas de interacción) de cada elemento de la interfaz.
3. Operar con la interfaz.
4. Finalmente, percibir y comprender el *feedback* de la operación

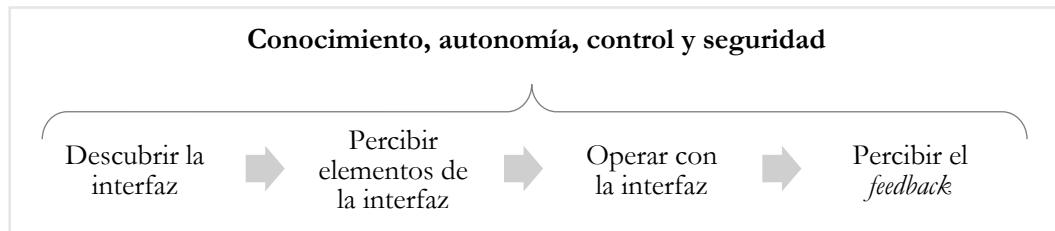


Ilustración 1 Las capacidades de la accesibilidad

Y todo ello permitiendo al usuario ser el dueño de su propio destino, es decir, permitiéndole que comprenda cada paso, y facilitando tomar sus decisiones con autonomía y de forma segura. Este proceso es continuo y secuencial, tantas veces con el usuario, a su voluntad, decida interactuar con el sitio web.



En el vídeo “Accesibilidad TIC. Secuencia de interacción” realizado por Alejandro Rodríguez-Ascaso para la UNED tienes un ejemplo práctico explicado.

<https://canal.uned.es/video/5a6f9224b111165148b459f>

La accesibilidad web es un derecho que tenemos todos a ser autónomos en internet, es decir, a ser capaces de acceder a los contenidos y funcionalidades independientemente de nuestra diversidad funcional (sensorial, motriz, intelectual o mental) o del contexto de uso (por ejemplo, las condiciones tecnológicas o ambientales).

Profesionales relacionados con la Accesibilidad Web

Implementar una web accesible es el día a día de muchos profesionales digitales². Cualquier cambio introducido, por pequeño que sea, puede suponer conseguir o perder la accesibilidad de un proceso o de un conjunto de páginas. Normalmente se hace hincapié en que los **creadores de contenidos, diseñadores y desarrolladores de front** deben conocer y aplicar las técnicas porque son quienes están en primera línea de batalla, en contacto día a día con el contenido, la interfaz y el código.

Pero sin duda **todos los perfiles implicados** en un sitio web influyen en el resultado final y deben tener, en mayor o menor grado, cierto conocimiento de lo que implican sus decisiones. La accesibilidad se consigue a través de una cadena de acciones que van desde el inicio del concepto hasta la entrega final al usuario:

- los **directores de proyectos** deben motivar y capacitar a su equipo, deben planificar y asignar recursos, así como contratar auditores externos llegado el caso;
- los **arquitectos de información** deben integrar una estructura y una navegación sencillas y con alternativas;
- los **investigadores de experiencia de usuario** deben incluir personas con discapacidad, así como situaciones limitantes, en sus *test* y en la aplicación de la técnica de *Personas* para garantizar la diversidad de los perfiles;
- los **especialistas en SEO** deben presionar para hacer páginas accesibles pues saben que los buscadores las adoran y consiguen posicionar mejor;
- los **diseñadores de servicio** deben pensar en situaciones extremas de uso de su idea de negocio para ampliar el *target* de venta y facilitar el uso para todos;
- los **consultores de analítica** pueden detectar problemas de accesibilidad mediante filtros en sus análisis (por ejemplo, filtrando por “edad” o “dispositivo” se pueden encontrar diferencias significativas en algún grupo de personas);
- los **creadores de videos y audios** deben tener en cuenta el subtitulado, los cambios intensos de luminosidad o el contraste del audio, entre otras cosas;
- los **auditores de calidad**, no sólo porque evalúan internamente el cumplimiento de las pautas, sino porque también deben incorporar en sus comprobaciones las herramientas que usan las personas con discapacidad y asegurar su correcto funcionamiento; o
- en el área de sistemas, los **desarrolladores back** y los **administradores de servidores y bases de datos** deben dar soporte en la configuración de servicios, la gestión de los datos o el desarrollo de las API que se necesiten en cada caso.

Pero no sólo los profesionales que están en contacto directo con el proyecto influyen en el resultado final. También el resto de la organización debe conocer su papel:

² En este libro se utiliza el masculino genérico para englobar a profesionales de todos los sexos. Las autoras de este libro en todo caso promueven y apoyan una mayor presencia de las mujeres en las diferentes profesiones digitales.

- los **responsables de compras** que deciden la contratación de herramientas informáticas de gestión de contenidos deben conocer las implicaciones de accesibilidad de ese software y qué barreras o facilidades supondrán para el equipo que lo va a usar;
- el **director de IT** debe establecer las políticas necesarias para planificar y gestionar la accesibilidad del sitio web a su cargo a largo plazo;
- el **director de recursos humanos** debe incorporar esta habilidad y sensibilidad en sus procesos de reclutamiento y en los planes de formación de empleados;
- los **directores del área legal y del área financiera** deben estar al tanto de posibles litigios y multas por el incumplimiento de la legislación;
- el **director de marketing, comunicación y relaciones públicas** probablemente quiera que la empresa tenga una buena imagen de cara a sus clientes y accionistas;
- el **director comercial** puede presentarse a convocatorias de proyectos donde se tengan en cuenta los procesos de calidad de la empresa, entre ellos la Accesibilidad Web; o
- en resumen, el **director general** debe ser quien lidere la filosofía de accesibilidad en toda la empresa, no solo en la web, sino en todos los aspectos de esta.

La accesibilidad es un trabajo de equipo, todos tenemos un papel que asumir: facilitando los medios adecuados, detectando problemas, integrando técnicas o revisando código. Delegar toda la responsabilidad en redactores, diseñadores y desarrolladores es un error común que debemos desterrar y empezar a responsabilizarnos de la parte de accesibilidad que nos toca.



El **W3C** ha establecido una serie de responsabilidades por cada perfil profesional.
https://www.w3.org/community/wai-engage/wiki/Accessibility_Responsibility_Breakdown

Ejemplo de la técnica *Personas* incluyendo perfiles con discapacidad para una academia *online*



SANDRA LÓPEZ

DEMOGRAFÍA

Mujer, 26 años

Soltera, vive con sus padres.

Reside en Badalona, pero trabaja en Barcelona (se desplaza en tren todos los días).

Trabaja como administrativa en una gestoría

Sueldo de 15.000€ brutos anuales

Estudios Secundarios (FP Administrativa)

DATOS RELEVANTES

Pérdida de audición del 100%

Sabe leer los labios y utiliza la lengua de signos

PERFIL TECNOLÓGICO

Usa el ordenador y el móvil perfectamente y con autonomía.

Tiene algunas aplicaciones adaptadas para utilizar subtítulos en los vídeos.

En su móvil usa una aplicación que transcribe los mensajes que le dejan en el contestador.

OBJETIVOS

Presentarse al examen de acceso de la universidad. Quiere estudiar Matemáticas. Para ello estudia en el tren.

RETOS

Comprender las teorías de matemáticas y física.

OBSTÁCULOS

Los libros con los que estudia son poco dinámicos y difíciles de seguir porque tienen un lenguaje muy complejo.

Los vídeos de YouTube que mira en el tren no tienen los subtítulos bien puestos y le cuesta seguir las explicaciones.

Qué debes saber si creas contenido, diseñas o desarrollas *front*

Una vez transmitida la necesidad de involucrar a todos los perfiles en la Accesibilidad Web, nos vamos a centrar en los creadores de contenidos, diseñadores y desarrolladores de *front*, que son los principales destinatarios de las técnicas de accesibilidad. Por ello, resumimos y explicamos los conceptos que les afectan en las siguientes páginas.

Esto no quiere decir que cada perfil deba ceñirse a esos conceptos, sino que debe revisar, al menos, lo que cae dentro de su ámbito de actuación y echar una mano al resto de profesionales. Además, en numerosas ocasiones comparten criterios, ya que la colaboración entre profesionales es necesaria para asegurar la accesibilidad del sitio.

Dependiendo del proyecto, existe una delgada y difusa línea que separa las responsabilidades de cada perfil: en algunas ocasiones, el diseñador entra en el contenido, el creador de contenidos toca el código, o el propio desarrollador crea contenido audiovisual mediante código. Del mismo modo, las tareas de un diseñador gráfico no son las mismas que las de un diseñador de interacción, un diseñador de interfaz y experiencia de usuario o un arquitecto de información.

A continuación, delimitamos las funciones de cada profesional, aunque luego en la realidad una persona puede asumir varios roles según los requisitos del proyecto:

- Cuando hablamos de **creadores de contenido**, nos referimos a las personas que crean textos, imágenes, vídeos, o audios y los introducen dentro de las páginas web, habitualmente a través de un gestor de contenidos.
- En cuanto a **diseñadores**, nos referimos a las personas que deciden el aspecto gráfico, la organización de los elementos y su interacción dentro del sitio web.
- Por último, los **desarrolladores** son los que tienen la responsabilidad de la maquetación HTML y CSS, con la implementación complementaria de ARIA, más el comportamiento en *JavaScript* (con JQuery, Angular, React, etc.).



En el capítulo *Resúmenes y Esquemas* tienes unos listados por perfil que puedes utilizar a modo de lista de comprobación.

Qué debes saber si eres creador de contenido

Los redactores de textos, los ilustradores, los grafistas, los fotógrafos, los editores de audio y video, así como las personas que manejan los gestores de contenidos, son los responsables de que los usuarios perciban y comprendan la información que quieren transmitir.

De la misma manera que los diseñadores deben actuar con claridad, los creadores de contenido deben buscar la **sencillez** en todas sus formas.

Para facilitar la **legibilidad** y comprensión de los textos deben estructurárselos en secciones precedidas de encabezados, claros y únicos en la página, y redactar con un **lenguaje sencillo**, explicando los tecnicismos o abreviaturas para que no haya dudas. Otros mecanismos que ayudan son el uso de listas, una breve introducción con el resumen del contenido, o la inclusión de versiones en “Lectura Fácil” o en audio.

Los textos, incluidas las instrucciones y ayudas, deben ser lo suficientemente explicativos e informativos, evitando aquellas instrucciones que se basan solo en características sensoriales como el color, la forma o la ubicación del componente.

La claridad es especialmente importante en la redacción del **título** de la página y de cada **enlace** de esta; en ambos casos deben transmitir sin lugar a duda el contenido que nos vamos a encontrar. Los enlaces deben ser consistentes, de tal manera que dos enlaces iguales naveguen siempre al mismo recurso; y el título de la página, corto, conciso y único en el portal.

Aparte, siempre hay que indicar el **idioma** de la página y de los contenidos para que los lectores de pantalla puedan pronunciarlos adecuadamente, y los traductores automáticos puedan hacer su trabajo de forma más eficiente.

Si en la página web se utilizan **recursos audiovisuales** como vídeos y audios, los creadores deben proporcionar alternativas para las personas que no pueden percibirlos en su forma original: por ejemplo, subtítulos, lengua de signos o una transcripción textual del contenido. Estos contenidos no deben comenzar a reproducirse por defecto y, en el caso del sonido, debe haber suficiente contraste entre la pista principal y el sonido de fondo para que se entienda bien.

Los creadores de contenidos deben saber que ciertos destellos y movimientos pueden causar **epilepsia y mareos**, por lo que deben tener precaución al hacerlos y comprobarlos antes de incluirlos.

En cuanto a las **imágenes**, debe incluirse una descripción concisa que transmita la misma información. Hay que evitar las imágenes de texto y debe tenerse en cuenta el contraste de colores y no transmitir información solo por el color.

Es fundamental que los editores de contenido usen correctamente las **opciones que les ofrece el gestor**: que no peguen contenido directamente desde Word, o que no simulen elementos, sino que incluyan los encabezados, listas o tablas con las opciones del mismo. El gestor permitirá en muchos casos marcar también las citas, las abreviaturas, las celdas de encabezado, incluir resúmenes en las tablas, etc.

Por último, como responsables de los **documentos** que se adjuntan al portal (PDF, documentos de texto, hojas de cálculo, etc.) deben saber que estos también deben ser accesibles, a lo cual dedicamos un capítulo entero en el libro.

Qué debes saber si eres diseñador

En primer lugar, probablemente lo más conocido por todos los diseñadores, es que deben **asegurar el contraste** entre el color del texto y del fondo para garantizar que se pueda leer. Deben aplicar también esta revisión a los elementos no textuales que transmiten información, como gráficas e iconos, así como al borde de los campos de formulario o al indicador del foco si se modifica su diseño.

El color puede usarse junto con otros recursos (tamaños, fuentes, textos, iconos, etc.) para **crear elementos fáciles de identificar**. Hay que pensar en las personas que no pueden distinguir los colores: si se utiliza solo el color para transmitir información, especialmente en gráficas, enlaces o calendarios, no van a poder comprenderlos.

El color no es el único aspecto de diseño para **facilitar la lectura** de los textos, también hay otros requisitos o recomendaciones relacionados con el alineamiento, los márgenes, el tipo de fuente o su tamaño, como no justificar los textos o que la separación de los párrafos sea mayor que el interlineado.

Por otro lado, una de las máximas que deben cumplir los diseñadores es la **consistencia**. Deben crear una estructura y un sistema de navegación consistente a lo largo de todo el sitio. También debemos informar al usuario de dónde se encuentra y adónde puede ir, proporcionándole varias vías para llegar al mismo contenido (a través de menús, enlaces relacionados, un mapa web, un buscador, etc.).

Otra máxima es la **claridad**, en caso de duda, elegir siempre la opción más sencilla. La claridad es fundamental por ejemplo en los enlaces, como ya comentamos en el apartado dedicado a los creadores de contenido, el usuario no puede tener ninguna duda de a dónde le llevarán, por eso, y por consistencia, dos enlaces iguales siempre deben tener el mismo destino. Aunque los desarrolladores podrán clarificar el destino después, es importante tenerlo ya en cuenta desde la definición de la arquitectura de información.

En cuanto a la **organización y presentación de los contenidos**, además de ser consistente a lo largo de las páginas, el diseñador debe utilizar los encabezados para dividir las diferentes partes de la página y del contenido en bloques coherentes. Aunque nos hayamos esforzado en hacer un diseño claro, hay que tener en cuenta que el usuario podrá personalizarlo con las opciones de los navegadores, con extensiones o con productos de apoyo, para adaptarlo a sus necesidades. Del mismo modo, nosotros también podemos incluir opciones de personalización en el sitio, como controles para aumentar el tamaño del texto o cambiar a una versión de alto contraste.

La claridad también debe ser una máxima en el **diseño de los formularios** en todas sus fases y estados (sin enviar, enviado, con errores, etc.). Por un lado, las etiquetas de los campos y las instrucciones deben facilitar la introducción de la información y prevenir que el usuario se equivoque. En el caso de que se equivoque, identificar claramente los errores y ofrecerle sugerencias para que se recupere del error.

La variedad de **tamaños** de pantallas es enorme, cosa que los diseñadores deben tener en cuenta diseñando las versiones de la página en diferentes resoluciones de pantalla. Deberán además respetar un tamaño mínimo recomendado para las áreas interactivas de 44 por 44 píxeles.

En diseño también se utilizan imágenes, vídeos, audios, animaciones, destellos, transiciones y efectos para enriquecer el mensaje. Los **recursos audiovisuales** deben tener una alternativa para las personas que no pueden percibirlos en su forma original, y, además, debe existir una forma que permita a los usuarios controlar su reproducción, por lo que hay que diseñar los mecanismos necesarios para ello.

Los diseñadores deben saber, al igual que los creadores de contenidos, que ciertos tipos de destellos y de movimientos pueden causar **epilepsia y mareos**, por lo que deben tener precaución al hacerlos y comprobarlos antes de incluirlos.

Por otro lado, al diseñar la interacción, hay que tener en cuenta que el usuario debe poder confirmar o revisar sus acciones y **no importunarle** con interrupciones inesperadas o pidiéndole que se vuelva a autenticar una y otra vez en el sistema. Se debe permitir que tomen **decisiones de forma autónoma**, como decidir cuándo se envía un formulario o cuándo se cambia de una página web a otra.

A nivel de interacción, los diseñadores deben asegurar que todos los usuarios puedan hacer un **manejo sencillo** de la página web, en especial aquellas personas que tienen dificultades para manejar el ratón o el móvil. Aunque los desarrolladores asegurarán la interacción por teclado, si se diseñan interacciones táctiles o de ratón complejas (*drag & drop*, gestos con varios dedos, etc.), habrá que prever que se deben ofrecer alternativas de teclado y con gestos sencillos y/o con opciones de configuración.

Por último, otra forma de facilitar la vida a los usuarios menos rápidos o hábiles es **gestionar el tiempo** que tienen para realizar las interacciones sin que ello les agobie, evitando así que tomen malas decisiones por prisas innecesarias.

Qué debes saber si eres desarrollador

Como hemos indicado previamente, en este apartado incluimos los perfiles responsables del código de las páginas, como maquetar o definir el comportamiento.

Su principal mandato es **seguir correctamente los estándares** del lenguaje de desarrollo que elija, creando código válido que permita a los navegadores y ayudas técnicas interpretarlo fielmente. Este mandato incluye no sólo que el código valide en los validadores automáticos, sino también que sea **semánticamente** adecuado a la información, relaciones, secuencias y controles que genera.

Para ello deberá usar controles estándar siempre que se pueda o valerse del estándar **WAI-ARIA**, al que dedicamos un capítulo entero. Este estándar permite añadir información semántica a los elementos de la página, y ayuda a hacer accesibles los elementos dinámicos (alertas, menús desplegables, carruseles, etc.).

Los desarrolladores deben ayudar a los usuarios a comprender la interfaz mediante la correcta **identificación de los controles y su propósito**. Esto es especialmente importante en el caso de los enlaces y de los campos de formulario, tradicionalmente complicados para los usuarios de productos de apoyo. Los desarrolladores deben ayudar a los usuarios a llenar los **formularios** mediante una identificación clara y consistente de los campos, las ayudas e instrucciones, y los errores, de tal manera que sean perceptibles para todos los usuarios.

Aunque es muy habitual ya el uso de archivos HTML, CSS y JS para la estructura, la presentación y la funcionalidad respectivamente, los diseñadores deben **separar el contenido de la presentación**, no sólo en la maquetación general de las páginas, sino también en las imágenes, evitando las imágenes de texto o definiendo las imágenes decorativas en la hoja CSS.

Es una responsabilidad compartida con el creador de contenidos que muchas de las imágenes del portal tengan una alternativa adecuada. Por otro parte, deberá seleccionar un **reproductor multimedia accesible** que admita las diferentes alternativas dispuestas para cada contenido de video o audio.

El desarrollador debe asegurar un **orden de lectura** adecuado en cualquier contexto, siendo una buena práctica que el orden visual del contenido coincida con su orden en el código o en DOM.

Además, tanto los campos como los enlaces y otros elementos de interacción tienen que estar disponibles para ser usados por **teclado** en un orden lógico, identificando en todo momento dónde está el foco y sin que este quede atrapado en un componente. Si se implementan **atajos de teclado** debe asegurarse de que no entren en conflicto con los del navegador y que no estén asociados a una única tecla.

Por otro lado, debe garantizar que el usuario obtenga el **feedback** necesario de las consecuencias de sus acciones, y tenga el **control** en todo momento, evitando comportamientos inesperados o no solicitados como el salto del foco o la navegación a otra página sin haberlo solicitado.

También depende de la implementación de los desarrolladores que la página se visualice correctamente en **diferentes dispositivos, configuraciones y tamaños de pantalla**, que se adapte al girar la pantalla, que los elementos se reajusten sin necesidad de hacer scroll horizontal o vertical, que el texto pueda ampliarse o que sea posible hacer zoom sin problemas.

Otra forma de facilitar la vida a los usuarios es **evitarles los bloques** de navegación repetitivos entre páginas, disponiendo mecanismos que les permitan saltarlos; evitarles el refresco automático de la página; o la pérdida de datos si expira la sesión.

Por último, al igual que los creadores de contenidos, deben indicar el **idioma** de la página y de sus diferentes contenidos, para que los lectores de pantalla puedan entonar correctamente. También pueden ser responsables del **título** de las páginas o el patrón a partir del cual se generan, deberán entonces tener en cuenta que tiene que ser corto, conciso y único en el portal e identificar claramente no solo la página sino también el sitio.

Implantar y gestionar la política de Accesibilidad Web de una organización

Integrar la accesibilidad web dentro de los procesos diarios de una web puede ser complicado sin el apoyo y promoción de la dirección de la compañía. La legislación y la responsabilidad social corporativa han empujado a las instituciones a implantar la accesibilidad en sus proyectos digitales, tanto web como apps móviles.

Esto implica la asignación de una serie de recursos, el establecimiento de actividades y la implicación activa de todos los que toman decisiones que afectan al producto digital.

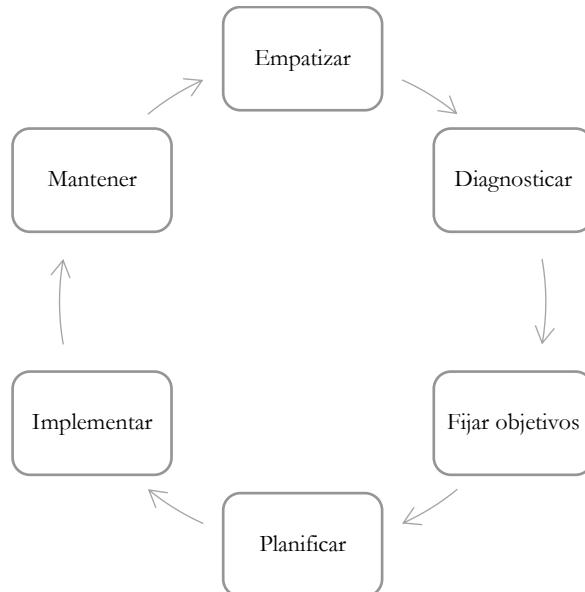


Ilustración 2 Ciclo de implementación y gestión de la Accesibilidad Web en una organización

En primer lugar, la organización debe **tomar conciencia** de las situaciones capacitantes y discapacitantes que ofrecen sus productos digitales, bien a través de demostraciones, formación, o viviéndolo en su propia piel. En el capítulo *Recursos y herramientas* tienes algunos simuladores de discapacidades para que puedas experimentar cómo perciben, comprenden y operan tu web muchos usuarios.

La empatía con el impacto directo en las personas con discapacidad no es el único motivo para implantar la accesibilidad web:

- Aumentar la facilidad de uso para todos, mejorando los índices de satisfacción y de experiencia de uso.
- Al aumentar la facilidad de uso, el *target* se expande, con la oportunidad de alcanzar un mayor público, ventas y beneficios.
- Mejorar en el posicionamiento en buscadores.

- Reducir los costes financieros, de tiempo y de recursos que aumentan a medida que se retrasa abordar la accesibilidad de sus productos.
- Evitar sanciones económicas y administrativas.
- Crear impacto positivo en la marca y en la responsabilidad social corporativa.

El siguiente paso es **diagnosticar** en qué estado se encuentra la organización en relación con la accesibilidad web para poder dimensionar el trabajo que queda por delante.

- ¿Cuáles son los activos digitales de la organización?
- ¿Quiénes se ocupan de estos activos? ¿Qué formación en accesibilidad tienen?
- ¿Qué políticas y procesos hay implantados actualmente?
- ¿Qué leyes de accesibilidad web le aplican? ¿Tiene fijada la organización un nivel mínimo que debe alcanzar?
- ¿Ha realizado alguna auditoría previa?

A partir de aquí podemos empezar a trabajar en construir, fijándonos **objetivos** específicos, medibles y alcanzables en un plazo de tiempo determinado. Probablemente la legislación obligue a alcanzar el nivel AA de las WCAG 2.1 (en el próximo capítulo las explicaremos) en una determinada fecha. Este es un buen objetivo para comenzar a trabajar. Este objetivo debe ser compartido por todas las personas que integran la organización, para poder remar todos en la misma dirección.

Dependiendo del diagnóstico previo y del objetivo que nos fijemos, es necesario trazar un **plan de trabajo** u otro atendiendo las necesidades y características de cada perfil profesional implicado. Hay que tener en cuenta que se debe agendar tiempo y dinero para, como mínimo, la formación del equipo, la revisión del sitio web y la implementación de los cambios necesarios. Además, debe haber personas concretas y comprometidas que se responsabilicen de cada tarea. Y todo ello, dentro de un marco temporal concreto.

El plan de trabajo podría incluir las siguientes tareas:

- **Documentar los procesos y flujos** de publicación de contenidos y de mantenimiento del sitio.
- **Identificar los puntos de control de la accesibilidad necesarios** dentro de estos procesos y flujos.
- **Definir un plan de revisiones**, con unos responsables claros, donde se especifique la frecuencia, el método y el alcance de las revisiones.
- **Formar y transmitir** el conocimiento a las personas implicadas en la publicación y el mantenimiento del sitio.
- **Incluir el feedback de los usuarios** mediante un mecanismo de contacto específico que permita recoger las quejas y sugerencias de los usuarios. El análisis y seguimiento de este *feedback* ayuda a definir acciones correctoras, a adaptar los sitios y aplicaciones web a las necesidades reales de los usuarios, y a la mejora continua.

Una vez que cada perfil tiene claro qué debe hacer, es el momento de **implementar** los cambios necesarios. Es muy recomendable hacer cambios progresivos y evaluar continuamente para tener una mayor sensación de progreso dentro de la búsqueda del objetivo común, así como asegurar su correcta implementación. En el capítulo *Evaluuar la accesibilidad con WCAG-EM* encontrarás una forma fiable de auditar la accesibilidad de la web de la organización.

Una vez que consigamos nuestro objetivo, es el momento de no bajar la guardia. Los sitios web son entes en continuo cambio y cualquiera de ellos puede suponer levantar de nuevo una barrera de uso. Es necesario seguir trabajando no sólo para **mantener** el objetivo perseguido, sino para alcanzar nuevos logros. Para ello, sigue siendo necesario empatizar, formarse, estar al tanto de la nueva legislación, evaluar la web, implementar mejoras y renovar la política de accesibilidad de la organización.

Cabe señalar que los pasos no tienen por qué ser secuenciales, sino que pueden darse situaciones de solapamientos: por ejemplo, puede existir un trabajo diario de implementación de mejoras antes de haber definido objetivos; o necesitar de una fuerte fase de empatía después de haber fijado unos objetivos concretos para convencer a los que tienen que implementar los cambios. La idea es tener el mayor control de lo que sucede, pero ser lo suficientemente flexible para que, si la situación lo pide, se pueda gestionar de la forma más adecuada.



El **W3C** ha elaborado una completa guía con consejos y recursos para ayudar a las organizaciones a planificar y gestionar su accesibilidad digital.

<http://www.w3.org/WAI/planning-and-managing/>

Las pautas WCAG 2.1



Qué son las Pautas WCAG

Las Pautas de Accesibilidad para el Contenido Web (WCAG por su nombre en inglés, *Web Content Accessibility Guidelines*) disponen de una serie de recomendaciones con el objetivo de hacer el contenido web más accesible, especialmente a personas con discapacidad.

Publicadas por el W3C, la principal organización mundial de estándares de internet, en la actualidad las WCAG son las pautas más reconocidas, seguidas y exigidas a nivel internacional en lo relativo a la accesibilidad web, hasta el punto de integrarse en la mayoría de las legislaciones nacionales y regionales³.

Las WCAG nacieron en su versión 1.0 en 1999. En 2008 el W3C publicó las WCAG 2.0, una evolución de las anteriores que habían quedado obsoletas debido a los avances tecnológicos. Diez años después, en 2018, el W3C ha aprobado una nueva evolución de dichas pautas, respondiendo a una mayor diversidad de tecnologías (por ejemplo, las pantallas táctiles de los móviles), o dando soporte a colectivos con discapacidad cognitiva, con baja visión o con dificultades para el aprendizaje. Sin embargo, las pautas WCAG 2.1 no dan una solución completa a todos los retos que suponen tanto las nuevas tecnologías como los colectivos antes mencionados, por lo que se plantean como una transición a las futuras WCAG 3.0.

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

En este libro hemos simplificado la organización y los textos de los cuatro documentos. Para llevar a cabo una auditoría, referencia siempre la documentación original.

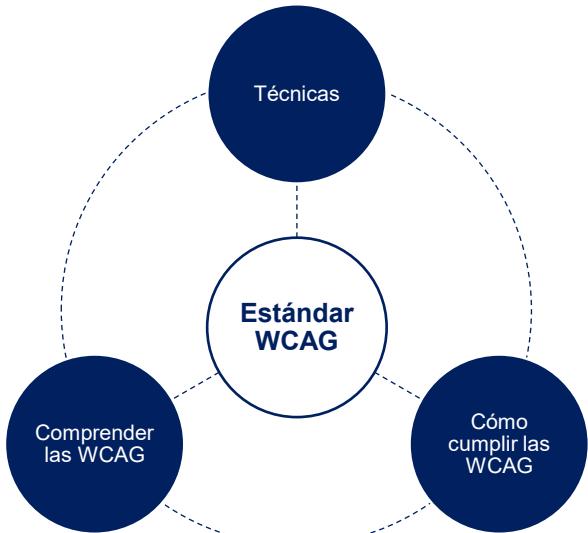


Ilustración 3 Documentos de las WCAG

³ El W3C mantiene un listado actualizado de legislaciones de diferentes países que hacen referencia a las WCAG. <https://www.w3.org/WAI/policies/>

Qué novedades introducen las WCAG 2.1 con respecto de las WCAG 2.0

Las pautas WCAG 2.1 incorporan 17 criterios nuevos y una pauta, que se suman a los que ya existían. La nueva pauta (la 2.5) agrupa los nuevos criterios relacionados con formas de introducir información en la interfaz. En concreto, los nuevos criterios y sus niveles son:

- 1.3.4 Orientación de la pantalla (AA)
- 1.3.5 Identificación del propósito del campo (AA)
- 1.3.6 Identificación del propósito (AAA)
- 1.4.10 Reajuste de elementos (AA)
- 1.4.11 Contraste no textual (AA)
- 1.4.12 Espaciado del texto (AA)
- 1.4.13 Contenido en *hover* o *focus* (AA)
- 2.1.4 Atajos de teclado (A)
- 2.2.6 Límites de tiempo (AAA)
- 2.3.3 Animaciones desde interacciones (AAA)
- 2.5.1 Gestos del puntero (A)
- 2.5.2 Cancelación del puntero (A)
- 2.5.3 Etiqueta en el nombre (A)
- 2.5.4 Actuación por movimiento (A)
- 2.5.5 Tamaño del área de interacción (AAA)
- 2.5.6 Mecanismos de entrada concurrentes (AAA)
- 4.1.3 Mensajes de estado (AA)

Para el criterio 1.3.3 de las WCAG 2.0, que decía que “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”, se añade también la característica del color. En realidad, es más una matización que una novedad, ya que el criterio 1.4.1 ya introducía el uso del color con esas mismas restricciones.

Se introducen también matizaciones relacionadas con la conformidad, como la inclusión en las declaraciones de conformidad de las diferentes versiones de la página según el tamaño de pantalla, y de nuevos elementos opcionales.

Además, las pautas incluyen nuevas definiciones en el glosario. En el libro hemos introducido los conceptos dentro de su contexto.

Cómo se organizan las WCAG 2.1

Las WCAG 2.1 están organizadas en cuatro capas, cuya complejidad crece exponencialmente.

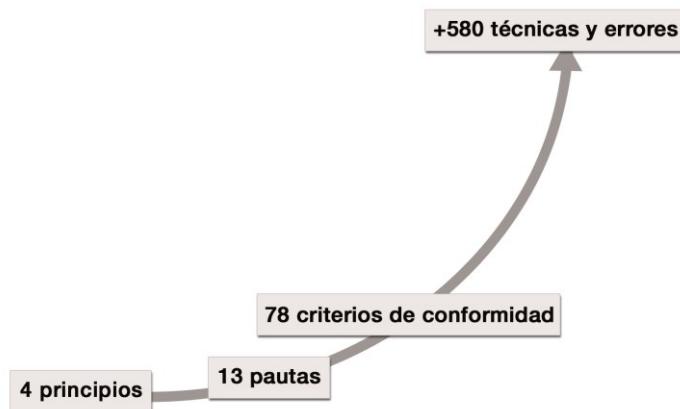


Ilustración 4 Organización de las WCAG 2.1

4 principios

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

13 pautas (1 nueva)

Los principios se dividen en pautas, con el objetivo de agrupar de un modo lógico los criterios de conformidad. Las pautas no son evaluables.

78 criterios de conformidad (17 nuevos)

Los criterios de conformidad sí son evaluables. Dependiendo de una serie de factores que veremos más adelante, los criterios de conformidad se escalan en 3 niveles: A (el más bajo), AA (el nivel medio), y AAA (el más alto).

Más de 580 técnicas y errores

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

Los cuatro principios y sus pautas

Como decíamos antes, las WCAG 2.1 se organizan en principios, y estos en pautas. El objetivo es superar los cuatro principios, para lo cual debemos cumplir todas sus pautas.

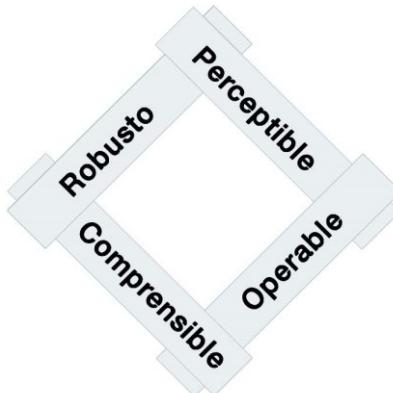


Ilustración 5 Los cuatro principios

El sitio web debe ser “Perceptible”

Nuestro sitio web puede ser visitado por personas con diferentes tipos de preferencias y capacidades, pero también por robots (buscadores, traductores...). Nuestra información e interfaz deben tener en cuenta esta necesidad y por ello debemos ofrecer alternativas a los usuarios que no pueden utilizar alguno de sus sentidos. Las pautas de este principio son:

Pauta 1.1 Alternativas textuales a contenidos no textuales

Pauta 1.2 Alternativas a medios tempodependientes, es decir, vídeo y audio

Pauta 1.3 El contenido es adaptable a diferentes formas de presentación

Pauta 1.4 Distinguible: El contenido es fácil de ver y escuchar

El sitio web debe ser “Operable”

Los diseñadores y programadores debemos proporcionar elementos de interacción y de navegación que puedan ser manejados por personas con diferentes capacidades. Las pautas de este principio son:

Pauta 2.1 Funcionalidad accesible por teclado

Pauta 2.2 Los usuarios tienen tiempo suficiente para leer y usar el contenido

Pauta 2.3 El contenido no causa convulsiones

Pauta 2.4 Navegable: Los usuarios pueden navegar, encontrar contenido y saber dónde están en todo momento

Pauta 2.5 Facilita formas de introducir información [Nueva 2.1]

El sitio web debe ser “Comprensible”

Si nuestro usuario no entiende de qué le estamos hablando, o le hacemos sentirse perdido, tenemos un problema. Debemos diseñar nuestro sitio web -incluyendo la información y la interfaz de usuario- fácil de usar. Las pautas de este principio son:

Pauta 3.1 El contenido es fácil de leer y de comprender

Pauta 3.2 El contenido aparece y se maneja de una forma predecible

Pauta 3.3 Ayuda en la introducción de datos para evitar y corregir errores

El sitio web debe ser “Robusto”

Este es el principio más dependiente de la tecnología. Se basa en la capacidad del sitio web de ser trasmítido e interpretado por los diferentes agentes de usuario.

Los **agentes de usuario** son todos aquellos programas que muestran contenidos de internet, como los navegadores (Internet Explorer, Firefox, Safari...), reproductores multimedia (QuickTime, Real Player, Windows Media Player...), *plug-ins* (Flash, Silverlight, Java...), y otros programas y dispositivos como los productos de apoyo.

Los **productos de apoyo**, también conocidos como *ayudas técnicas*, son programas o dispositivos que proporcionan la funcionalidad necesaria para cubrir las necesidades de los usuarios con discapacidad, más allá de la que proporcionan las aplicaciones de usuario principales. Son productos de apoyo, por ejemplo, un magnificador de pantalla, un teclado en pantalla, un puntero grande...

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

La única pauta de este principio es:

Pauta 4.1 El contenido es compatible con las herramientas de usuario actuales y futuras

Los criterios de conformidad

Como decíamos anteriormente, las pautas engloban los criterios de conformidad, los cuales se clasifican por su parte en 3 niveles:

- **A** (o *Simple-A*): el más bajo.
- **AA** (o *Doble-A*): nivel medio.
- **AAA** (o *Triple-A*): el más alto.

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

- ¿Este criterio de conformidad es esencial? Es decir, ¿si no se cumple este criterio, los agentes de usuario (incluyendo los productos de apoyo) pueden o no reproducir la página web?
- ¿Este criterio de conformidad es aplicable para todos los sitios web, independientemente de su tema, del tipo de contenido, de la tecnología usada, etc.?
- ¿Los creadores de contenidos pueden aprender en menos de una semana a cumplir este criterio de conformidad?
- ¿Cómo limitaría este criterio de conformidad la apariencia, la funcionalidad, la forma de presentación, la libertad de expresión, el diseño y la estética de la página?
- ¿Hay una alternativa u otro camino para acceder al contenido si no se cumple el criterio de conformidad?

Todos los criterios de conformidad son importantes: los más bajos aseguran que se pueda acceder al contenido de las páginas web y los más altos ayudan a mejorar su usabilidad y a llegar a un mayor número de personas.

Cuando se evalúan los sitios web, se repasan los criterios y se comprueba si se cumplen o no. En el capítulo siguiente se explica la forma de hacerlo.

Las técnicas y los errores

Cada criterio de conformidad recomienda una serie de **técnicas**, las cuales disponen de procedimientos de prueba que nos van a permitir saber si el criterio se cumple o no.

Observa que hemos utilizado el verbo “recomendar”, porque ninguna técnica es obligatoria, ya que las WCAG 2.1 permiten crear tus propios mecanismos alternativos para superar el criterio. Es decir, que aparte de las técnicas documentadas por el W3C, éste admite que pueda haber otras formas de cumplir los criterios de conformidad, que serán aceptadas siempre y cuando:

- Permitan alcanzar el criterio de conformidad;
- Cumplan -o no hagan fallar- el resto de los criterios.

Hay **2 categorías de técnicas**: las *suficientes* y las *aconsejables*. Las primeras ayudan a superar o cumplir el criterio de conformidad. Las segundas son recomendaciones para mejorar la accesibilidad y usabilidad del sitio web.

Las técnicas pueden ser generales o referirse a una tecnología concreta (HTML, CSS, SMIL, Flash, Silverlight, ARIA, PDF...)⁴.

Los **errores** documentados por las WCAG 2.1 son prácticas que dificultan o impiden la accesibilidad de la página. Al igual que las técnicas, disponen de procedimientos de prueba para verificar que no estamos tropezando de nuevo con la misma piedra.

Tanto las técnicas como los errores pueden estar asociados a más de un criterio de conformidad, por lo que podemos encontrarnos técnicas y errores usados en más de una ocasión.

⁴ Las técnicas de Flash y Silverlight, por el desuso de estas tecnologías, no se mencionan en este libro. Sin embargo, las técnicas de PDF y ARIA, muy utilizadas, se explican en sendos capítulos aparte. El resto de las técnicas se explican integradas dentro de cada criterio.

¿A qué tecnologías se aplican las WCAG 2.1?

Las WCAG 2.1 son neutrales con respecto de la tecnología, es decir, para construir un sitio web se puede usar cualquier tecnología. Existen dos conceptos relacionados:

Una tecnología es **compatible con la accesibilidad** (en inglés, *accessibility supported technology*) 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 esa tecnología. La clave es cómo las tecnologías incluyen los mecanismos necesarios para que los agentes de usuario y los productos de apoyo puedan entenderlas y manejarlas, es decir, puedan acceder al contenido, presentarlo al usuario, y éste, percibirlo y manipularlo. Incluso es posible que sea necesario modificar los agentes de usuario o los productos de apoyo para dar soporte a esa tecnología, por ejemplo, descargando un *plug-in*.

La **dependencia tecnológica** (en inglés, *technologies that are relied upon*) hace referencia a que, cuando una página web utiliza una determinada tecnología para mostrar el contenido, si el agente de usuario no reconoce o no soporta esa tecnología, o si está desactivada, el contenido dejará de ser conforme. Por ejemplo, una web en Flash no se puede visualizar en un iPad, porque estos dispositivos no lo permiten; sin embargo, sí se visualizará en un navegador de sobremesa con el *plug-in* en el navegador. Esto no es obstáculo para que Flash pueda ser calificada como una tecnología compatible con la accesibilidad, ya que ofrece opciones para hacer un uso accesible de sus contenidos (aunque sea bajo determinadas condiciones). Sin embargo, genera una dependencia tecnológica.

Por ejemplo, puedes usar HTML mal formateado, y será considerado “no accesible”; pero puedes crear tu sitio web con Flash utilizando correctamente todas sus opciones de accesibilidad y será considerado accesible.

Aunque el W3C adopta una postura neutral con respecto a la tecnología, no lo hace sobre su uso. **El modo en que usamos la tecnología determina si la tecnología es o no accesible.** Por lo tanto, no hay una lista oficial de tecnologías compatibles con la accesibilidad, pero por ejemplo PDF o JavaScript se considera que lo son. Usar estas tecnologías de “forma compatible con la accesibilidad” es hacerlo de forma que puedan ser utilizadas sin problemas por los productos de apoyo, como un lector de pantalla. En el capítulo “WAI-ARIA” se explica cómo hacerlo en el caso del JavaScript, y en el capítulo de “Documento accesibles” cómo hacerlo en el caso de los PDF.

Si usamos tecnologías no compatibles con la accesibilidad, o usamos tecnologías compatibles con la accesibilidad pero las usamos de una forma no compatible, deberemos proporcionar una versión alternativa. Por ejemplo, si incluimos un PDF no accesible, deberemos proporcionar una alternativa accesible al mismo, por ejemplo, en HTML accesible. Pero si nuestro PDF ya es accesible de forma nativa, no necesitamos incluir una versión en HTML accesible.

Por qué seguir las WCAG 2.1

- Porque son la recomendación oficial del **W3C**, la organización de desarrollo de estándares web más reconocida del mundo.
- Porque son el fruto de muchos **años de trabajo** y del consenso de una comunidad internacional de especialistas en accesibilidad.
- Porque la versión 2.0 ha sido **reconocida** por las principales organizaciones de estandarización, como la ISO a nivel internacional, el CEN/ETSI a nivel europeo y AENOR a nivel nacional en España, creando los estándares: ISO 40500, EN 301 549 y UNE 139803:2012, respectivamente; y se espera que hagan lo mismo con las 2.1. De hecho, la EN 301 549 ya se ha adaptado a las WCAG 2.1.
- Porque las WCAG 2.0 son **legalmente obligatorias** en muchos países, como en EE.UU. o en los estados miembro de la Unión Europea, y se espera que de igual manera adopten las WCAG 2.1, por tanto, sería buena que idea que fueras teniéndolas ya en cuenta. Por ejemplo, en Europa son obligación legal para los sitios web y aplicaciones móviles de los organismos del sector público desde la aprobación de la Directiva 2016/2102. En España, la legislación en materia de accesibilidad web⁵ hace referencia a la norma UNE 139803:2012 (equivalente a las WCAG 2.0) y a la EN 301 549 (adaptada a las WCAG 2.1), y además obliga no solo a los portales de la administración pública sino también a muchos portales de empresas privadas.
- Porque son **sumatorias**: cumpliendo con las WCAG 2.1 también se cumple con las WCAG 2.0, dado que es una actualización que suma, no cambia los anteriores criterios o técnicas.
- Porque se puede comprobar su cumplimiento. Incluso cuentan con su propia **Metodología de Evaluación** oficial, perfectamente aplicable también con las WCAG 2.1, la cual se explica en el capítulo “Evaluar la accesibilidad”.
- Porque existen **validadores** automáticos y herramientas que facilitan la evaluación. Nos servirán igualmente los validadores de las WCAG 2.0, pero pronto veremos que comienzan a actualizarse a las WCAG 2.1.
- Porque sirven para **todas las tecnologías web**, y están pensadas para no quedarse obsoletas. En Europa, las aplicaciones móviles del sector público deberán ser conformes a la EN 301 549 a partir de junio de 2021.
- Porque disponen de **normas claras** para evaluar la conformidad de una página.
- Porque **demuestran** quién está realmente interesado en la accesibilidad.
- Porque las WCAG 2.1 **mejoran** la accesibilidad para los usuarios que acceden mediante dispositivos móviles, y tienen más en cuenta las necesidades de las personas con baja visión y con discapacidad cognitiva.

⁵ Puedes consultarla en “Legislación sobre accesibilidad web en España, Europa y otros países” de Olga Carreras <https://olgacarreras.blogspot.com.es/2005/01/referencia-sobre-legislacion-espaola.html>

Requisitos de Conformidad

En este capítulo se presentan los requerimientos que una página web debe cumplir para que se considere conforme con las WCAG 2.1.

Además, explicamos cómo comunicar el nivel alcanzado mediante las declaraciones de conformidad, que, aun siendo opcionales, son recomendables.



Los 5 requisitos

Para que el contenido de un sitio se considere “conforme” con las WCAG 2.1 debe:

1) Alcanzar uno de los 3 niveles de conformidad

Los criterios de conformidad se categorizan en tres niveles: A, AA y AAA. En el siguiente epígrafe se explican en profundidad.

2) Aplicar a páginas web completas

Las WCAG 2.1 consideran una página web en su conjunto, incluyendo contenidos, funcionalidades, estructura, estilos, textos, vídeos, subtítulos, juegos, sonidos, etc. La *conformidad*, la *conformidad parcial* o la *no-conformidad* se refieren a la **página web completa**. Las distintas variaciones de la página, como las generadas automáticamente para diferentes tamaños y resoluciones de pantalla, también forman parte de esa página y deben ser conformes todas ellas.

Del mismo modo, si la página web tiene información o funcionalidades no accesibles, pero existe una **versión alternativa accesible** enlazada, que los agentes de usuario y productos de apoyo pueden entender y manejar, esa versión alternativa se considera parte de la página, y entonces podemos considerar que la página es accesible.

3) Aplicar a procesos completos

Por otro lado, si la página web es parte de un proceso (por ejemplo, un formulario de compra dividido en varias páginas), debemos considerar la accesibilidad del **proceso completo**. No podemos afirmar que la página de confirmación del pedido es accesible cuando la página anterior de solicitud de pedido no lo es.

4) Usar tecnologías compatibles con la accesibilidad

En todo momento estamos hablando de páginas, de contenidos o de funcionalidades, no de tecnologías. En el capítulo anterior hablamos de las tecnologías compatibles con la accesibilidad y su uso compatible con la accesibilidad. Por ello, ya no hacen falta versiones alternativas a los PDF o al JavaScript si estas tecnologías se hacen accesibles de forma nativa. Recuerda que **sólo podemos comprobar páginas** en su conjunto, no tecnologías.

5) No tener interferencias

Por último, debemos asegurarnos de que determinados contenidos **no impiden el acceso al resto de la página**. Por ello, aunque tengan una alternativa, es obligatorio que siempre se cumplan estos cuatro criterios de conformidad:

- 1.4.2 Control del sonido.
- 2.1.2 Sin trampas para el foco del teclado.
- 2.2.2 Poner en pausa, parar y ocultar.
- 2.3.1 Umbral de tres destellos o menos.

Niveles de conformidad de las páginas web

Dependiendo de los criterios de conformidad superados y de sus niveles, podemos certificar que una **página web** ha alcanzado alguno de estos 3 niveles:

- **Nivel A:** la página (o su alternativa) cumple los 30 criterios de conformidad de nivel A.
- **Nivel AA:** la página (o su alternativa) cumple 50 criterios de conformidad: los 30 de nivel A más los 20 de nivel AA.
- **Nivel AAA:** la página (o su alternativa) cumple los 78 criterios de conformidad: los 30 de nivel A, los 20 de nivel AA y los 28 de nivel AAA.

Recordemos que el nivel más bajo (nivel A) facilita acceder al contenido y el más alto (nivel AAA) mejora su usabilidad y el número de personas que podrán acceder.

La decisión de que nuestras páginas alcancen un nivel u otro viene dado fundamentalmente por los siguientes motivos:

- **Adecuación al público objetivo:** si nuestra página web va a ser usada por personas con diversidad funcional, es lógico y necesario que se adecúe a sus características. Cuanto mayor sea el nivel, mejor complaceremos a nuestro público.
- **Obligación legal:** por ejemplo, en España, los organismos públicos y las empresas privadas con más de 100 trabajadores o que facturen más de 6 millones de euros están obligadas, como mínimo, al nivel AA.
- **Decisiones políticas y personales:** en muchas organizaciones, la sensibilidad hacia los temas sociales implica la decisión de alcanzar mayores niveles de accesibilidad que los legalmente requeridos.
- **Recursos disponibles:** mejorar un sitio web conlleva una serie de recursos humanos, técnicos y económicos que no todas las organizaciones se pueden permitir. La formación del personal, las herramientas a nuestro alcance, el tiempo necesario... son elementos clave que determinarán hasta qué nivel podemos llegar. Estas limitaciones no deberían utilizarse como excusa, sino como indicadores a mejorar en nuestros procesos para aumentar la accesibilidad de los sitios que creamos.

La clave para alcanzar un nivel alto de accesibilidad es la formación de los diferentes perfiles implicados en el desarrollo y mantenimiento del sitio, así como una correcta planificación de los departamentos de sistemas informáticos, diseño, programación y contenidos.

Habitualmente es mucho más complicado reparar sitios web que crearlos desde cero bien planificados. Por ello, prevenir e identificar los problemas de accesibilidad a tiempo reduce significativamente el coste y el tiempo de desarrollo. Para conseguirlo debemos integrar la accesibilidad en todas las fases del desarrollo, tener en cuenta las necesidades de las personas con discapacidad desde el inicio del proyecto y no solo al final del mismo.

Declaraciones de conformidad

Una vez que has comprobado que tu página web es accesible, probablemente quieras comunicar que te has preocupado por ello. La declaración de conformidad indica el nivel de accesibilidad de tus páginas, y normalmente se incluye en un apartado “Accesibilidad” en la cabecera o el pie de las páginas.

Las WCAG 2.1 no obligan a añadir una declaración de conformidad, pero si se hace, hay cierta información que debe incluir obligatoriamente, mientras que otra es opcional. Al final de este capítulo puedes encontrar una plantilla que te ayudará a crear una declaración de conformidad.

Información obligatoria

- Fecha de la declaración de conformidad.
- Las pautas que se han seguido: el título, la versión y la URI.
- El nivel de conformidad alcanzado: A, AA o AAA.
- El alcance de la declaración, es decir, las páginas que cumplen la conformidad. En los ejemplos mostrados más adelante se explica cómo listar las páginas.
- Las tecnologías de contenido web usadas de las que se depende, listadas en la propia declaración, o en una página aparte.

Información opcional

- Criterios de conformidad satisfechos **superiores** al nivel declarado:

“También se cumplen los siguientes criterios de conformidad ... de nivel ...”

- Las **tecnologías** de contenido web usadas, pero de las que no se depende, listadas de la misma manera que las que sí generan dependencia.
- **Agentes de usuario** y productos de apoyo con los que se ha probado la página:

“Los agentes de usuario y productos de apoyo utilizados en las pruebas son: [navegador y versión], en [sistema operativo y versión], con [nombre de plug-in y versión] y con [producto de apoyo y versión]”

- **Pasos dados** -más allá de lo indicado por las WCAG 2.1- para mejorar la accesibilidad. Por ejemplo, si estamos haciendo una página en España con la obligación legal de ser accesible, debemos ofrecer una forma de contacto para dudas, quejas o sugerencias sobre la accesibilidad del sitio web:

“En caso de necesitar ayuda, puede dirigirse a accesibilidad@dominio.com.”

En este apartado, también podemos incluir otras informaciones. Por ejemplo, cuando se usan tecnologías que no son compatibles, pero se enlaza a una versión alternativa accesible, podemos indicar su ubicación y la tecnología empleada:

“Existe una versión accesible del contenido en [url del dominio/version_accesible.html], donde se han utilizado las tecnologías...”

- Cuando haya unos **metadatos** aprobados y publicados por el W3C, podemos incluir una versión de la declaración de conformidad en metadatos, así como una lista de características específicas de accesibilidad del contenido, o un listado de tecnologías usadas de las que se depende.

Ten en cuenta que, aunque las WCAG 2.1 no obligan a incluir la declaración de conformidad, la legislación española sí. En concreto obliga a indicar el **nivel de accesibilidad**, la **fecha de la última revisión** y una forma específica de **contacto** en todos los sitios que tienen la obligación legal de ser accesibles.

Por otra parte, si incluyes el logotipo de conformidad de las WCAG, éste constituye una declaración de conformidad, y por tanto debe ir acompañado de toda la información obligatoria de una declaración de conformidad.



Ilustración 6 Logotipos de conformidad según niveles



Puedes descargar los logotipos de conformidad en el sitio oficial
<https://www.w3.org/WAI/standards-guidelines/wcag/conformance-logos/>

Conformidades parciales

Imagina que has hecho un gran trabajo en el diseño de tu sitio web. Ahora es el momento de que los usuarios añadan el contenido. Sin embargo, esos usuarios pueden ser cualquier persona, incluso de fuera de tu organización, y tú no puedes estar en todos los sitios a todas horas para revisar lo que hacen.

Las WCAG 2.1 ofrecen dos posibles soluciones a este problema:

- Si **monitorizas y reparas** los errores en el contenido externo de tu sitio web en 2 días laborales, puedes usar la declaración de conformidad normal.
- Si no puedes monitorizar y reparar los errores, puedes utilizar una **declaración de conformidad parcial de contenidos procedentes de fuentes no controladas**. Solamente puedes usar esta declaración si ese contenido no está realmente bajo tu control, como los comentarios en un blog, la publicidad insertada dinámicamente o un widget de Twitter. 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 las WCAG 2.1, pero podría cumplirlo si se quitaran los siguientes contenidos procedentes de fuentes no controladas: ...”

Otra figura que contemplan las WCAG 2.1 es la **declaración de conformidad parcial debida al idioma**. Esta declaración se puede usar cuando la tecnología utilizada no posibilita el uso accesible con el idioma (o idiomas) empleados en la página. Por ejemplo, imagina que tienes un sitio web con versiones en español y chino; sin embargo, por limitaciones de la tecnología, los atributos *alt* de las imágenes sólo se muestran en español, por lo que los usuarios de la versión china no podrán comprender la descripción de las imágenes. La forma de indicar esta conformidad parcial sería:

“Esta página no cumple el nivel de conformidad [A, AA, AAA] de las WCAG 2.1, pero podría cumplirlo si se hubiera posibilitado el uso accesible para el idioma chino.”

Ejemplos de declaración de conformidad

Existen varias formas de indicar el alcance de la declaración de conformidad dependiendo de la cantidad de páginas y de las relaciones entre ellas.

Si es aplicable al **sitio completo**, el ideal:

“A [fecha], todas las páginas web en [url del dominio] cumplen con las Pautas de Accesibilidad para el Contenido Web 2.1 (<https://www.w3.org/TR/WCAG21/>) Nivel de conformidad [A, AA, AAA]. Las tecnologías utilizadas de las que se depende son...”

Si es aplicable a **una única página**:

“A [fecha], la página web [título de la página] en [URI de la página] cumple con las Pautas de Accesibilidad para el Contenido Web 2.1 (<https://www.w3.org/TR/WCAG21/>) Nivel de conformidad [A, AA, AAA]. Las tecnologías utilizadas de las que se depende son...”

Si solamente es aplicable a **varias páginas del sitio**, puedes listar las páginas, usar expresiones regulares o lógica booleana. Por simplicidad, recomendamos el listado:

“A [fecha], las páginas web listadas a continuación:
- [título de la página 1] en [URL de la página 1]
- [título de la página 2] en [URL de la página 2]
- ...
- [título de la página n] en [URL de la página n]
cumplen con las Pautas de Accesibilidad para el Contenido Web 2.1
(<https://www.w3.org/TR/WCAG21/>) Nivel de conformidad [A, AA, AAA]. Las tecnologías utilizadas de las que se depende son...”

Si se ha realizado una evaluación con la metodología WCAG-EM, en el capítulo siguiente explicamos cómo declararlo.

Plantilla de declaración de conformidad

Con la ayuda de las siguientes tablas puedes crear las declaraciones de conformidad.

Conformidad (Obligatorio)	
Fecha de la conformidad	[dd/mm/aaaa]
Pautas	WCAG - <i>Web Content Accessibility Guidelines</i>
Versión	2.1
URI	https://www.w3.org/TR/WCAG21/
Nivel de Cumplimiento	[A AA AAA]
Criterios de conformidad superiores alcanzados	(Opcional)
Otras acciones que permiten una mayor accesibilidad	(Opcional)

Alcance de la declaración (Obligatorio)	
Título	URL
...	...

Tecnologías utilizadas (Obligatorio)		
Tecnología	Versión	Genera dependencia
HTML XHTML	...	[Sí No]
ARIA	...	[Sí No]
CSS	...	[Sí No]
JPG	...	[Sí No]
PNG	...	[Sí No]
GIF	...	[Sí No]
SVG	...	[Sí No]
MP3	...	[Sí No]
AVI	...	[Sí No]
Real Video	...	[Sí No]
JavaScript	...	[Sí No]
PDF	...	[Sí No]
Flash	...	[Sí No]
Silverlight	...	[Sí No]
Java	...	[Sí No]
...	...	[Sí No]

Agentes de usuario utilizados en la prueba (Opcional)				
Navegador	Versión	Sistema Operativo	Plug-ins	Productos de apoyo
Internet Explorer
Firefox
...

Evaluar la accesibilidad con WCAG-EM



La metodología WCAG-EM

Existen diferentes metodologías para evaluar la accesibilidad de un sitio web completo. La más recomendable actualmente es la metodología WCAG-EM del W3C, un método fiable que permite determinar el nivel de accesibilidad de cualquier sitio web. Incluso la Comisión Europea recomienda seguirla para la futura metodología europea de monitorización de la accesibilidad de los sitios web y aplicaciones móviles del sector público.



En “Metodologías, certificaciones y entidades certificadoras de la accesibilidad web en España” de Olga Carreras puedes conocer otras formas de evaluar la accesibilidad.

<https://olgacarreras.blogspot.com.es/2007/04/metodologa-certificaciones-y-entidades.html>

Los objetivos de la metodología WCAG-EM son:

- Servir como guía a los evaluadores.
- Promover una serie de buenas prácticas.
- Evitar errores habituales.
- Lograr que los resultados de diferentes evaluaciones, del mismo sitio o de sitios diferentes, sean comparables.

La metodología WCAG-EM consta de 5 pasos, algunos de los cuales pueden solaparse o lleverse a cabo en paralelo:

1. Definir el alcance de la evaluación.
2. Explorar el sitio web.
3. Seleccionar una muestra representativa de páginas.
4. Auditar la muestra seleccionada.
5. Registrar los resultados de la evaluación y elaborar un informe.



Ilustración 7 Pasos de la metodología WCAG-EM

Paso 1. Definir el alcance de la evaluación

El primer paso es determinar sin ambigüedades el alcance de la evaluación.

La metodología se aplica siempre a un **sitio web completo**: no pueden excluirse páginas que sirvan a la finalidad y funcionalidad principal del sitio, y que por tanto forman parte de la navegación, el diseño y los procesos completos del mismo.

Sin embargo, la evaluación puede centrarse en ámbitos claramente separables de un sitio web, como podría ser la parte pública y la parte privada, o diferentes versiones de este: versión móvil, versión en determinado idioma, etc.

En esta fase se define también:

- El **nivel de adecuación** (A, AA, AAA) que se va a evaluar.
- El **listado de navegadores web, productos de apoyo u otros agentes de usuario** con los que las características de accesibilidad deben ser compatibles. Este listado puede ser muy diferente según estemos analizando un sitio público o una intranet.
- Los **requisitos de evaluación adicionales** acordados entre el evaluador y el cliente, por ejemplo: el nivel de detalle del informe final, casos de uso especiales, la participación de personas con discapacidad, etc.

Paso 2. Explorar el sitio web

Una vez definido el alcance es necesario comprender el uso, propósito y funcionalidad del sitio para determinar qué páginas incluir en la muestra a evaluar.

Para ello es necesario identificar:

- Las páginas o estados de **páginas relevantes**. Deben incluirse las páginas:
 - de inicio,
 - de inicio de sesión,
 - de mapa del sitio,
 - de contacto,
 - de ayuda,
 - de información legal
 - otras similares normalmente enlazadas desde todas las páginas.
- Las **funcionalidades clave**, por ejemplo, “selección y compra de un producto”, “registrar una cuenta”, “rellenar y enviar un formulario”, etc.
- **Diferentes tipos de páginas y estados** de las mismas, por ejemplo:
 - creadas a partir de diferentes plantillas,
 - con diferentes contenidos: formularios, tablas, multimedia, etc.
 - con diferentes componentes funcionales: carruseles, acordeones, ventanas modales, etc.
 - que usan distintas tecnologías: JavaScript, WAI-ARIA, PDF, etc.
- Las **páginas o estados de páginas relevantes para las personas con discapacidad** o para la accesibilidad del sitio, como las páginas de accesibilidad, ayuda o preferencias.
- Las **tecnologías usadas de las que depende el contenido**, es decir, que el contenido no sería conforme si esa tecnología se desconectara o el navegador no la soportara. Las tecnologías pueden ser HTML, CSS, JavaScript, ARIA, SMIL, SVG, PDF, Flash, Silverlight, etc.

Paso 3. Seleccionar una muestra representativa de páginas

Lo ideal sería poder evaluar el sitio completo, pero en muchos casos el volumen de páginas del portal imposibilita la evaluación manual de todas ellas.

Ahora que conocemos en profundidad el sitio web, podemos seleccionar una muestra de páginas que lo represente, de manera que la evaluación de esa muestra refleje la accesibilidad de todo el sitio con suficiente fiabilidad.

El tamaño de la muestra dependerá de diferentes factores, como el volumen, la complejidad o la consistencia del sitio. Si seleccionamos cuidadosamente las páginas y procesos que se incluyen en la muestra, reduciremos el tamaño de la misma, pero seguirá representando adecuadamente a todo el sitio.

En la muestra se tienen que **incluir procesos completos**. Por ejemplo, si un proceso de compra está compuesto por tres pasos, no se puede incluir en la muestra solo el primer paso, se deberían incluir las páginas de todos los pasos.

Nuestra muestra se compone de dos partes.

Por un lado, una **muestra de páginas seleccionadas**, en la que debemos incluir:

- Las páginas relevantes para todos los usuarios.
- Las páginas relevantes específicamente para las personas con discapacidad.
- Las páginas con funcionalidades esenciales.
- Páginas de diferente tipo.
- Páginas que dependen de diferentes tecnologías.

Por otro lado, una **muestra de control**, que consiste en una selección de páginas y estados de páginas dentro del alcance definido, pero al azar, escogidas sin un patrón predecible. Su tamaño debe ser un 10% de la muestra de páginas seleccionadas, es decir, si la muestra tiene 50 páginas, deberemos añadir al azar 5 páginas más como muestra de control.

La muestra de control será el indicador que nos permita verificar los resultados y aumentar la confianza en los mismos.

Paso 4. Auditar la muestra seleccionada

Es el momento de verificar si las páginas de la muestra cumplen con todos los criterios de conformidad del nivel elegido y con los cinco requisitos de conformidad de las WCAG 2.1

En la evaluación de cada página se incluye cualquier contenido que contenga, aunque sea agregado o embebido, como un mapa o como un contenido de terceros.

Cada componente de la interfaz debe evaluarse **en sus diferentes estados** para verificar que cumple los criterios de conformidad en todos ellos. Por ejemplo, un campo de texto de un formulario debe evaluarse en su estado inicial, cuando recibe el foco, cuando se ha escrito en su interior, cuando se ha producido un error de validación, etc.

Además, deben comprobarse con diferentes navegadores (Chrome, Firefox, Internet Explorer, Edge, Safari, etc.) y productos de apoyo (por ejemplo, los lectores de pantalla NVDA, JAWS, VoiceOver, etc.; el magnificador de pantalla ZoomText; etc.) acordados durante la definición del alcance en el paso 1.

Cuando los **componentes son comunes a muchas páginas**, como la cabecera o el pie de página, no necesitan ser evaluados repetidamente. Cada uno de ellos puede analizarse de forma independiente para no incluir los resultados en el análisis de cada página.

Por otra parte, **si un contenido** relacionado con un determinado criterio de conformidad **no está presente** (por ejemplo, no hay videos en la página y el criterio evalúa si los videos están subtitulados), el criterio se considera “satisfactorio” o “no presente”.

Una vez obtenidos los resultados, se comparan los de la muestra seleccionada de forma estructurada con los de la muestra de control.

Si en la muestra al azar hay contenidos o resultados no representados en la muestra estructurada, es que la muestra estructurada no es suficientemente representativa. En este caso, deberemos volver al paso 2 y 3 para redefinir la muestra.



En el capítulo *Recursos y herramientas* puedes encontrar aplicaciones que te ayudarán a realizar la auditoría.

Paso 5. Registrar los resultados y elaborar un informe

Debemos ir documentando todo el proceso, y especialmente los resultados de la evaluación, para poder elaborar finalmente un informe. No todos los datos registrados se tienen (o se pueden) incluir obligatoriamente en el informe, por ejemplo, por motivos de confidencialidad.

Al menos se debe documentar:

- **Acerca de la evaluación:** nombre del evaluador; nombre de la persona, empresa u organización que ha solicitado la evaluación; y la fecha en la que se ha llevado a cabo la misma.
- **Alcance de la evaluación** (paso 1): alcance, nivel de adecuación (A, AA, AAA) evaluado, soporte de accesibilidad definido y requisitos adicionales acordados.
- **Exploración del sitio** (paso 2): tecnologías de las que se depende y, opcionalmente, las páginas y tecnologías identificadas.
- **Muestra representativa** (paso 3): listado de todas las páginas que conforman la muestra evaluada, diferenciando las páginas de la muestra seleccionada de forma estructurada y las páginas de la muestra de control.
- **Auditoría de la muestra** (paso 4): el resultado de la evaluación, bien por cada página de la muestra o bien por la muestra en su conjunto. Debemos indicar si se cumple o no el nivel acordado (A, AA o AAA) incluyendo al menos un ejemplo por cada criterio que no se cumple. También podemos añadir más información, según el nivel de detalle acordado en el paso 1, como el cumplimiento detallado por cada criterio de conformidad, o como todos los errores encontrados y las soluciones recomendadas para cada uno de ellos.

Aunque sea a nivel interno, se recomienda guardar todos los detalles de la evaluación, ya que puede ser muy útil en caso de conflicto: herramientas utilizadas, capturas de pantalla, datos introducidos en las páginas para poder replicar los resultados, etc.

Declarar la conformidad tras seguir la WCAG-EM

Como vimos anteriormente, la declaración de conformidad de las WCAG 2.1 no puede hacerse para sitios web completos en base a la evaluación de un subconjunto de páginas, ya que siempre es posible que haya errores de conformidad en otras páginas.

Aunque con la metodología WCAG-EM se minimiza este problema, en realidad no estamos analizando todas las páginas del sitio, y por ello no puede dar como resultado una declaración de conformidad para el sitio completo.

La declaración de conformidad resultante deberá ser sobre los resultados obtenidos tras aplicar esta metodología, y tendrá la misma información que la definida como obligatoria en la declaración de conformidad de las WCAG 2.1.

Mostramos dos declaraciones de conformidad para resaltar las diferencias entre ambas versiones:

Ejemplo de declaración de conformidad de sitio completo:

“A 20 de diciembre de 2018, todas las páginas de <http://www.ejemplo.com> son conformes con las Web Content Accessibility Guidelines 2.1 (WCAG 2.1) en su nivel de conformidad AA.

El conjunto de las tecnologías compatibles con la accesibilidad de las que se depende para esta declaración está listado en <http://ejemplo.com/listado.html>. ”

Ejemplo de declaración de conformidad tras una auditoría basada en la selección de una muestra de páginas:

“A 20 de diciembre de 2018, se ha realizado una auditoría de accesibilidad del sitio <http://www.ejemplo.com> siguiendo la metodología WCAG-EM del W3C.

El resultado de la auditoría satisface el nivel de conformidad AA de las Web Content Accessibility Guidelines 2.1 (WCAG 2.1).

El conjunto de las tecnologías compatibles con la accesibilidad de las que se depende para esta declaración está listado en <http://ejemplo.com/listado.html>”

Guías y enlaces de interés

Documentación oficial de la metodología WCAG-EM.

<http://www.w3.org/TR/WCAG-EM/>

Traducción resumida de la metodología WCAG-EM de Olga Carreras.
<https://olgacarreras.blogspot.com.es/2012/04/metodologia-de-evaluacion-de.html>

Principio 1. Perceptible

Nuestro sitio web puede ser visitado por personas con necesidades y preferencias muy diferentes, pero también por robots (arañas de buscadores, traductores automáticos...).

Nuestra información y los componentes de la interfaz de usuario deben tener en cuenta esta circunstancia. Debemos proporcionar alternativas si el usuario no puede usar uno de sus sentidos.



Pauta 1.1

Ofrece alternativas textuales

Proporciona 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.

La información textual se puede representar de una forma visual, sonora o táctil, o mediante una combinación de estas formas para adecuarse a las necesidades del usuario. Por ejemplo, una persona ciega puede comprender una imagen si se le ofrece una descripción textual de la misma; o una persona sorda puede comprender un sonido si se le presenta un texto que lo describa.

Contenido no textual

Todo el contenido no textual que se presenta al usuario tiene una alternativa textual que cumple el mismo propósito, excepto en:

Controles o mecanismos para que el usuario introduzca datos: tienen un nombre que describe su propósito.

Presentaciones multimedia con desarrollo temporal: las alternativas textuales proporcionan al menos una identificación descriptiva del contenido no textual.

Pruebas o ejercicios que no serían válidos si se presentaran en forma de texto: las alternativas textuales proporcionan al menos una identificación descriptiva del contenido no textual.

Contenidos cuyo objetivo principal es ofrecer una experiencia sensorial específica: las alternativas textuales proporcionan al menos una identificación descriptiva del contenido no textual.

CAPTCHA o método análogo para confirmar que quien está accediendo al contenido es una persona y no una máquina: se proporcionan alternativas textuales que identifican y describen el propósito del contenido no textual y se proporcionan formas alternativas de verificación mediante distintos tipos de percepciones sensoriales.

Contenido decorativo, utilizado para dar formato a la página o que no se presenta a los usuarios: se implementa de forma que pueda ser ignorado por los productos de apoyo.

Dependiendo de la situación, se debe ofrecer una alternativa de descripción larga o de descripción corta. A continuación, se describen y se explica cuándo usar una u otra.



Si dudas sobre qué técnica de descripción corta o larga debes utilizar,
sigue los pasos de estos recursos de Olga Carreras:

Wizard

https://www.usableyaccesible.com/textosalternativosaccesibles/texto_alternativo_wizard.php

Mapa de decisión

https://www.usableyaccesible.com/textosalternativosaccesibles/mapa_decision_texto_alternativo.php

Alternativas de descripción corta

- Utiliza el atributo *alt* en imágenes, las *area* de los mapas de imagen y los *applets*.
- Usa los atributos *aria-label* o *aria-labelledby*.
- Utiliza el cuerpo del elemento *object*.
- Combina una imagen y un texto adyacente si enlazan al mismo sitio.
- Ofrece una alternativa textual para los dibujos en arte ASCII, los emoticones y el texto *leet*.
- Describe textualmente el objetivo de los enlaces.
- En un grupo de imágenes relacionadas entre sí, utiliza la descripción del primer elemento para describirlos a todos.

Alternativas de descripción larga

- Utiliza el atributo *longdesc* en imágenes.
- Utiliza el atributo *aria-describedby* en imágenes.
- Incluye la descripción larga en un archivo externo, y referéncialo en la descripción corta o junto al contenido no textual.
- Utiliza el cuerpo en el elemento *object*.

Técnicas para cumplir el criterio

Si una descripción corta puede cumplir el mismo objetivo y presentar la misma información que el contenido no textual:

- Utiliza una alternativa de descripción corta.

Si una descripción corta no puede cumplir el mismo objetivo y presentar la misma información que el contenido no textual (por ejemplo, una gráfica o un diagrama):

- Utiliza una alternativa de descripción corta más una de descripción larga.

Si el contenido no textual es un control o un campo de entrada de formulario, identifica su objetivo con una de estas técnicas:

- en las *area* de mapas de imagen o en las imágenes usadas como botones, el atributo *alt*;
- en controles y campos de formulario, elementos *label* o atributos *aria-label* o *aria-labelledby*; y si no puedes usarlos, el atributo *title*;
- en enlaces, textos descriptivos;

Si el contenido no textual es un audio o un video:

- Describe el contenido no textual o su objetivo con una alternativa de descripción corta.

Si el contenido no textual es un examen o ejercicio que no sería válido si se presentara en modo texto:

- Describe el contenido no textual o su objetivo con una alternativa de descripción corta.

Si el principal objetivo del contenido no textual es una experiencia sensorial específica:

- Describe el contenido no textual o su objetivo con una alternativa de descripción corta.

Si el contenido no textual es un CAPTCHA:

- Describe su objetivo y ofrece un mecanismo diferente para conseguir ese mismo fin.

Si las imágenes deben ser ignoradas por los productos de apoyo:

- Deja el atributo *alt* vacío y no añadas el *title*; o
- Inclúyelas como imágenes decorativas mediante CSS.

Técnicas para mejorar la usabilidad

- Identifica el contenido informativo no textual.
- Haz breve la descripción corta, y describe el contenido no textual más profundamente sólo cuando sea necesario.
- Ofrece diferentes tamaños para el contenido no textual cuando no pueda haber una alternativa accesible equivalente.

Si usas contenido no textual para mejorar el aspecto gráfico de la página:

- Formatea la página con las reglas de caja de CSS en lugar de con imágenes.
- Incluye imágenes decorativas con la propiedad *background* o los pseudo-elementos *:before* y *:after* de CSS.
- Muestra las celdas vacías de las tablas.
- Indica qué elementos son meramente de presentación con el rol *presentation* de ARIA.

Si usas marcos:

- Diseña para navegadores que no soporten el elemento *frame*.
- Ofrece contenido alternativo para el elemento *iframe*, pero no uses descripciones largas.

Si el contenido no textual es multimedia:

- Asocia contenidos en vídeo y solo audio con transcripciones de texto con metadatos.
- Complementa los elementos *embed* con los elementos *noembed*.

Si el contenido es una emisión en directo:

- Enlaza la información textual que ofrezca información similar. Por ejemplo, para una *webcam* de tráfico, incluye un enlace a una página con información del tráfico.

Si el contenido no textual es un mapa de imágenes:

- Incluye enlaces de texto redundantes junto al mapa.

Si el contenido no textual es una imagen de texto:

- Ofrece el mismo texto que se incluye en las imágenes de texto.
- Redimensiona las imágenes de texto mediante programación de servidor.

Si el contenido no textual es un CAPTCHA:

- Ofrece más de dos modalidades de CAPTCHA.
- Proporciona acceso al personal de atención al cliente para poder eludir el CAPTCHA.
- No pidas un CAPTCHA a usuarios que ya se hayan autenticado.

¡No lo hagas!

- No omitas el atributo *alt* en los elementos *img*, *area* e *input* de tipo *image*, aunque éstos se usen sólo para decorar. Déjalo vacío.
- No utilices una alternativa textual que no sea realmente descriptiva. Por ejemplo, *alt=spacer*, *alt=icon.jpg*, *alt=imagen2...*
- No utilices una alternativa textual descriptiva para imágenes que deben ser ignoradas.
- No utilices CSS para incluir imágenes que contengan información.
- No olvides actualizar las alternativas textuales cuando cambie el contenido no textual.
- No utilices efectos visuales para representar texto sin ofrecer una alternativa textual.
- No utilices arte ASCII sin ofrecer una alternativa textual.

Pauta 1.2

Medios tempodependientes

Proporciona alternativas a los medios tempodependientes.

En este capítulo vamos a hablar de los contenidos en vídeo y audio, así como de otros componentes interactivos en los que el tiempo es una parte importante de la experiencia sensorial.

Cuando hablamos de **medios tempodependientes**, nos referimos a:

- audio solo;
- vídeo solo (entendido como imágenes en movimiento);
- audio y vídeo combinados;
- audio y/o vídeo combinados con mecanismos de interacción.

Estos medios tempodependientes pueden ser tanto grabados como en directo.

Por otro lado, cuando nos referimos a **medios sincronizados**, nos referimos a los contenidos de audio o de vídeo sincronizados con otro formato para presentar información y/o con componentes interactivos basados en el tiempo (excepto cuando es un contenido multimedia alternativo a un texto). Por ejemplo, una película se considera “multimedia sincronizado” porque tiene audio y vídeo sincronizado entre sí. O por ejemplo un juego interactivo, que presenta contenido visual y/o auditivo y cuyos elementos de interacción usamos en determinados momentos.

Otro término que aparece en esta pauta es el de **documento alternativo al medio tempodependiente** (lo que entendemos habitualmente por transcripción), el cual es un contenido que incluye una secuencia correcta de descripciones textuales de la información visual o auditiva y que permite al usuario conseguir los mismos resultados que cualquier interacción basada en el tiempo. Por ejemplo, el guion empleado para crear un contenido multimedia sincronizado si representa con precisión el contenido resultante, es decir, si describe tanto la parte visual (acciones, personas, texto en pantalla, expresiones corporales, etc.) como la parte auditiva (diálogos, música, risas, aplausos, etc.) del contenido.

En la siguiente página mostramos un cuadro resumen de los diferentes medios y alternativas según sus características y el nivel de accesibilidad al que deseemos llegar. Es posible que eches de menos algún caso, como medios sincronizados en directo. En esos casos, utiliza el sentido común y los medios a tu alcance para ofrecer la alternativa más adecuada.

Tabla 2 Resumen de medios tempodependientes

Medio	Tipo	Nivel	Alternativa	Criterio
Solo Audio	Grabado	A	Transcripción textual	1.2.1
	En directo	AAA	Transcripción textual	1.2.9
Solo Vídeo	Grabado	A	Transcripción textual o Pista de audio con la misma información	1.2.1
			Transcripción textual	1.2.8
		AAA		
Medio sincronizado vídeo + audio o vídeo + audio + interacción o audio+interacción o vídeo+interacción	Grabado	A	Subtítulos + Audiodescripción o Transcripción textual	1.2.2 1.2.3
		AA	Subtítulos + Audiodescripción	1.2.5
		AAA	Subtítulos + Audiodescripción (ampliada si es necesario) + Intérprete en lengua de signos + Transcripción textual	1.2.6 1.2.7
		En directo	Subtítulos	1.2.4
		AA		



Además, **todos los medios tempodependientes, ya sean en directo o grabados,** deben contar con un título y/o una descripción, que permita a los usuarios decidir si les merece la pena activar ese contenido. (Criterio 1.1.1)

Audio solo o vídeo solo (grabado)

Audio solo grabado: se proporciona un *documento alternativo al medio tempodependiente* que presenta información equivalente a su contenido.

Vídeo solo grabado: se proporciona un *documento alternativo al medio tempodependiente* o se proporciona una pista sonora que presenta información equivalente a su contenido.

Excepciones: cuando el audio o el vídeo son una alternativa al texto y están claramente identificados como tales.

Técnicas para cumplir el criterio

Si el contenido es audio solo grabado:

- Ofrece una alternativa que presente la misma información.

Si el contenido es vídeo solo grabado:

- Ofrece una alternativa que presente la misma información; o bien
- Ofrece un contenido sonoro que describa el contenido más importante del vídeo.

Técnicas para mejorar la usabilidad

- Ofrece una transcripción del contenido sonoro.
- Enlaza a textos que ofrezcan la misma información.

¡No lo hagas!

- No utilices una alternativa textual que no sea descriptiva.
- No ofrezcas descripciones largas que no cumplan el mismo objetivo o no presenten la misma información que el contenido no textual.

Audio sincronizado con subtítulos (grabado)

Audio grabado dentro de un contenido multimedia sincronizado: se proporcionan subtítulos para el contenido.

Excepción: si el contenido multimedia sincronizado es una alternativa al texto y está claramente identificado como tal.

Para entender este criterio, es necesario conocer algunos conceptos sobre los subtítulos. Existen varios tipos de subtítulos. Dependiendo de su funcionalidad, pueden ser:

- **Subtítulos generales** (en inglés *subtitles*): acompañan los diálogos de los personajes, voces en *off* y elementos que se traducen.
- **Subtítulos para sordos** (en inglés *captions*): subtítulos generales que incluyen información sobre los sonidos que suceden (“murmurlos de fondo”) o la entonación (“gritando”).

Dependiendo de cómo se incorporan a la imagen, pueden ser:

- **De forma abierta** (en inglés *open*): están incrustados en la propia imagen del vídeo, por lo que no se pueden alterar ni ocultar, salvo ubicando encima otros subtítulos.
- **De forma cerrada** (en inglés *closed*): disponibles en un fichero externo, un programa auxiliar los presenta junto al vídeo. Se pueden cambiar de color, posición o tamaño.

Técnicas para cumplir el criterio

- Ofrece subtítulos para sordos de forma abierta.
- Ofrece subtítulos para sordos de forma cerrada usando la tecnología SMIL u otro formato que soporte el reproductor.

Técnicas para mejorar la usabilidad

- Si un vídeo no tiene sonido, avísalo diciendo “Vídeo sin sonido”.
- Subtitula en todos los idiomas para los cuales haya pistas de sonido con SMIL.

¡No lo hagas!

- No omitas los diálogos o efectos de sonido importantes en los subtítulos para sordos.
- No ofrezcas un medio sincronizado sin subtítulos para sordos cuando ese medio presente información en la página.
- No te olvides de etiquetar la alternativa textual para un medio sincronizado.

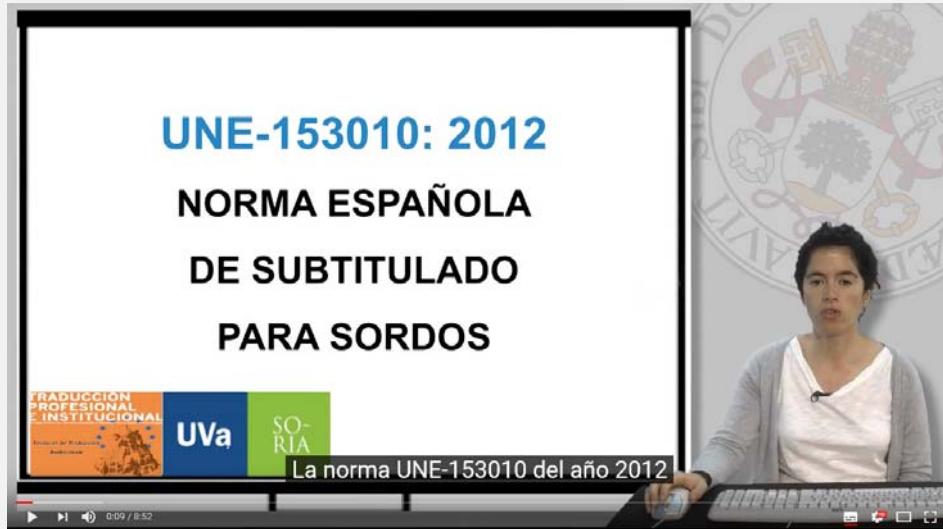


La norma UNE 153010:2012 define buenas prácticas a seguir en la generación de subtítulos (color, tipografía, colocación de los subtítulos, número de líneas y caracteres por línea, velocidad, identificación de caracteres, etc.).

<https://www.aenor.com/normas-y-libros/buscador-de-normas/UNE/N0049426>

En el vídeo “UNE-153010:2012 Norma Española De Subtitulado Para Sordos” de la Universidad de Valladolid la explican detalladamente.

<https://www.youtube.com/watch?v=UhUfnctQKYQ>



Vídeo con audiodescripción o medio alternativo (grabado)

Vídeo grabado dentro de un contenido multimedia sincronizado: se proporciona un documento alternativo al medio tempodependiente o una audiodescripción de todo el contenido.

Excepción: si el contenido multimedia sincronizado es una alternativa al texto y está claramente identificado como tal.

La **audiodescripción** es una técnica destinada a personas con discapacidad visual, mediante la cual se suministra información sonora que traduce o explica un contenido visual, como los gestos de un personaje, el vestuario, los paisajes... aprovechando los silencios entre los diálogos.

Existe también la variante de **audiodescripción ampliada**, la cual es una audiodescripción que se añade a una presentación audiovisual pausando el vídeo, por lo que se obtiene el tiempo suficiente para añadir las informaciones adicionales necesarias para explicar el contenido.



En esta muestra del corto “PACO” de Catarsis Producciones y Contrasentido PC tienes un ejemplo de audiodescripción https://www.youtube.com/watch?v=S1MPQbwS_Q

Una captura de pantalla del corto "PACO". La imagen es negra y en el centro aparece el nombre "PACO" en grandes letras blancas con un efecto de sombra. La "A" tiene un efecto degradado que pasa por el color rosa.

Técnicas para cumplir el criterio

- Ofrece una alternativa para medios tempodependientes enlazando la alternativa junto al contenido no textual; o usando el cuerpo del elemento *object*.
- Añade una segunda pista de sonido, seleccionable por el usuario, que incluya la audiodescripción.
- Ofrece una versión del vídeo con una audiodescripción -o una audiodescripción ampliada- usando SMIL o usando cualquier reproductor que soporte audio y vídeo.

Técnicas para mejorar la usabilidad

- Añade audiodescripciones en varios idiomas usando SMIL.

Audio sincronizado con subtítulos (en directo)

Audio en directo en medios sincronizados: se proporcionan subtítulos para todo el contenido.

Técnicas para cumplir el criterio

- Ofrece subtítulos para sordos de forma abierta; o bien
- Ofrece subtítulos para sordos de forma cerrada usando la tecnología SMIL u otro formato que soporte el reproductor.

Vídeo con audiodescripción (grabado)

Vídeo grabado dentro de un contenido multimedia sincronizado: se proporciona una audiodescripción de todo el contenido.

Técnicas para cumplir el criterio

- Aporta una segunda pista de sonido, seleccionable por el usuario, que incluya una audiodescripción.
- Ofrece una versión del vídeo con una audiodescripción -o una audiodescripción ampliada- usando SMIL o usando cualquier reproductor que soporte audio y vídeo.

Técnicas para mejorar la usabilidad

- Añade una audiodescripción en varios idiomas usando SMIL.
- Ofrece una audiodescripción para medios sincronizados emitidos en directo.

Audio sincronizado con lengua de signos (grabado)

Audio grabado dentro de contenido multimedia sincronizado: se proporciona una interpretación en lengua de signos para todo el contenido.

Existen distintas formas de mostrar al intérprete de lengua de signos o lengua de señas, bien incrustado dentro del flujo de vídeo, o bien en una pantalla diferente que se puede ubicar en cualquier posición, incluso superpuesta en pantalla.



El reproductor Able Player puede integrar la interpretación en lenguaje de signos tanto dentro del flujo (primera imagen) como en una pantalla diferente (segunda imagen)
<http://sidar.org/ableplayer2/tests/test16.html>



Técnicas para cumplir el criterio

- Incluye un intérprete de lengua de signos dentro del flujo de vídeo.
- Ofrece un vídeo sincronizado del intérprete de lengua de signos que puede ser mostrado en una pantalla diferente o superpuestado en la imagen usando SMIL.

Técnicas para mejorar la usabilidad

- Asocia la alternativa en lengua de signos al vídeo con metadatos para permitir al usuario elegir la lengua de signos.

Vídeo con audiodescripción ampliada (grabado)

Vídeo grabado dentro de contenido multimedia sincronizado: si las pausas en el audio del primer plano son insuficientes para permitir que la audiodescripción comunique el significado del vídeo, se proporciona una audiodescripción ampliada para todos los contenidos.

Recuerda que la audiodescripción ampliada es una audiodescripción que se añade a una presentación audiovisual poniendo en pausa el vídeo, por lo que permite disponer de tiempo para añadir informaciones adicionales.

Técnicas para cumplir el criterio

- Ofrece un vídeo con una audiodescripción ampliada usando SMIL o usando cualquier reproductor que soporte audio y vídeo.

Técnicas para mejorar la usabilidad

- Añade audiodescripciones ampliadas en varios idiomas usando SMIL.

Vídeo solo o medio sincronizado con un medio alternativo (grabados)

Vídeo solo grabado: se proporciona un documento alternativo al medio tempodependiente.

Medio sincronizado grabado: se proporciona un documento alternativo al medio tempodependiente.

Técnicas para cumplir el criterio

Si el contenido es un medio sincronizado grabado:

- Enlaza la alternativa inmediatamente junto al contenido no textual o dentro del cuerpo del elemento *object*.

Si el contenido es vídeo solo grabado:

- Ofrece una alternativa al contenido en medios tempodependientes.

Técnicas para mejorar la usabilidad

- Complementa los elementos *embed* con el atributo *noembed* para proporcionar la alternativa adicional.
- Ofrece el guion del vídeo.
- Añade detalles a la audiodescripción para que sea más comprensible.

¡No lo hagas!

- No omitas la alternativa textual correctamente etiquetada en los medios sincronizados.

Audio solo (en directo)

Audio solo en directo: se proporciona un *documento alternativo al medio tempodependiente* que presenta información equivalente al contenido.

Técnicas para cumplir el criterio

- Ofrece un enlace a la transcripción textual del guion previsto antes del directo o del guion que finalmente se siguió.
- Ofrece alternativas basadas en texto para contenido de sólo audio en directo.
- Incorpora un servicio de subtítulado en directo en la página web.

Técnicas para mejorar la usabilidad

- Asocia las transcripciones textuales al contenido mediante metadatos.

Pauta 1.3

Adaptable, presenta el contenido de diferentes formas

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

Los usuarios de tu sitio web tienen necesidades y preferencias muy diferentes, por lo que tienes que comprobar que tu contenido esté disponible de tal forma que pueda ser percibido por todos los usuarios en diferentes maneras, con diferentes agentes de usuarios y en diferentes dispositivos.

Si hay información incorporada dentro de la estructura de la página y los productos de apoyo no pueden diferenciar la información de la estructura, el usuario tampoco podrá acceder a la información (por ejemplo, si se ha maquetado la página con tablas anidadas complejas). Para conseguirlo, hay que hacer que toda la información sea identificable y procesable mediante software.

Las WCAG 2.1 definen la **estructura** como “la forma en la que tanto las páginas que componen el sitio, como los elementos que componen una página web, se organizan y se relacionan unos con otros”. Asimismo, definen la **presentación** como “el procesado y entrega del contenido de forma que pueda ser percibido por los usuarios”.

Información y relaciones

La información, estructura y relaciones comunicadas a través de la presentación pueden ser determinadas por software o están disponibles como texto.

Técnicas para cumplir el criterio

Si la tecnología ofrece una estructura semántica para comunicar la información y las relaciones (por ejemplo, HTML):

- Formatea la estructura y el texto con elementos semánticos.
- Utiliza roles de ARIA para identificar las zonas y regiones de la página (cabecera, menú, zona principal, pie, etc.) y dales un nombre con el atributo *aria-labelledby*.
- Comunica con texto la información que es transmitida por las variaciones en la presentación del texto.
- Separa 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.
- Haz que la información y las relaciones transmitidas a través de la presentación también estén presentes en el código mediante:
 - los elementos *h1-h6* para identificar encabezados, o el *role=heading* de ARIA;
 - los elementos *ol*, *ul* y *dl* para listas;
 - el marcado semántico cuando se utilizan claves de colores (por ejemplo, si se colorea el texto en rojo para indicar que es un error);
 - las funciones del DOM para añadir contenido a la página mediante programación.
- Si tienes que presentar datos tabulares utiliza el elemento *table* y añade:
 - el elemento *caption* para titular la tabla de datos;
 - el atributo *summary*⁶ o, preferiblemente, *aria-describedby* para resumir la tabla;
 - el atributo *scope* para asociar celdas de encabezado con celdas de datos;
 - atributos *id* y *header* para asociar celdas de datos con celdas de encabezado.
- Estructura los formularios mediante:
 - 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 el *label*;

⁶ El atributo *summary* se utilizaba en tablas en HTML 4 y XHTML. En HTML 5 ya no se admite, y no se debe confundir con el elemento *summary* dentro del elemento *details*. Más información sobre este elemento en <https://developer.mozilla.org/en-US/docs/Web/HTML/Element/summary>

- los elementos *fieldset* y *legend* para describir grupos de controles de formulario;
- el atributo *optgroup* para agrupar elementos *option* dentro de un elemento *select*.
- los roles de ARIA para identificar controles relacionados y el atributo *aria-labelledby* para darles un nombre

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

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

Técnicas para mejorar la usabilidad

- Controla la presentación visual del texto con CSS.
- Maqueta la página con CSS, en lugar de hacerlo con tablas.
- Ubica las etiquetas para maximizar la predictibilidad de las relaciones entre éstas y los controles a los que representan.
- Ofrece etiquetas para todos los controles de formulario que no tienen etiquetas implícitas.
- Organiza la página usando encabezados.
- Identifica los campos obligatorios de formulario con el atributo *aria-required*
- Usa el atributo *aria-describedby* para dar una descripción a las etiquetas de los controles

¡No lo hagas!

- No utilices los cambios de presentación del texto para mostrar información sin usar el marcado o el texto adecuado.
- No confundas al navegador con información insuficiente en el DOM (*Document Object Model*) para poder determinar las relaciones entre elementos en HTML o en XML (por ejemplo, dos etiquetas con el mismo atributo *id*).
- No utilices caracteres en blanco en el texto plano para crear columnas.
- No maquetes el texto plano como si fuera una tabla.
- No utilices eventos de programación para emular enlaces en una forma que no sea determinable por software.
- No utilices el marcado de estructura en una forma que no represente relaciones con el contenido.
- No utilices elementos *th* o *caption*; o atributos *summary* llenos en tablas que sirvan para diseñar la página.

- No utilices el elemento *pre* para marcar información tabular.
- No asocies un elemento *label* a un control de interfaz de usuario en una forma que no sea determinable por software.
- No insertes contenido informativo usando los selectores *:before* y *:after*, o la propiedad *content* en CSS.

Secuencia significativa

Cuando la secuencia en que se presenta el contenido afecta a su significado, se puede determinar por software la secuencia correcta de lectura.

Técnicas para cumplir el criterio

- Ajusta el orden del DOM al orden visual.
- Ordena todos los contenidos de la página web en un orden que tenga sentido y maquétalos:
 - usando marcas de Unicode para mezclar texto “de derecha a izquierda” y “de izquierda a derecha”.
 - añadiendo el atributo *dir* en elementos en línea para resolver problemas con textos que necesitan diferentes direcciones de lectura y que se presentan anidados.
 - ubicando el contenido según el marcado de estructura.
 - controlando el espacio entre letras mediante CSS.

Técnicas para mejorar la usabilidad

- Alinea el texto a la izquierda en idiomas que se escriban de izquierda a derecha; y alinea el texto a la derecha en idiomas que se escriban de derecha a izquierda.
- Ofrece un enlace a una representación lineal del contenido.
- Ofrece cambiar el estilo de la presentación de los elementos mediante hojas de estilo que cambien el orden de presentación.

¡No lo hagas!

- No utilices una tabla para maquetar la página que no tenga sentido cuando se muestre de forma lineal.
- No cambies con CSS el significado del contenido por la forma de ubicar la información.
- No utilices caracteres de espacio en blanco:
 - para formatear contenido como si fuera una tabla;
 - para crear varias columnas en texto plano; o
 - para controlar el espaciado dentro de una palabra.

Características sensoriales

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

Los usuarios de lector de pantalla tendrán difícil pulsar en el “botón redondo, grande, rojo, a la derecha de la pantalla”, por muy bien que esté descrito, puesto que no lo pueden diferenciar del resto de controles que les dicta su programa de apoyo.

Por ello es necesario etiquetar e identificar correctamente cada control, y no servirse solamente de las peculiaridades visuales o auditivas de los elementos en pantalla.

Técnicas para cumplir el criterio

- Ofrece identificación textual de los elementos que de otra manera sólo dependerían de la información sensorial para ser comprendidos.

Técnicas para mejorar la usabilidad

- Incluye los símbolos gráficos mediante imágenes con una alternativa textual, evitando los dibujos (glifos) de las fuentes Unicode.

¡No lo hagas!

- No identifiques el contenido sólo por su forma o localización.
- No utilices únicamente un símbolo gráfico para comunicar la información.

Orientación de la pantalla [NUEVO 2.1]

El contenido no obliga a poder ser visto y manejado en una única orientación de la pantalla, como en formato vertical o apaisado, a no ser que sea esencial.

Ejemplos de excepciones podrían ser una aplicación de piano, de realidad virtual, o las diapositivas de una presentación. En estos casos, se debe avisar al usuario de que es obligatoria una orientación concreta.

Este criterio responde a las necesidades de los usuarios con la pantalla en posición fija. El objetivo es que el contenido se pueda disfrutar en cualquier orientación, aunque para ello haya que reajustar los controles a la orientación de la pantalla.

Técnicas para cumplir el criterio

- Utiliza CSS para permitir que el contenido se muestre tanto en formato vertical como apaisado.
- Dispón controles de mostrar/ocultar para facilitar el acceso a los contenidos en diferentes orientaciones de pantalla.

¡No lo hagas!

- No bloques la orientación de la pantalla.

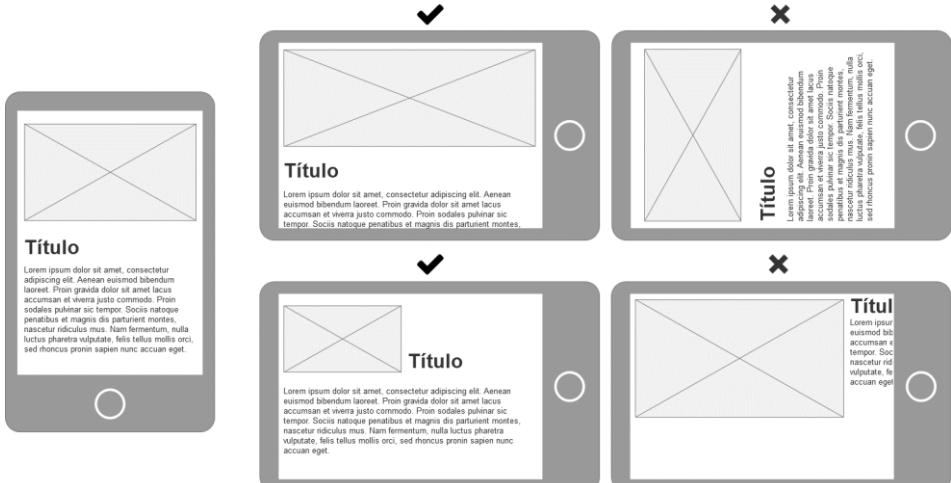


Ilustración 8 Formas correctas e incorrectas de disponer contenidos en formato apaisado

Identificación del propósito del campo [NUEVO 2.1]

El propósito de cada campo que recoge información del usuario puede ser determinado por software cuando:

- el campo tiene un propósito identificado y concreto; y
- el campo está implementado usando tecnologías que permiten identificar su significado esperado en un formulario.

El objetivo de este criterio es ayudar a las personas a reconocer y comprender la finalidad de cada campo de los formularios, permitiendo además su procesamiento automático. Por ejemplo, si en un formulario se nos pide el teléfono, y ya lo hemos introducido en otro sitio distinto anteriormente, si ambos campos han usado el mismo código para identificarse, el navegador puede autocompletar el campo sin necesidad de que lo haga el usuario, reduciendo su esfuerzo y la posibilidad de cometer un error.

El propósito identificado y concreto del campo debe ser uno de los listados por las WCAG 2.1. Se trata de un listado de campos habituales en la identificación de una persona, como nombre, teléfono o tarjeta de crédito. Más adelante mostramos este listado de campos, para qué se debe usar y qué tipo de campo hay que usar.

Este criterio complementa y da un paso más allá del criterio 4.1.2 “Nombre, función y valor”, ofreciendo un valor real directo a la experiencia del usuario.

Técnicas para cumplir el criterio

- Utiliza el atributo *autocomplete* en los campos de formulario con HTML 5.2.

El atributo *autocomplete*

Los agentes de usuario a menudo tienen características que ayudan a los usuarios a llenar formularios basándose en formularios llenados anteriormente, y para ello se formuló el atributo *autocomplete* dentro de HTML5.2.

Existen dos formas de implementar el atributo *autocomplete*:

- Como “**ancla**”, describiendo el significado del valor.
- Como “**expectativa**”, describiendo qué se espera de los usuarios.

En los elementos *input* de tipo *hidden*, el atributo *autocomplete* se comporta como “ancla”. En el resto de los casos, se comporta como “expectativa”.

Cuando se comporta como “expectativa”, el atributo *autocomplete* tiene 3 posibles valores:

- *off* cuando no queremos que el sistema recuerde un campo, por seguridad para el usuario o porque nunca lo va a tener que reusar otra vez.
- *on* cuando queremos que el agente de usuario ofrezca valores, pero sin indicarle de qué tipo, y ya el sistema decidirá por sí mismo.
- Una **cadena de tokens**, además de decir al agente de usuario que ofrezca valores, limitamos qué tipo de valores va a ofrecer. Los *tokens* se deben mostrar en este orden:
 1. Opcionalmente, un *token* que empiece por “section-” para indicar que se agrupa con otros campos.
 2. Opcionalmente, otro *token* con dos posibles valores *shipping* o *billing*, para indicar que ese campo es parte de la dirección de envío o de facturación.
 3. A continuación, una de estas dos opciones:
 - a. Un *token* con uno de los posibles valores listados en las páginas siguientes.
 - b. Dos *tokens* concretos, en el siguiente orden:
 - i. Una indicación de si el contacto es de casa (*home*), del trabajo (*work*), su móvil (*mobile*), un fax (*fax*), o un busca (*pager*).
 - ii. El detalle del email (*email*), del teléfono, (*tel*, entre otras opciones), o de la mensajería instantánea (*impp*).

En el siguiente ejemplo, observa cómo se comporta el atributo *autocomplete* en el último paso de un proceso de compra.

- Para el importe, los dos primeros campos son del tipo *hidden*, y funcionan como anclas, arrastrando la cantidad y la moneda de la transacción de pasos anteriores.
- Para el medio de pago, los tres siguientes campos solicitan información sobre la tarjeta de crédito, su fecha de caducidad y el código de seguridad.

```

<!-- importe -->
<input type=hidden autocomplete=transaction-currency value="EUR">
<input type=hidden autocomplete=transaction-amount value="20.00">

<!-- medio de pago -->
<label for=numero>Tarjeta de crédito</label>
<input id=numero type=text autocomplete=cc-number >

<label for=caducidad>Fecha de caducidad</label>
<input id=caducidad type=month autocomplete=cc-exp >

<label for=codigo>Código de seguridad</label>
<input id=codigo type=text autocomplete=cc-csc >

```

En este otro ejemplo, pedimos al usuario que introduzca la dirección de envío y la dirección de facturación. Observa cómo se concatenan los atributos *autocomplete*:

```

<!-- dirección de envío -->
<fieldset>
<legend>¿Dónde enviamos el paquete?</legend>

<label for=edireccion>Dirección</label>
<input name=edireccion id=edireccion
       autocomplete="shipping street-address" >

<label for=eciudad>Ciudad </label>
<input name=eciudad id=eciudad
       autocomplete="shipping address-level3" >

<label for=ecp>Código postal</label>
<input name=ecp id=ecp autocomplete="shipping postal-code" >

</fieldset>

<!-- dirección de facturación -->
<fieldset>
<legend>¿Dónde enviamos la factura?</legend>

<label for=fdireccion>Dirección</label>
<input name=fcalle id=fdireccion
       autocomplete="billing street-address" >

<label for=fciudad>Ciudad </label>
<input name=fciudad id=fciudad
       autocomplete="billing address-level3" >

<label for=fcp>Código postal</label>
<input name=fcp id=fcp
       autocomplete="billing postal-code" >
</fieldset>
<!-- botón enviar -->
<input type=submit value="Seguir">

```

Puedes ampliar información sobre el atributo *autocomplete* en la especificación de HTML 5.2
<http://www.w3.org/TR/html52/sec-forms.html#sec-autofill>

Tabla 3 Campos con propósitos identificados y concretos

Autocomplete	Descripción	Tipo
name	Nombre completo	<i>text</i>
honorific-prefix	Prefijo honorífico (por ejemplo, <i>Señora, Doctora...</i>)	<i>text</i>
given-name	Nombre	<i>text</i>
additional-name	Segundo nombre	<i>text</i>
family-name	Apellido	<i>text</i>
honorific-suffix	Sufijo honorífico (por ejemplo, <i>Ph.D, IV, Jr....</i>)	<i>text</i>
nickname	Nombre que se muestra en pantalla	<i>text</i>
organization-title	Puesto laboral (por ejemplo, <i>Diseñadora de Interacción</i>)	<i>text</i>
username	Nombre de usuario	<i>text</i>
new-password	Nueva contraseña	<i>password</i>
current-password	Contraseña actual	<i>password</i>
organization	Compañía	<i>text</i>
street-address	Dirección, en un campo de varias líneas	<i>textarea</i>
address-line1	Dirección, primera línea	<i>text</i>
address-line2	Dirección, segunda línea	<i>text</i>
address-line3	Dirección, tercera línea	<i>text</i>
address-level1	En una dirección, la 1 ^a región administrativa más alta (por ejemplo, la comunidad autónoma)	<i>text</i>
address-level2	En una dirección, la 2 ^a región administrativa más alta (por ejemplo, la provincia)	<i>text</i>
address-level3	En una dirección, la 3 ^a región administrativa más alta (por ejemplo, la localidad)	<i>text</i>
address-level4	En una dirección, la 4 ^a región administrativa más alta (por ejemplo, el barrio)	<i>text</i>
country	Código de país	<i>text</i>
country-name	Nombre del país	<i>text</i>
postal-code	Código postal	<i>text</i>
cc-name	Nombre completo del titular del medio de pago	<i>text</i>
cc-given-name	Nombre del titular del medio de pago	<i>text</i>
cc-additional-name	Segundo nombre del titular del medio de pago	<i>text</i>
cc-family-name	Apellido del titular del medio de pago	<i>text</i>
cc-number	Número o código del medio de pago	<i>text</i>
cc-exp	Fecha de caducidad del medio de pago	<i>number</i>
cc-exp-month	Mes de caducidad del medio de pago	<i>text</i>
cc-exp-year	Año de caducidad del medio de pago	<i>text</i>
cc-csc	Código de seguridad (CVC, CVV) del medio de pago	<i>text</i>

Autocomplete	Descripción	Tipo
<i>cc-type</i>	Tipo de medio de pago	<i>text</i>
<i>transaction-currency</i>	Moneda de la transacción	<i>text</i>
<i>transaction-amount</i>	Cantidad de la transacción	<i>number</i>
<i>language</i>	Idioma favorito	<i>text</i>
<i>bday</i>	Fecha de nacimiento	<i>date</i>
<i>bday-day</i>	Día de la fecha de nacimiento	<i>number</i>
<i>bday-month</i>	Mes de la fecha de nacimiento	<i>number</i>
<i>bday-year</i>	Año de la fecha de nacimiento	<i>number</i>
<i>sex</i>	Sexo	<i>text</i>
<i>url</i>	Página web	<i>url</i>
<i>photo</i>	Fotografía	<i>url</i>
<i>tel</i>	Teléfono completo	<i>tel</i>
<i>tel-country-code</i>	Código telefónico del país	<i>text</i>
<i>tel-national</i>	Teléfono sin el código telefónico del país	<i>text</i>
<i>tel-area-code</i>	Código telefónico de la región	<i>text</i>
<i>tel-local</i>	Teléfono sin el código del país ni de la región	<i>text</i>
<i>tel-local-prefix</i>	Prefijo local	<i>text</i>
<i>tel-local-suffix</i>	Sufijo local	<i>text</i>
<i>tel-extension</i>	Extensión telefónica	<i>text</i>
<i>email</i>	Correo electrónico	<i>email</i>
<i>impp</i>	Enlace directo mediante mensajería instantánea	<i>url</i>

Identificación del propósito [NUEVO 2.1]

En contenidos implementados con lenguajes de marcado, el propósito de los componentes de la interfaz de usuario, iconos y regiones pueden ser determinados por programación.

El objetivo de este criterio es mejorar la personalización y las preferencias de usuario. Esto requiere que los diseñadores añadamos contexto, propósitos y el significado de los símbolos, regiones, botones, enlaces y campos.

Técnicas para cumplir el criterio

Agrega semántica a los elementos con las tecnologías COGA⁷, SCHEMA⁸, Microformatos⁹, o con ARIA.

Técnicas para mejorar la usabilidad

- Permite que los agentes de usuario encuentren la versión del contenido que mejor se adapte a sus intereses.
- Utiliza semántica para identificar características importantes con *coga-simplification="simplest"*.
- Utiliza los atributos de *aria-invalid* y *aria-required*.

⁷ COGA es el grupo de trabajo relacionado con las discapacidades cognitivas y de aprendizaje del W3C (*Cognitive and Learning Disabilities Accessibility Task Force*)
<https://github.com/w3c/coga/master/techniques/index.html>

⁸ Schema es una comunidad colaborativa con la misión de crear, mantener y promover esquemas de datos estructurados. <http://schema.org/>

⁹ Los microformatos son pequeños patrones de HTML usados para representar cosas comunes como personas, eventos, blogs, lugares, etc. <http://microformats.org>

Pauta 1.4

Distinguible, separa el fondo del primer plano

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

Esta pauta intenta hacer que la presentación predefinida sea tan fácil de percibir como sea posible. Trata especialmente sobre el contraste, no sólo de los colores, sino que también se aplica al contenido sonoro.

Consejos, a modo general

- Elige fuentes legibles.
- El texto en imágenes debe ser al menos de 14 puntos y tener buen contraste¹⁰.
- Los enlaces y controles deben destacar cuando reciben el foco de teclado.

¹⁰ En esta pauta veremos diferentes ratios de contraste. Puedes usar el analizador de contraste de color sugerido en el capítulo *Recursos y herramientas* para saber más sobre las fórmulas detrás de estas ratios.

Uso del color

El color no se usa como el único medio visual de transmitir la información, indicar una acción, solicitar una respuesta o distinguir un elemento visual.

Técnicas para cumplir el criterio

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

- Comprueba que la información representada por colores está disponible en texto.
- Incluye una indicación de texto para las etiquetas de los controles de formulario coloreadas.
- Comprueba que hay señales visuales adicionales cuando las diferencias de color en el texto transmiten una información.
- Cuando el color indica que el foco se posa en un enlace o en un control, garantiza una ratio de contraste mínimo de 3:1 con el contenido adyacente y ofrece señales visuales adicionales.

Si se comunica información mediante el color dentro de una imagen:

- Diferencia los datos no solo mediante los colores sino también con patrones e iconos de diferente forma.
- Comprueba que la información comunicada por las diferencias de color está también disponible en el texto.

Técnicas para mejorar la usabilidad

- Destaca y redunda la información transmitida en texto con colores.
- Cambia la presentación de los componentes de la interfaz de usuario cuando reciben el foco mediante CSS.

¡No lo hagas!

- No ofrezcas una alternativa textual que no incluya la información comunicada por las diferencias de color en la imagen.
- No crees enlaces que las personas con daltonismo no puedan distinguir.
- No identifiques campos obligatorios o con errores usando sólo diferencias de color.

Control del audio

Si el audio de una página web suena automáticamente durante más de tres segundos, se proporciona un mecanismo para pausar o detener el audio, o un mecanismo para controlar el volumen del sonido que sea independiente del nivel de volumen global del sistema.



Dado que cualquier contenido que no satisfaga este criterio puede interferir con la capacidad del usuario para emplear la página como un todo, **todos los contenidos de la página web deben satisfacer este criterio.**

Técnicas para cumplir el criterio

- Si un sonido se reproduce automáticamente, apágalo también automáticamente antes de tres segundos.
- Ofrece un control al usuario en el inicio de la página web para que pueda apagar los sonidos que suenan automáticamente.
- Reproduce los sonidos sólo a petición del usuario.

Técnicas para mejorar la usabilidad

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

¡No lo hagas!

- No reproduzcas ningún sonido durante más de tres segundos sin un mecanismo para que el usuario pueda apagarlo.

Contraste mínimo

La presentación visual de texto e imágenes de texto tiene una relación de contraste al menos de 4,5:1.

Excepción 1: textos e imágenes de texto con un tamaño de letra grande que tienen una relación de contraste al menos de 3:1.

Excepción 2: no es necesario que cumplan este criterio los textos e imágenes de texto si:

- forman parte de un componente inactivo de la interfaz de usuario,
- son decorativos,
- no se presentan a los usuarios,
- forman parte de una imagen que contiene otros elementos visuales significativos, o
- forman parte de un logotipo o nombre de marca.

Técnicas para cumplir el criterio

La ratio mínima de contraste entre el texto (incluido el texto de las imágenes de texto) y el fondo debe ser:

Si el texto está en negrita:

Menor de 14 puntos	Igual o mayor de 14 puntos
4,5:1	3:1

Si el texto no está en negrita:

Menor de 18 puntos	Igual o mayor de 18 puntos
4,5:1	3:1

En ambos casos:

- Puedes elegir no especificar un color de fondo y de texto, pero no cambies los predefinidos por el agente de usuario.
- Ofrece un mecanismo con suficiente contraste que permita a los usuarios cambiar el contraste.

Técnicas para mejorar la usabilidad

- Emplea una ratio de contraste 3:1 o mayor como presentación predefinida.
- Aumenta el contraste en textos que están sobre un fondo con motivos de diseño, en las líneas de los diagramas y en la combinación de texto rojo sobre fondo negro.
- Ofrece suficiente contraste de color en gráficas, diagramas y campos de texto vacíos.
- Define el estilo del texto con las CSS en lugar de decorar con imágenes de texto.
- Elige componentes centrales del espectro visible para los colores claros; y de los extremos del espectro visible para los colores oscuros.¹¹
- Escoge un fondo de color pastel claro mejor que un fondo blanco detrás de un texto negro para crear un contraste suficiente pero no extremo.
- Dibuja iconos de trazo simple que cumplan las necesidades de contraste del texto.
- Elige una tecnología que permita cambiar el color de primer plano y de fondo de los bloques de texto.

¡No lo hagas!

- No especifiques colores de primer plano sin especificar colores de fondo o viceversa.
- No utilices imágenes de fondo que no ofrezcan suficiente contraste con el texto (o imágenes de texto) del primer plano.

¹¹ Puedes leer más sobre el espectro visible y sus valores en http://es.wikipedia.org/wiki/Espectro_visible.

Cambio de tamaño del texto

Todo el texto puede ser ajustado sin productos de apoyo hasta un 200% sin que se pierdan el contenido o la funcionalidad.

Excepción: Subtítulos y las imágenes de texto

Técnicas para cumplir el criterio

- Elige una tecnología con la cual los agentes de usuario habituales puedan agrandar el contenido, como los navegadores o los lectores de archivos PDF.
- Especifica medidas que sean relativas a otras medidas (tamaños de fuente indicados en porcentajes, en unidades *em* o con nombres) y comprueba que los contenedores de texto redimensionan cuando el texto redimensiona usando:
 - unidades *em* para los contenedores de texto;
 - diseños líquidos; o
 - calculando el tamaño computado y la posición en una forma que escale con el tamaño del texto.
- Ofrece un mecanismo en la página web que permita a los usuarios incrementar el tamaño de todos los textos de la página hasta el 200%.
- Comprueba que no hay pérdida de contenido o funcionalidad cuando el texto redimensiona y los contenedores de texto no.

Técnicas para mejorar la usabilidad

Sobre la presentación:

- Evita el texto justificado.
- Controla la presentación visual del texto mediante estilos de CSS.
- Ofrece suficiente espacio entre líneas y entre columnas.

Sobre el contenido no textual:

- Evita usar texto en imágenes rasterizadas.
- Ofrece diferentes tamaños para el contenido no textual cuando no puede haber una alternativa equivalente accesible.
- Comprueba que el tamaño de texto en imágenes rasterizadas es al menos 18pt.
- Ofrece un mecanismo que permita al usuario agrandar los subtítulos.
- Redimensiona las imágenes de texto con programación de servidor.

Sobre los contenedores:

- Especifica los tamaños de los contenedores mediante los porcentajes de la página.
- Especifica medidas relativas para los anchos de columna de tal modo que quepan una media de 80 caracteres o menos por línea cuando se redimensiona el navegador.

Sobre el texto:

- Ofrece fuentes grandes de forma predefinida.
- Escala el texto hasta el 50% menos y comprueba que se ve bien.
- Evita escalar tamaños de fuente menores que los predefinidos por el agente de usuario.
- Escala también los elementos de formulario que contienen texto.

¡No lo hagas!

Al redimensionar el texto mostrado visualmente hasta el 200%:

- Las imágenes de texto o los controles no pueden quedar cortados, superpuestos u ocultos.
- No te olvides de redimensionar también los controles de formulario de texto.

Imágenes de texto

Se utiliza el texto para transmitir la información en lugar de imágenes de texto si se puede lograr la presentación visual deseada con las tecnologías utilizadas.

Sólo se permite la imagen de texto si es visualmente configurable según los requisitos del usuario; o si la forma particular de presentar el texto resulta **esencial** para la información que se transmite (*por ejemplo, un logotipo*).

Técnicas para cumplir el criterio

- Separa 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.
- Controla la presentación visual del texto mediante CSS.
- Sustituye el texto con imágenes de texto mediante CSS y ofrece un mecanismo al usuario para que revierta el cambio.

Técnicas para mejorar la usabilidad

- Identifica el contenido no textual que transmita información.
- Indica los tamaños de las fuentes en porcentajes, en unidades *em* o con nombres.
- Controlar el espaciado de letras dentro de las palabras mediante CSS.
- Ubica el contenido mediante el marcado de estructura.
- Evita aplicar estilo de texto a caracteres dentro una palabra.

Contraste mejorado

La presentación visual de texto e imágenes de texto tiene una relación de contraste al menos de 7:1.

Excepción 1: textos e imágenes de texto, con un tamaño de letra grande, que tienen una relación de contraste al menos de 4,5:1.

Excepción 2: no es necesario que cumplan este criterio los textos e imágenes de texto si:

- forman parte de un componente inactivo de la interfaz de usuario,
- son decorativos,
- no se presentan a los usuarios,
- forman parte de una imagen que contiene otros elementos visuales significativos, o
- forman parte de un logotipo o nombre de marca.

Técnicas para cumplir el criterio

La ratio mínima de contraste entre el texto (incluido el texto de las imágenes de texto) y el fondo debe ser:

Si el texto está en negrita:

Menor de 14 puntos	Igual o mayor de 14 puntos
7:1	4,5:1

Si el texto no está en negrita:

Menor de 18 puntos	Igual o mayor de 18 puntos
7:1	4,5:1

En ambos casos:

- Si no especificas el color de fondo o color de texto, no cambies los predefinidos.
- Ofrece un mecanismo con suficiente contraste para permitir a los usuarios cambiar a una presentación con suficiente contraste.



Además, sigue las mismas técnicas para mejorar la usabilidad y evita los mismos errores que en el Criterio de Conformidad 1.4.3.

Sonido de fondo bajo o ausente

Para el contenido de solo audio grabado que

- contiene habla en primer plano,
- no es un CAPTCHA sonoro o un logo sonoro, y
- no es una vocalización cuya intención principal es servir como expresión musical (como el canto o el rap),

se cumple al menos uno de los siguientes casos:

- no hay sonido de fondo,
- los sonidos de fondo se pueden apagar, o
- los sonidos de fondo son 20 decibelios más bajos que las voces de primer plano (excepto sonidos ocasionales que duren solamente 1 o 2 segundos).

Técnicas para cumplir el criterio

- Mezcla los archivos de audio para que los sonidos no procedentes de una voz estén por lo menos 20 decibelios por debajo del contenido sonoro de la voz.

Técnicas para mejorar la usabilidad

- Ofrece una manera para que los usuarios puedan ajustar los niveles auditivos del sonido de primer plano y del sonido de fondo independientemente.



La diferencia de 20 decibelios equivale aproximadamente a que los sonidos de fondo tienen un volumen percibido de 4 veces más silencioso que las voces de primer plano.

En el capítulo *Recursos y herramientas* puedes encontrar una herramienta y un tutorial para comprobar el contraste de sonidos.

Presentación visual

En la presentación visual de bloques de texto, se proporciona un mecanismo para que:

- los colores de fondo y primer plano puedan ser elegidos por el usuario.
- el ancho sea menor o igual a 80 caracteres o signos (40 si es chino, japonés o coreano).
- el texto no esté justificado (alineado a los márgenes izquierdo y derecho a la vez).
- el espaciado entre líneas (interlineado) sea al menos de un espacio y medio dentro de los párrafos.
- el espacio entre párrafos sea al menos 1,5 veces mayor que el interlineado.
- el texto pueda ser redimensionado hasta un 200% sin ayuda de los productos de apoyo y sin que necesite de un *scroll* horizontal para leer una línea de texto en una ventana a pantalla completa.

Técnicas para cumplir el criterio

Para que el usuario pueda cambiar los colores de fondo y primer plano, elige una de las siguientes opciones:

- Especifica los colores de fondo y de texto de contenidos secundarios (*banners*, menús de navegación...) si no has especificado colores de fondo y texto del contenido principal.
- Delimita las áreas de la página web con los bordes y el diseño si no has especificado los colores de texto y de fondo.
- Elige una tecnología que permita cambiar el color del primer plano y de fondo.
- Mantén los colores de texto y de fondo predefinidos por el navegador.
- Ofrece en la página un mecanismo de selección de colores que permita al usuario elegir los colores de fondo y de primer plano.

Para que el ancho sea menor o igual a 80 caracteres o signos (40 si es chino, japonés o coreano), elige una de las siguientes opciones:

- Deja que el agente de usuario gestione el reposicionamiento del texto cuando se estrecha la ventana.
- Especifica medidas relativas para los anchos de columna para que queden una media de 80 caracteres o menos por línea cuando se redimensiona el navegador.

Para que el texto no esté justificado (alineado a los márgenes izquierdo y derecho a la vez), elige una de las siguientes opciones:

- Especifica la alineación del texto a la izquierda o a la derecha.
- Ofrece un mecanismo para quitar la justificación completa de texto.
- Alinea el texto en un único lado.

Para que el espaciado de línea (interlineado) sea al menos de un espacio y medio dentro de los párrafos; y el espacio entre párrafos sea al menos 1,5 veces mayor que el interlineado, elige una de las siguientes opciones:

- Especifica el espaciado de línea.
- Ofrece un mecanismo en la página que permita al usuario incrementar el espaciado de línea y de párrafo.

Para que el texto pueda ser redimensionado hasta un 200% sin ayuda de los productos de apoyo y sin que necesite de un *scroll* horizontal para leer una línea de texto en una ventana a pantalla completa, elige una de las siguientes opciones:

- Deja que el agente de usuario gestione el reposicionamiento del texto cuando se estrecha la ventana.
- Utiliza un diseño líquido y medidas relativas con una de estas técnicas:
 - indicando los tamaños de las fuentes en porcentajes, en unidades *em* o con nombres;
 - indicando los tamaños de los contenedores en porcentajes; o
 - calculando el tamaño y la ubicación de los elementos cuando se escala el tamaño de texto.
- Ofrece opciones dentro del contenido para cambiar el diseño sin que requieran hacer *scroll* horizontal para leer el contenido.

Técnicas para mejorar la usabilidad

Sobre los tamaños:

- Ofrece fuentes grandes de forma predefinida.
- Deja un espacio *em* o dos espacios después de cada oración.
- Evita escalar tamaños de fuente más pequeños que los predefinidos en el agente de usuario.
- Ofrece suficiente espaciado entre columnas.

Sobre la presentación:

- Presenta el texto en una fuente de palo seco (*sans serif*) o bien ofrece un mecanismo para que el usuario pueda cambiarla.
- Evita grandes trozos de texto en itálica.
- Distingue visualmente los enlaces.
- Evita el alineado central de texto.
- Evita abusar de diferentes estilos en las páginas.
- Evita el texto en imágenes rasterizadas.
- Destaca los párrafos, las listas de elementos o las celdas de tabla con el efecto *hover*.
- Utiliza las mayúsculas y las minúsculas según las convenciones del idioma.

Sobre las listas:

- Usa listas verticales (con puntos o numeradas) mejor que listas dispuestas en una línea.
- Ofrece viñetas en las listas que sean ampliables.
- Ofrece un mecanismo para mostrar y ocultar las viñetas.

¡No lo hagas!

- No especifiques colores de primer plano sin especificar colores de fondo o viceversa.
- No justifiques el texto (alineado a los márgenes izquierdo y derecho a la vez).

Imágenes de texto (sin excepciones)

Las imágenes de texto sólo se utilizan como simple decoración o cuando una forma de presentación particular del texto resulta esencial para la información transmitida.

Los logotipos (textos que son parte de un logo o de un nombre de marca) se consideran contenidos esenciales.

Técnicas para cumplir el criterio

- Separa 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.
- Controla la presentación visual del texto mediante CSS.
- Sustituye el texto con imágenes de texto mediante CSS y ofrece un mecanismo al usuario para que pueda revertir el cambio

Técnicas para mejorar la usabilidad

- Redimensiona las imágenes de texto con programación de lado del servidor.
- Indica los tamaños de las fuentes en porcentajes, en unidades *em* o con nombres.
- Controla el espaciado de letras dentro de las palabras mediante CSS.
- Ubica el contenido mediante el marcado de estructura.
- Evita aplicar estilo de texto a caracteres dentro una palabra.

Reajuste de elementos [NUEVO 2.1]

El contenido se puede presentar sin perder información o funcionalidad, y sin requerir *scroll* en dos dimensiones para:

- Contenido que se desplaza en vertical con una anchura equivalente a 320 píxeles CSS.
- Contenido que se desplaza en horizontal con una altura equivalente a 256 píxeles CSS.

Excepción: aquellas partes del contenido que requieren ese desplazamiento en dos dimensiones por su uso o significado.

El objetivo de este criterio es ayudar a las personas con baja visión que hacen zoom en el navegador para aumentar el tamaño del contenido al 400%, sin que ello les obligue a desplazarse en dos direcciones. Con el 400% de zoom, 320 píxeles CSS equivalen a un ancho de pantalla inicial (*viewport width*) de 1280 píxeles, y 256 píxeles CSS equivalen a 1024 píxeles.

Ejemplos de contenidos que requieren desplazamiento en dos dimensiones, y que entran dentro de las excepciones, son las imágenes grandes, mapas, videojuegos, tablas de datos...

Técnicas para cumplir el criterio

- Da opciones dentro del contenido para cambiar a un diseño que no requiera *scroll* para leer una línea de texto.
- Utiliza *media queries* en CSS y
- Reajusta los contenidos mediante CSS usando *grids* o Flexbox; o
- Calcula por programación tamaños y posiciones de los elementos de forma que escalen con el tamaño del texto.

Técnicas para mejorar la usabilidad

- Con CSS, ajusta las imágenes al tamaño de pantalla (*viewport*).
- Con CSS, reajusta las tablas de datos simples y haz que las celdas quepan en la anchura de la pantalla (*viewport*).
- Ofrece un mecanismo para cambiar a la vista móvil en cualquier momento.

¡No lo hagas!

- No uses medidas fijas para el tamaño de los contenedores o la posición del contenido.
- No interfieras en la capacidad del navegador de hacer zoom.
- No uses texto pre-formateado o no lo consideres una excepción a la hora de tener un doble *scroll* en la página.

Contraste no textual [NUEVO 2.1]

La presentación visual de los siguientes elementos tiene una ratio de contraste de la menos 3:1 con los colores adyacentes:

- **Componentes de interfaz de usuario:** información visual usada para indicar estados y límites del interfaz de usuario.
- **Objetos gráficos:** partes de los gráficos necesarias para comprender el contenido, excepto cuando una presentación concreta sea esencial para la información mostrada.

Los botones, campos, iconos, gráficas (y sus diferentes partes), etc. tanto en sus estados normales, *hover* o *focus*, necesitan poder ser percibidos, comprendidos y diferenciados claramente por todas las personas. Por ejemplo, un campo de formulario con una combinación de tonalidades que contrastan menos de 3:1 será difícil de percibir, y por lo tanto de utilizar.



Ilustración 9 Formulario con insuficiente contraste (izquierda) y con suficiente contraste (derecha)

No tienen la obligación de pasar esta ratio de contraste: los campos deshabilitados, los logotipos, las banderas, los gradientes de colores, los dibujos naturalistas o las fotografías.

Fíjate en las siguientes gráficas. Las dos representan la misma información, pero en la primera los sectores A y B tienen tonos muy parecidos, y son difíciles de diferenciar. En la segunda gráfica los sectores se pueden diferenciar claramente gracias a las líneas gruesas blancas y negras entre sectores.

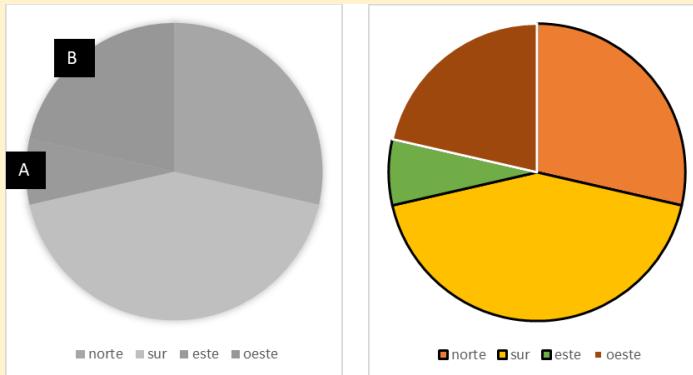


Ilustración 10 Gráficas sin suficiente contraste (izquierda) y con suficiente contraste (derecha)

Además, según el criterio “1.4.1 Uso del color”, es obligatorio que la información no se transmita solo por el color y, por lo tanto, se debe incluir también la etiqueta asociada a cada zona de la gráfica y no solo mediante una leyenda.

Otra forma de presentar la información sería con zonas de gráficas contiguas, pero con colores suficientemente contrastados entre sí, por lo que no serían necesarias las líneas de separación.



Ilustración 11 Gráfica con colores yuxtapuestos suficientemente contrastados y valores asociados

Técnicas para cumplir el criterio

Gráficas con suficiente contraste

- Contrasta los colores entre los objetos de las gráficas.
- Incluye líneas con suficiente contraste entre colores adyacentes.

Texto sobre gráficos

- Incluye etiquetas y valores con el gráfico.
- Asegúrate de que hay una ratio de contraste de al menos 3:1 entre el texto (incluido el texto de las imágenes de texto) y su fondo.
- Ofrece un control con suficiente contraste que permita a los usuarios cambiar a una presentación que utilice suficiente contraste.
- Usa suficiente contraste en las imágenes que transmiten información.

Técnicas para mejorar la usabilidad

Usa una ratio de contraste de 4,5:1 con el texto de alrededor y provee de pistas visuales adicionales en el estado *focus* para los enlaces o controles donde sólo se usa el color para identificarlos.

Espaciado del texto [NUEVO 2.1]

En el contenido implementado con un lenguaje de marcado que permita que el contenido se pueda formatear de la siguiente manera, no se pierde contenido ni funcionalidad si se cambian estas propiedades, sin necesidad de tocar otros estilos:

- **Espacio entre líneas:** al menos 1.5 veces el tamaño de la fuente.
- **Espacio entre párrafos:** al menos 2 veces el tamaño de la fuente.
- **Espacio entre letras:** al menos 0.12 veces el tamaño de la fuente.
- **Espacio entre palabras:** al menos 0.16 veces el tamaño de la fuente.

Excepciones: los lenguajes humanos y programaciones que no utilizan alguna de estas propiedades pueden alcanzar la conformidad usando sólo las propiedades que existan para esa combinación de lenguaje y programación.

El objetivo de este criterio es ayudar a las personas con dislexia o con baja visión a que puedan percibir y leer mejor los textos, pues asegura que la distancia entre líneas, párrafos, palabras o caracteres seguirá siendo adecuada para la lectura, aunque se modifique la presentación.

Los usuarios también pueden usar sus propias hojas de estilo, por lo que debemos diseñar de tal modo que, con las medidas establecidas en el criterio, ningún texto quede cortado, superpuesto u oculto.

Este criterio no nos obliga a usar un espaciado concreto (como sí sucede en el criterio 1.4.8), sino que indica que, si el usuario configura el espaciado de la manera que se especifica, no se pierde contenido ni funcionalidad.

En el capítulo *Recursos y Herramientas* encontrarás una forma de revisar este criterio.

Técnicas para cumplir el criterio

- Permite que el espaciado de tu texto se pueda cambiar.

Técnicas para mejorar la usabilidad

- Define el espacio entre letras con la propiedad *letter-spacing* en CSS.
- Define el espacio entre líneas con la propiedad *line-height* en CSS.
- Define el tamaño de los contenedores de texto usando unidades *em*.

¡No lo hagas!

- No uses medidas fijas para el tamaño de los textos.

Contenido en *hover* o *focus* [NUEVO 2.1]

Cuando un componente de la interfaz coge y luego pierde el foco, y esto genera un contenido que se hace visible y luego se oculta:

- **El nuevo contenido es descartable** sin mover el puntero o sin poner el foco sobre el mismo (por ejemplo, con la tecla ESC); a no ser que el contenido comunique un error de entrada de datos o no tape o no reemplace a otro contenido.
- **Se puede hacer *hover*:** si el puntero del cursor puede activar el contenido adicional, entonces el puntero se puede mover sobre él sin que desaparezca.
- **El nuevo contenido es persistente**, es decir, sigue visible hasta que el *hover* o el *focus* cambian, el usuario lo descarta o su información ya no es válida.

Excepto: la presentación visual del contenido adicional está controlada por el agente de usuario y no la modifica el diseño, por ejemplo, el *tooltip* generado por el atributo *title*.

Ejemplos de estos contenidos nuevos generados en el *hover* o *focus* de los componentes son los submenús desplegables o las *tooltips* personalizadas.

El objetivo de este criterio es ayudar a las personas que han activado la interacción por error; a las que no se han dado cuenta de que ha aparecido un nuevo contenido; o a las que el nuevo contenido les interfiere en su capacidad de realizar una tarea. Con este criterio nos aseguramos de que los usuarios pueden percibir el nuevo contenido y que lo pueden descartar sin problemas.

Técnicas para cumplir el criterio

- Usa el *role=tooltip* de ARIA
- Usa las pseudo-clases *hover* y *focus* en CSS

¡No lo hagas!

- No escondas los elementos que desencadenan los cambios.
- No escondas los nuevos mensajes cuando se haga *hover* sobre ellos.
- No escondas los nuevos mensajes hasta que el usuario los descarte o sean incorrectos.

Principio 2. Operable

Los diseñadores web debemos ser conscientes de los diferentes dispositivos, periféricos y productos de apoyo que los usuarios utilizan para acceder a nuestros sitios web, por lo que debemos hacer los componentes de la interfaz de usuario y los elementos de navegación de tal forma que todos los usuarios puedan manejarlos.



Pauta 2.1

Manejable por teclado, no olvides a los usuarios sin ratón

Proporciona acceso a todas las funcionalidades mediante el teclado.

Cuando una funcionalidad puede ser manejada usando sólo el teclado, también se podrá manejar utilizando otro sistema de entrada, como un ratón. Pero, si sólo diseñas tu página web para ser usada con el ratón, o con la voz, los productos de apoyo y algunos usuarios encontrarán dificultades para manejar tu página.

Además, las WCAG animan a los desarrolladores a proporcionar métodos adicionales de introducción de datos aparte del teclado y del ratón, como la interacción por voz.

Teclado

Se puede manejar toda la funcionalidad del contenido a través de una interfaz de teclado sin necesidad de alcanzar una determinada velocidad para cada pulsación de las teclas, excepto cuando la función requiera que la introducción de datos dependa del trayecto de los movimientos del usuario y no solo de los puntos inicial y final.

La mayoría de las acciones que se realizan con un puntero pueden hacerse también con el teclado. Sin embargo, algunas sólo se pueden efectuar con punteros, como dibujar a mano alzada. Por el contrario, dibujar líneas rectas o formas geométricas, seleccionar, escalar o arrastrar objetos, sí se pueden realizar sólo con teclado, y no entrarían dentro de la excepción.

Este criterio aplica no solo a los teclados físicos, sino también a teclados que se muestran en pantalla o teclados virtuales proyectados sobre superficies u otros que se inventen en el futuro.

Técnicas para cumplir el criterio

- Identifica todas las funcionalidades del contenido, controles de formulario y enlaces; y comprueba que pueden ser manejados sólo con el teclado.
- Ofrece manejadores de eventos de teclado usando una de las siguientes opciones:
 - incluyendo funciones específicas de teclado y de otro dispositivo.
 - haciendo que el evento *onclick* en enlaces y botones sea accesible por teclado.
 - haciendo redundantes los eventos de ratón y teclado.

Técnicas para mejorar la usabilidad

- Emplea los atributos de rol, estados y valores de (X)HTML si quieres que los elementos estáticos de la interfaz de usuario se comporten como componentes interactivos; y añade acciones accesibles por teclado para elementos estáticos.
- Ofrece atajos de teclado para acceder a los enlaces y controles más importantes.
- Comienza cada elemento de una lista con una combinación de letras diferenciadora.
- Cuando utilices programación de lado del cliente, elige el manejador de eventos más abstracto posible, preferentemente el evento *onactivate*.
- Si vas a utilizar combinaciones de teclas para efectuar acciones, elige combinaciones de teclas que no sean las específicas que utilizan los agentes de usuario para otros objetivos.

¡No lo hagas!

- No utilices sólo manejadores de eventos específicos de punteros como *onmousedown*.
- No quites el foco de un control de formulario o de un enlace cuando lo reciba.
- No emules enlaces con eventos, pues no son reconocibles por programación.

Sin trampas para el foco del 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 sólo el teclado y, si se requiere algo más que las teclas de dirección o de tabulación, se informa al usuario del método apropiado para mover el foco.



Dado que cualquier contenido que no satisfaga este criterio puede interferir con la capacidad del usuario para emplear la página como un todo, **todos los contenidos de la página web deben satisfacer este criterio.**

Técnicas para cumplir el criterio

- Comprueba que los usuarios no quedan atrapados en el contenido.

¡No lo hagas!

- No combines varios formatos de contenido de tal forma que los usuarios puedan quedar atrapados dentro un tipo de formato.

Teclado (sin excepciones)

Toda la funcionalidad del contenido se puede manejar a través de una interfaz de teclado sin requerir una determinada velocidad en la pulsación de las teclas.

Para cumplir este criterio, sigue las técnicas que se citan en el Criterio de Conformidad 2.1.1, simplemente ten en cuenta que no se permite ninguna excepción.

Si es necesario u obligatorio utilizar una entrada que dependa del trayecto de los movimientos del usuario y no solo de los puntos inicial y final, como un dibujo a mano alzada, entonces no será posible cumplir este criterio.

Atajos de teclado [NUEVO 2.1]

Si se implementan atajos de teclado usando una sola letra, un número, un signo de puntuación o un símbolo, al menos se puede hacer una de las siguientes acciones:

- Desactivar el atajo de teclado.
- Reasignar el atajo a uno o más caracteres de teclado no imprimible (Control, *Alt*, etc.).
- Activar el atajo sólo cuando el foco está en el componente que lo controla.

Los usuarios que interactúan con la voz a menudo utilizan caracteres únicos para usar el navegador. Si en nuestro sitio web implementamos atajos de teclado que sean de un carácter único, interferirán con los atajos de su programa de reconocimiento de voz.

Además, los usuarios de teclado con problemas de movilidad también pueden tocar accidentalmente una tecla, y activar funcionalidades de forma inesperada.

Este criterio no se aplica a la implementación de atajos de teclado complejos (por ejemplo, *Alt+F*), ni al atributo *accesskey* (que se maneja de otra manera), ni al manejo de *drop-downs* (ya que sus teclas solo funcionan cuando está el foco en ese *drop-down*).

Técnicas para cumplir el criterio

- Los usuarios tienen una forma de desactivar los atajos de teclado de un solo carácter.
- Existe un mecanismo que permite a los usuarios modificar los atajos de teclado y reasignarlos a caracteres no imprimibles. Los atajos alternativos pueden ser cadenas de hasta 25 caracteres, que servirían como comandos de voz.

¡No lo hagas!

- No implementes atajos de teclado de un solo carácter sin una forma de desactivarlos o de reasignarlos a una combinación de teclas.

Pauta 2.2

Da **tiempo suficiente**, no metas prisa

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

Imagina que estás leyendo una página web y, en medio del proceso, la página se recarga automáticamente y cambia, por lo que no puedes terminar de leer lo que estabas leyendo. Molesto cuando menos, ¿no? Todos necesitamos un tiempo diferente para completar las tareas, y este tiempo normalmente es mayor para ancianos y personas con diversidad funcional.

Esta pauta trata de eliminar los límites de tiempo o de ofrecer el tiempo suficiente para acceder al contenido o para completar las tareas.

Tiempo ajustable

Para los límites de tiempo impuestos por el contenido, se cumple al menos uno de los siguientes casos:

- 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).
- Se advierte 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 la barra espaciadora).

Excepciones:

- 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.
- El límite de tiempo es esencial y, si se extendiera, invalidaría la actividad.
- El límite de tiempo es mayor a 20 horas.

Este Criterio de Conformidad ayuda a que los usuarios puedan completar una tarea sin que haya cambios inesperados en el contenido o en el contexto que sean el resultado de un límite de tiempo.

Este Criterio de Conformidad está relacionado con el Criterio de Conformidad 3.2.1, que limita los cambios de contenido o contexto como resultado de una acción del usuario, y con el 2.2.6 que trata las consecuencias de la inactividad del usuario.

Técnicas para cumplir el criterio

Si hay un límite de tiempo de sesión:

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

Si el límite de tiempo es controlado por la programación de la página:

- Ofrece un mecanismo que permita a los usuarios anular el límite de tiempo.
- Ofrece un medio para aumentar el límite de tiempo 10 veces al límite predefinido.
- Avisa al usuario cuando el límite de tiempo esté próximo a expirar; y permite que el usuario pueda aumentar el límite predefinido.

Si hay un límite de tiempo para leer los contenidos:

- Permite al usuario pausar el contenido y volver a reproducirlo desde donde fue parado.
- Ofrece una manera para que el usuario quite el límite de tiempo.

- Cuando el contenido se desplaza de forma automática mediante programación, se ofrece un mecanismo para pausarlo.
- Ofrece un mecanismo que permita a los usuarios mostrar en una ventana separada, o en un área más grande de la página, aquellos textos que se mueven o se actualizan solos.

Técnicas para mejorar la usabilidad

- Usa programación para preguntar al servidor si existe un límite de tiempo; y notifícaselo al usuario en el caso de que exista.
- Usa sonidos para enfocar la atención del usuario.

¡No lo hagas!

- No utilices las etiquetas *meta redirect* o *refresh* para refrescar la página o redirigir a otra página automáticamente después de un límite de tiempo.
- No utilices técnicas de servidor para redirigir páginas automáticamente después de un límite de tiempo.

Poner en pausa, detener, ocultar

Para la información que se mueve, parpadea¹² o se desplaza, si comienza automáticamente, dura más de 5 segundos y se presenta en paralelo con otro contenido, existe un mecanismo para que el usuario la pueda poner en pausa, detener u ocultar, excepto si el movimiento, parpadeo o desplazamiento es una parte esencial de una actividad.

Para la información que se actualiza automáticamente, si comienza automáticamente y se presenta en paralelo con otro contenido, existe un mecanismo para que el usuario la 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.



Dado que cualquier contenido que no satisfaga este criterio puede interferir con la capacidad del usuario para emplear la página como un todo, **todos los contenidos de la página web deben satisfacer este criterio.**

Si la información se actualiza automáticamente (por programación o porque se recibe por *streaming*) e incluimos la opción de pausa, no es obligatorio preservar o presentar la información que sucedió entre el inicio de una pausa y el reinicio de la presentación, eso podría ser técnicamente imposible o podría mandar un mensaje erróneo o engañoso.

Por ejemplo, un partido en directo en *streaming* que el usuario pausa en un momento concreto. Cuando lo vuelve a reiniciar, al ser en directo, se volvería a conectar en directo con el partido. Esto no impide que se ofrezca la posibilidad de reiniciar desde donde se pausó, si la tecnología utilizada lo permite y se avisa claramente.

Por otro lado, imagina una animación que indica la precarga de un contenido: si el usuario pudiera pararla o cambiar su frecuencia, se podría mandar un mensaje engañoso. En ese caso la información actualizada se consideraría esencial y se aplicaría la excepción del criterio.

Técnicas para cumplir el criterio

- Permite al usuario pausar el contenido y recomenzar desde donde fue pausado.
- Si desplazas el contenido de forma automática mediante programación, ofrece un mecanismo para pausarlo.
- Crea contenido que parpadee menos de 5 segundos.
- Usa tecnologías que permitan que el usuario puede apagar el parpadeo del contenido desde su agente de usuario.
- Diseña los *gifs* animados para que paren después de n ciclos reproducidos en un total de 5 segundos.
- Controla el parpadeo con programación y páralo en 5 segundos o menos.

¹² Puedes encontrar la definición de *parpadeo* en la pauta 2.3.

- Ofrece un mecanismo en la página que pare el movimiento, el parpadeo o el refresco del contenido.
- Ofrece un mecanismo para recargar la página sin contenidos parpadeantes.

Técnicas para mejorar la usabilidad

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

¡No lo hagas!

- No incluyas contenido que se desplace si ese movimiento no es esencial para la actividad sin incluir también un mecanismo para pausar y recomenzar el contenido.
- No utilices el elemento *blink*.
- No crees contenido parpadeante (con la propiedad *text-decoration: blink* en CSS, con programación, o dentro de un *object* o *applet*) sin ofrecer también un mecanismo para que el usuario pueda detenerlo a los 5 segundos o menos.

Sin tiempo

El tiempo no es parte esencial del evento o actividad presentada por el contenido, exceptuando los medios sincronizados no interactivos y los eventos en tiempo real.

Técnicas para cumplir el criterio

Permite a los usuarios que puedan completar una actividad sin ningún límite de tiempo.

Interrupciones

El usuario puede postergar o suprimir las interrupciones, excepto cuando las interrupciones implican una emergencia.

El objetivo general es no interrumpir al usuario en su trabajo, a menos que necesitemos advertir al usuario de situaciones que puedan perjudicar su salud, su seguridad o su dinero. Por ejemplo, cuando el usuario vaya a efectuar una operación que implique el borrado de datos, como se tratará también en los criterios 3.3.4 y 3.3.6.

Técnicas para cumplir el criterio

- Programa que las alertas no esenciales sean opcionales.
- Ofrece un mecanismo para posponer cualquier actualización de contenido.
- Ofrece un mecanismo para solicitar una actualización de contenido en lugar de actualizarlo automáticamente.

¡No lo hagas!

- No utilices las etiquetas *meta redirect* o *refresh* para refrescar la página o redirigir a otra página automáticamente después de un límite de tiempo.

Re-autentificación

Cuando expira una sesión autenticada, el usuario puede continuar su actividad sin perder los datos tras volver a identificarse.

Técnicas para cumplir el criterio

Ofrece la opción de continuar la actividad sin pérdida de datos:

- guardando los datos (el usuario puede retomarlos tras la re-autentificación); o
- codificando los datos del usuario como datos ocultos o cifrados en la página de re-autentificación.

¡No lo hagas!

No limites el tiempo de las sesiones sin ofrecer un mecanismo que guarde las entradas de usuario y que restablezca la información después de la re-autentificación.

Límites de tiempo [NUEVO 2.1]

Se avisa a los usuarios que su inactividad durante un tiempo determinado puede causar una pérdida de datos, a menos que los datos se conserven durante más de 20 horas sin que el usuario haga nada.

Los formularios muy complejos, como la compra de un billete de avión o la reserva de una habitación de hotel, pueden ser agobiantes para personas con limitaciones cognitivas, que necesitan hacer pausas para tomar decisiones o traducir textos. Si en ese momento de pausa los datos se pierden, el usuario debería empezar de nuevo el proceso, con la consiguiente carga mental añadida.

Por eso es importante avisar de forma correcta de los límites de tiempo, y de ser posible, añadir mecanismos para ajustar, extender o cancelar estos límites.

Las regulaciones de privacidad pueden requerir el consentimiento expreso de los usuarios antes de su identificación y antes de que los datos sean guardados, por ejemplo, con la normativa RGPD europea. El propio W3C recomienda consultar con un especialista en derecho la aplicación de este criterio en cada país.

Técnicas para cumplir el criterio

- Expira la sesión cuando pasen al menos 20 horas de inactividad.
- Guarda los datos del usuario durante más de 20 horas.
- Avisa de la duración de la inactividad al comienzo del proceso.

Pauta 2.3

No causes reacciones físicas y psíquicas negativas

No diseñes contenido de un modo que se sepa que puede provocar mareos, vértigos, ataques, espasmos o convulsiones.

Las personas que sufren epilepsia fotosensitiva reaccionan ante cambios bruscos de luminosidad. La velocidad más alta permitida para mostrar contenido *destelleante* es de tres destellos por segundo. Sin embargo, hay personas que reaccionan a velocidades menores, por lo que la recomendación es eliminar este tipo de contenido.

Para delimitar el concepto de *destello*, es necesario recuperar el concepto de *parpadeo*, que aparecía en el criterio 2.2.2:

- **Parpadeo:** alternancia entre dos estados visuales para llamar la atención. Los parpadeos distraen a algunos usuarios, y, como vimos en el criterio 2.2.2, solo se permiten si duran menos de 5 segundos o pueden detenerse.
- **Destello:** cambios de luminosidad relativa (cambios bruscos de luz a oscuridad) que pueden causar convulsiones, mareos y dolores de cabeza en algunas personas si superan cierto tamaño o se encuentran en un determinado rango de frecuencia. Los destellos, salvo que cumplan ciertos requisitos, no se permiten ni un segundo porque podrían provocar el daño antes de que el usuario pudiera evitarlos, o incluso haber muchos usuarios que no saben que son sensibles a los destellos.

¿Cómo saber si el cambio en la luminosidad del contenido es más rápido de 3 veces por segundo? Puedes utilizar la herramienta descrita en el capítulo *Recursos y Herramientas*.

Umbral de tres destellos o menos

Las páginas web 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.



Dado que cualquier contenido que no satisfaga este criterio puede interferir con la capacidad del usuario para emplear la página como un todo, **todos los contenidos de la página web deben satisfacer este criterio.**

Un **destello general** se define como un par de incrementos seguidos por decrementos (o viceversa) del 10% o más en la luminosidad relativa máxima y donde la luminosidad relativa en la imagen más oscura es menor de 0,80.

Un **destello rojo** implica un par de incrementos seguidos por decrementos (o viceversa) que involucran un rojo saturado.

En el capítulo *Recursos y Herramientas* te presentamos PEAT, que te ayudará a comprobar si tu contenido viola estos umbrales.

Técnicas para cumplir el criterio

- Comprueba que ningún componente del contenido destellea más de 3 veces por segundo.
- Haz que el contenido que destella ocupe un espacio pequeño. El área segura para una resolución de 1024x768 a una distancia de visión de entre 22 y 26 pulgadas, es cualquier forma con un área inferior a 21824 píxeles, o lo que es lo mismo, un cuadrado aproximado de 148 píxeles de lado.
- Asegura que el contenido no viola el umbral del destello general o el umbral de destello rojo.

Técnicas para mejorar la usabilidad

- Reduce el contraste de cualquier contenido que destella.
- Evita los rojos totalmente saturados en cualquier contenido que destella.
- Reduce el número de destellos incluso si éstos no violan los umbrales.
- Ofrece un mecanismo para suprimir contenido que destella antes de que empiece.
- Reduce la velocidad de los vídeos en directo para evitar destellos rápidos (como los flashes de fotografía).
- Si se detectan tres destellos dentro un segundo, congela la imagen momentáneamente; o reduce la ratio de contraste.
- Permite a los usuarios establecer un límite de velocidad personalizado.

Tres destellos

Las páginas web no contienen nada que destellee más de 3 veces por segundo.

Técnicas para cumplir el criterio

Comprueba que ningún componente del contenido destellea más de 3 veces por segundo.



Además, sigue las técnicas para mejorar la usabilidad del Criterio de Conformidad 2.3.1

Animaciones desde interacciones [NUEVO 2.1]

Los movimientos animados desencadenados por una interacción pueden ser deshabilitados, a menos que la animación sea esencial para la funcionalidad o para la información que se transmite.

Por **movimientos animados** se entiende la adición de pasos entre posiciones o tamaños distintos para crear la ilusión de movimiento o de transición suave. Sin embargo, los movimientos animados no incluyen cambios de color, de nitidez o de transparencia (para ello están los criterios 2.3.1 y 2.3.2 relativos a destellos).

Un ejemplo de animaciones que pueden producir problemas son los “efectos *parallax*” donde las capas se deslizan a diferente velocidad a medida que el usuario sube o baja la página. Otro ejemplo son las transiciones de fantasía entre páginas, o un zoom rápido en los elementos en los que se pulsa.

El objetivo de este criterio es ayudar a las personas que sufren mareos, vértigos u otros trastornos vestibulares. Una decisión “segura” sería eliminar las animaciones, pero es una decisión demasiado radical y no siempre es la mejor opción. Las animaciones, cuando se usan con criterio, pueden ser una buena manera de comunicar la relación entre diferentes partes, atraer la atención del usuario o simplemente añadir algo de diversión a la interacción.

Técnicas para cumplir el criterio

- Permite a los usuarios definir una preferencia que impida la animación.
- Utiliza la *media query* `{prefers-reduce-motion}` para evitar las animaciones en base a la configuración de accesibilidad del agente de usuario.

Pauta 2.4

Céntrate en la Arquitectura de Información para que sea fácilmente **navegable**

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

Encontrar el contenido y saber dónde se encuentran pueden ser tareas difíciles para las personas con discapacidad, especialmente para los usuarios que acceden con lectores de pantalla o usuarios con discapacidad cognitiva.

Consejos, a modo general

- Limita el número de enlaces por página.
- Ofrece mecanismos que permitan navegar por las diferentes secciones de la página.
- Distingue visualmente los enlaces.
- Resalta los términos de búsqueda.

Evitar bloques

Existe un mecanismo que permite al usuario evitar los bloques de contenido que se repiten en múltiples páginas web.

Técnicas para cumplir el criterio

Evita 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.

Evita al usuario tener que pasar por bloques de contenido o elementos que no le interesan con una de estas opciones:

- Identificar las regiones de la página mediante *landmarks roles* de ARIA
- Elementos de encabezado al inicio de cada sección de contenido.
- Elementos *map* en HTML.
- Elementos *iframe* y *frame* identificados con el atributo *title*.
- Menús expansibles y contraíbles.

Técnicas para mejorar la usabilidad

- Ofrece acceso mediante teclado a los enlaces y los controles de formulario importantes.
- Ofrece enlaces directos al contenido para mejorar la navegación en la página.
- Ofrece teclas de acceso directo (*accesskeys*).
- Elige tecnologías que permitan a los agentes de usuario y productos de apoyo navegar por la estructura.
- Ubica el contenido mediante el marcado de estructura.

Titulado de páginas

Las páginas web tienen títulos que describen su temática o propósito.

Técnicas para cumplir el criterio

- Ofrece un título descriptivo para cada página web.

Técnicas para mejorar la usabilidad

- Identifica la relación de la página web con un conjunto de páginas. Por ejemplo, en el *title* de la página incluye el nombre del sitio.
- Identifica el tema de la página web.
- Identifica los elementos *frame* con nombres con sentido.
- Usa un título único para cada página del sitio.
- Ofrece un encabezado descriptivo en la parte superior de la página.

¡No lo hagas!

- No identifiques la página o sus contenidos con textos como “Documento sin título” o similar.

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.

Técnicas para cumplir el criterio

- Sitúa los elementos interactivos en una secuencia que siga el orden y las relaciones dentro del contenido.
- Sitúa el foco en los elementos en un orden secuencial. Elige una de estas opciones:
 - Crea un orden lógico de tabulación entre enlaces, controles de formulario y objetos.
 - Haz que el orden del DOM sea igual que el orden visual.
- Si quieras que una página cambie dinámicamente, elige una de estas opciones:
 - Inserta contenido dinámico en el DOM inmediatamente después del elemento disparador.
 - Crea diálogos personalizados de forma independiente al dispositivo.
 - Reordena las secciones de la página con el DOM.

Técnicas para mejorar la usabilidad

- Destaca 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, ofrece al usuario un mecanismo para usar un orden de presentación alternativo más sencillo y evidente.

¡No lo hagas!

- No utilices el atributo *tabindex* para crear un orden de tabulación que no preserve el significado y la operatividad de la página.
- No utilices diálogos o menús que no estén junto a su control disparador en el orden de la secuencia de navegación.

Propósito de los enlaces (en contexto)

El propósito de cada enlace puede ser determinado con sólo 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.

Técnicas para cumplir el criterio

- Identifica el objetivo de un enlace usando:
 - un texto clarificador en el enlace;
 - un texto en el enlace combinado con el texto de la oración que lo engloba; o
 - un texto de enlace combinado con su contexto de forma que sea determinable por software. El contexto puede ser:
 - el párrafo que contiene el enlace (ejemplo 1).
 - el elemento de la lista que contiene el enlace (ejemplo 2).
 - en una lista anidada, el elemento padre que contiene la lista en la que se encuentra el enlace (ejemplo 3).
 - la celda de tabla que contiene el enlace más sus encabezados asociados.
- Si usas un elemento *anchor*, ofrece un enlace de texto que describa su objetivo.
- Si usas un mapa de imagen, ofrece alternativas textuales para los elementos *area*.
- Permite al usuario elegir enlaces cortos o largos con un control al inicio de la página o por programación.
- Proporciona una información complementaria sobre el objetivo del enlace:
 - Hazlo en el propio texto de enlace y si necesitas ocultar un trozo del texto del enlace, hazlo con CSS¹³
 - Utiliza el atributo *title* del enlace.

En la siguiente tabla mostramos tres ejemplos de un texto de enlace combinado con su contexto de forma que pueda ser determinable por software:

¹³ Es muy importante que lo escondas con una técnica que lo oculte visualmente pero no para el lector de pantalla. Puedes usar *text-indent:-1000px* pero no *display:none* Más información en el artículo “Ocultar contenido sin comprometer la accesibilidad ni el posicionamiento de la página” en <https://olgacarreras.blogspot.com.es/2011/11/ocultar-contenido-sin-comprometer-la.html>

Tabla 4 Texto de enlace combinado con contexto determinable por software

Ejemplo	Forma correcta	Forma incorrecta
1	<pre><p>Informe anual en PDF</p></pre>	<pre><p>Informe anual</p> <p>en PDF</p></pre>
2	<pre> Informe anual (PDF) ... </pre>	<pre> Informe anual (PDF) </pre>
3	<pre> Informe anual 2016-2017 (HTML) (PDF) ... </pre>	<pre><p>Informe anual 2016-2017</p> (HTML) (PDF) </pre>

Técnicas para mejorar la usabilidad

- Identifica el propósito del enlace usando el texto del enlace combinado con el encabezado que le precede.
- Si hay una imagen y un texto que están adyacentes y que enlazan al mismo sitio, combínalos.

¡No lo hagas!

- No ofrezcas enlaces sin un texto que lo contextualice (como “Pulsa aquí” o “Leer más”).
- No dejes vacío el atributo *alt* de una imagen cuando la imagen sea el único contenido de un enlace.

Múltiples vías

Existe más de un camino para llegar a una página web dentro de un conjunto de páginas web, excepto cuando la página es un paso intermedio o el resultado de un proceso.

Técnicas para cumplir el criterio

Ofrece dos o más de los siguientes mecanismos:

- enlaces para navegar a páginas web relacionadas.
- una tabla de contenidos.
- un mapa del sitio.
- un buscador.
- una lista de enlaces a todas las páginas del sitio en la página de inicio.
- una lista de enlaces a todas las páginas del sitio en todas las páginas del sitio.

Técnicas para mejorar la usabilidad

- Identifica la relación de la página web con el conjunto de páginas mediante elementos *link* más el atributo *rel*, lo cual puede ser usado por las herramientas de navegación que disponen algunos navegadores.
- Incluye información sobre los modos de presentación en tablas de contenidos y en mapas conceptuales.

Encabezados y etiquetas

Los encabezados y las etiquetas describen el tema o propósito.

Técnicas para cumplir el criterio

Ofrece encabezados y etiquetas que sean descriptivos.

Técnicas para mejorar la usabilidad

- Ofrece encabezados de sección únicos en la página web.
- Comienza los encabezados de sección con información única.

Foco visible

El indicador del foco del teclado es visible en cualquier elemento de la interfaz de usuario que sea operable por teclado.

Técnicas para cumplir el criterio

- Diseña un indicador de foco de sistema muy visible.
- Usa el indicador predefinido de foco de sistema, el cual es muy visible.
- Usa los componentes de la interfaz que son resaltados por el agente de usuario cuando reciben el foco.
- Si quieras cambiar el diseño de los elementos cuando reciban el foco, hazlo con CSS o con programación.

Técnicas para mejorar la usabilidad

- Destaca claramente los enlaces y controles cuando el ratón pase por encima de ellos o cuando reciban el foco por teclado.

¡No lo hagas!

- No quites el foco de un elemento por programación justo cuando lo reciba.
- No ocultes el indicador visual del foco quitando los estilos del borde o de la línea exterior del elemento.

Ubicación

Se informa de la ubicación del usuario dentro del conjunto de páginas web.

Técnicas para cumplir el criterio

- Ofrece un camino de migas de pan.
- Ofrece un mapa del sitio.
- Indica la localización actual en el menú de navegación.
- Identifica la relación de la página web con un conjunto de páginas con elementos *link* más el atributo *rel*, lo cual puede ser usado por las herramientas de navegación que disponen algunos navegadores.

Técnicas para mejorar la usabilidad

- Ofrece un enlace a la página de inicio o a la página principal.
- Ofrece un resumen fácil de leer al inicio de cada sección de contenido.
- Ofrece información fácil de leer y en lengua de señas sobre la organización de un conjunto de páginas web.

Propósito de los enlaces (sólo enlaces)

Se proporciona un mecanismo que permite identificar el propósito de cada enlace con tan sólo el texto del enlace, excepto cuando el propósito del enlace deba resultar ambiguo para los usuarios en general.

Técnicas para cumplir el criterio

- Identifica el objetivo de un enlace:
 - usando el atributo *aria-label*
 - un texto clarificador en el enlace;
- Si usas un elemento *anchor*, ofrece un enlace de texto que describa su objetivo.
- Si usas un mapa de imagen, ofrece alternativas textuales para los elementos *area*.
- Permite al usuario elegir entre enlaces cortos o largos mediante:
 - un control al inicio de la página web que cambie el texto de los enlaces; o
 - una programación que cambie el texto de los enlaces.
- Proporciona una información complementaria sobre el objetivo del enlace. Hazlo en el propio texto de enlace y si necesitas ocultar un trozo del texto del enlace, hazlo con CSS.

Técnicas para mejorar la usabilidad

- Combina una imagen y un texto en un único enlace en lugar de tener dos enlaces, uno para la imagen y otro para el texto. (Ejemplo 1 y 2)
- Identifica el propósito del enlace usando su *title*. (Ejemplo 3)

Ejemplo	Forma correcta
1	Memoria 2018
2	 Inicio
3	OK

¡No lo hagas!

- No ofrezcas enlaces sin un texto que lo contextualice (como “Pulsa aquí” o “Leer más”).
- No dejes vacío el atributo *alt* de una imagen cuando la imagen sea el único contenido de un enlace.

Encabezados de sección

El contenido está organizado mediante encabezados de sección.

Este criterio alude a estructurar los contenidos, no los componentes de la interfaz de usuario.

Técnicas para cumplir el criterio

- Organiza la página usando encabezados.
- Proporciona encabezados (*h1-h6*) al comienzo de cada sección del contenido.

Técnicas para mejorar la usabilidad

- Ofrece un mecanismo para navegar por las secciones de contenido de la página web.
- Marca las regiones de la página que se actualizan automáticamente con la propiedad *aria-live* de ARIA.

Consejos sobre encabezados

El Real Instituto Nacional de Personas Ciegas del Reino Unido recomienda también:

1. Los encabezados deben dividir trozos grandes de texto.
2. Los encabezados deben ser cortos para permitir el escaneo rápido de la página.
3. Todos los encabezados visuales deben tener un encabezado de estructura.
4. Los encabezados deben describir la estructura de la página y no sólo ser usados para mostrar efectos visuales.
5. La estructura de encabezados debe ser consistente en todo el sitio web.
6. Los encabezados deben estar situados justo antes de su contenido asociado.
7. Los encabezados deben resumir su contenido asociado.
8. Las páginas deben tener un único encabezado *h1*, el cual es el título principal de la página y debe estar ubicado en el inicio del contenido principal de la página. (Sobre este tema hay bastante controversia, sobre todo en discusión con los especialistas en posicionamiento).
9. Los subencabezados de *h1* deben ser *h2*. Los subencabezados de *h2* deben ser *h3*, y así sucesivamente. Por ejemplo, no se debe saltar de un *h1* a un *h3*.
10. Usa los elementos *legend* y *label* en los formularios, no los encabezados.

Pauta 2.5 [NUEVA]

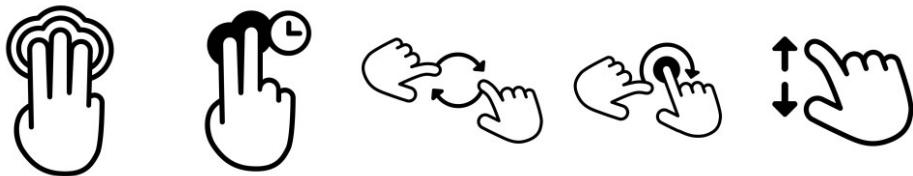
Formas de introducción de la información

Facilita el uso de las funcionalidades a través de varias modalidades de input, más allá del teclado.

Esta nueva pauta agrupa seis nuevos criterios relacionados con las interacciones táctiles, por voz, por puntero u otras que puedan surgir en el futuro.

Gestos del puntero [NUEVO 2.1]

Todas las funcionalidades que usen varios puntos de interacción simultáneos o se basen en un recorrido gestual, deben poder ser operados también por un único punto sin obligar a seguir un recorrido (a menos que el uso de varios puntos o el recorrido sea totalmente esencial).



En estas imágenes¹⁴ se muestran algunos ejemplos de estos recorridos gestuales o interacción con varios puntos al mismo tiempo: dos toques con tres dedos al mismo tiempo, una pulsación larga con dos dedos, dar vueltas con dos dedos, pulsar con un dedo y rotar con otro... o el clásico de agrandar la pantalla con un índice y el pulgar en direcciones opuestas.

Las personas con baja movilidad que no pueden realizar estos movimientos, o personas con dificultades de aprendizaje, necesitan de un mecanismo alternativo más sencillo para poder interactuar con el sistema, como un puntero (independientemente de que también se deba poder manejar por teclado).

Ejemplos de activación de un solo punto son: toques (clic), doble toque (doble clic) y pulsaciones largas.

Técnicas para cumplir el criterio

- No bases la interacción solamente en recorridos gestuales o en varios toques.
- Ofrece controles alternativos sencillos con la misma función.

¡No lo hagas!

- No obligues a que tu funcionalidad no pueda ser operada con la activación de un único punto.

¹⁴ Iconos de Jeff Portaro en The Noun Project (<https://thenounproject.com/jeffportaro>)

Cancelación del puntero [NUEVO 2.1]

Para las funcionalidades que se pueden manejar con un puntero sencillo, al menos se cumple una de las siguientes premisas:

- El evento “abajo” del puntero no se usa para realizar ninguna parte de la función, a menos que sea esencial.
- La finalización de la función está en el evento “arriba”, y existe algún mecanismo para abortar la función antes de realizarse, o deshacer la función una vez hecha.
- El evento “arriba” invierte cualquier resultado del evento “abajo” anterior.

En primer lugar, se hace necesario explicar los eventos “arriba” y “abajo”. Cuando accionamos una tecla, un botón del ratón o un control en la pantalla, hay al menos dos movimientos, uno de pulsar hacia “abajo” (*onmousedown*, *onkeydown*, *onkeypress*, *ontouchstart*) y otro de soltar hacia “arriba” (*onmouseup*, *onkeyup*, *ontouchend*).

En las pantallas táctiles donde se pueden hacer deslizamientos (*scroll* o *swipe*), por ejemplo, en un *scroll* infinito, un carrusel o en la aplicación móvil Swype¹⁵, el evento “arriba” correspondería con la última posición del puntero.

Las personas con dificultad para manejar el puntero, con baja visión o limitaciones cognitivas necesitan esa seguridad de que, cuando manejan el contenido, un toque “abajo” por error puede ser corregido.

Una excepción sería, por ejemplo, un teclado mostrado en pantalla. Estamos habituados a que cuando se pulse la tecla se muestre en pantalla ese carácter elegido, es decir, que la funcionalidad se dispare en el evento “abajo”, y por tanto sería una característica esencial de la funcionalidad.

Pero la norma sí aplicaría a un botón de “Enviar”. Dependiendo de dónde o cómo el usuario realiza el evento “arriba”, se dispararía la funcionalidad o no, o ésta se podría cancelar.

Técnicas para cumplir el criterio

- Activar los controles usando los eventos “arriba”.
- Los eventos de toque de pantalla sólo se activan cuando se retira el toque del control.

¡No lo hagas!

- No actives un control en un evento “abajo” si el evento “arriba” no coincide con la misma localización del puntero.

¹⁵ <http://www.swype.com>

Etiqueta en el nombre [NUEVO 2.1]

Para los componentes de interfaz con etiquetas que incluyen texto o imágenes de texto, el nombre accesible contiene el texto que se presenta visualmente.

Una buena práctica es poner el texto de la etiqueta al comienzo del nombre accesible.

Muchos elementos de HTML tienen nombres accesibles y se calculan de varias maneras: de su contenido, de un atributo o de un elemento asociado¹⁶. En los siguientes ejemplos, el nombre accesible será en todos los casos “Volver a inicio”:

```
<a href="inicio.htm">Volver a inicio</a>

<a href="inicio.htm"></a>
<a href="inicio.htm">Volver a </a>

<input type="button" value="Volver a inicio" />
<input type="checkbox" id="xx">Marca esta casilla para <label
for="xx">Volver a inicio</label> al finalizar la compra.
```

Con los atributos de etiqueta de ARIA (*aria-label* y *aria-labelledby*) podemos modificar estos nombres accesibles. En los siguientes ejemplos el nombre accesible sería, una vez más “Volver a inicio”, en lugar de “X”, que sería su nombre accesible nativo y visible:

```
<button aria-label="Volver a
inicio">X</button>

<button aria-
labelledby="btnInicio">X</button>
<a href="" id="btnInicio">Volver a
inicio</a>
```

Los nombres accesibles permiten a los usuarios de productos de apoyo diferenciar los componentes, navegar por ellos y finalmente manejarlos.

Veamos por ejemplo el caso de:

```
<button aria-label="Volver a
inicio">X</button>
```



¹⁶ Puedes leer más sobre cómo los navegadores calculan este nombre accesible por cada elemento en <https://www.w3.org/TR/acname-1.1/> y en cómo mapear los nombres de cada elemento con *aria-label*, *aria-labelledby*, y *aria-describedby* en <https://www.w3.org/TR/core-aam-1.1/>

Los usuarios que utilizan un lector de pantalla y que ven el botón, por ejemplo, usuarios con baja visión, tendrán una mejor experiencia:

- si lo que anuncia el lector es igual que la etiqueta visible; o
- si lo que anuncia el lector contiene el texto de la etiqueta visible, preferiblemente al comienzo del nombre accesible.

En nuestro ejemplo, sería más adecuada esta forma:

```
<button aria-label="X: Volver a inicio">X</button>
```

Para los usuarios que utilizan un programa de reconocimiento de voz, que ven la etiqueta y usan la voz como medio de entrada, el problema es más grave.

Veamos por ejemplo el caso de:

```
<button aria-label="Aceptar">OK</button>
```

Estos usuarios intentarán activar el botón, el menú o el enlace con su etiqueta visible (“OK” en nuestro ejemplo) pero no funcionará si tiene un nombre accesible diferente (“Aceptar” en este caso). Además, puede ocurrir que lo activen por error si dicen su nombre accesible. En este segundo ejemplo, lo correcto es que la etiqueta visual y el nombre accesible coincidan.

Técnicas para cumplir el criterio

Asegúrate de que las etiquetas visibles, incluido el estado *hover* de los iconos, concuerdan con las etiquetas ocultas.

¡No lo hagas!

No pongas nombres accesibles que:

- No contengan el texto visible.
- Que contenga el texto visible pero las palabras de la etiqueta visible estén en un orden diferente.
- Que contengan el texto visible pero una o más palabras se entremezclan con la etiqueta.
- Que contengan el texto visible pero que haya una o más palabras adicionales antes del texto de la etiqueta visible.

Actuación por movimiento [NUEVO 2.1]

Las funcionalidades que se pueden manejar con el movimiento del dispositivo o del usuario pueden ser operadas por componentes de la interfaz de usuario, y la respuesta al movimiento puede ser deshabilitada para prevenir una activación accidental.

Excepciones:

- La interfaz lo permite: se usa el movimiento para operar la funcionalidad mediante una interfaz compatible con la accesibilidad.
- El movimiento es esencial para la funcionalidad y si no se hiciera se invalidaría la actividad.

El objetivo de este criterio es asegurar que las funciones que se activan con un movimiento del dispositivo (por ejemplo, inclinar o agitar el móvil), o con gestos (por ejemplo, mover la mano sobre la *webcam*), pueden ser activadas por componentes de la interfaz de usuario, a menos que el movimiento sea esencial para esa función.

Este criterio sólo se aplica a los sensores de movimiento inherentes a los dispositivos (acelerómetros y giroscopios), no a la geolocalización GPS, ni a los movimientos asociados de forma indirecta al uso del teclado, el puntero y otras tecnologías.

Además, las personas con temblores o discapacidades motoras deben poder desactivar este tipo de actuación por movimiento para evitar realizarlo de forma accidental.

Técnicas para cumplir el criterio

- No uses el evento *devicemotion* para activar funcionalidades y contenidos.
- Proporciona formas alternativas de introducir información cuando se usan los sensores de movimiento para activar funcionalidades y contenidos.
- Proporciona ajustes en la aplicación que permitan desactivar la actuación por movimiento.
- Da apoyo a las características del sistema que permitan al usuario desactivar la actuación por movimiento.

¡No lo hagas!

- No actives funcionalidades y contenidos solamente mediante el evento *devicemotion*.
- No desabilites, ofusques o entorpezcas de algún modo la posibilidad del usuario de desactivar la actuación por movimiento

Tamaño del área de interacción [NUEVO 2.1]

El tamaño mínimo de las áreas interactivas debe ser 44 por 44 píxeles CSS.

Excepciones:

- Existe un enlace o control equivalente en la misma página que cumple esas medidas mínimas.
- El área está dentro de una frase o un bloque de texto.
- El tamaño del área lo determina automáticamente el agente de usuario y no es modificada por el autor.
- Es esencial que el área tenga un tamaño menor para la información que muestra.

El objetivo de este criterio es ayudar a las personas que sufren de temblores, tienen baja visión, o poca destreza apuntando a objetivos pequeños con el dedo o con un puntero, o que simplemente viajan en el autobús y están expuestas a movimientos del propio transporte.

Aunque 44 por 44 píxeles es la medida mínima, **se recomienda que sea mayor** cuando:

- el control se deba usar frecuentemente;
- el resultado de su interacción no se pueda deshacer fácilmente;
- esté cerca del borde de la pantalla;
- sea difícil de alcanzar;
- sea parte de una tarea secuencial.

Técnicas para cumplir el criterio

- Asegúrate de que las áreas interactivas tienen al menos 44 por 44 píxeles.
- Ofrece un mecanismo que permita cambiar el tamaño de las áreas independientemente de la ampliación.

Técnicas para mejorar la usabilidad

- Asegúrate de que los enlaces en línea tienen un área de activación lo suficientemente grande.

¡No lo hagas!

- No diseñas áreas interactivas ni párrafos que a la vez son enlaces menores de 44 por 44 píxeles.

Mecanismos de entrada concurrentes [NUEVO 2.1]

El contenido web no restringe el uso de las diferentes formas de introducir información excepto cuando la restricción es esencial, obligatoria para garantizar la seguridad del contenido, o requerida para respetar las preferencias del usuario.

No debemos asumir que el usuario siempre usará un único mecanismo de entrada concreto, por lo que debemos permitir usar cualquier mecanismo de entrada a su disposición: teclado, ratón, voz, pantalla táctil, etc.

También debemos permitir al usuario cambiar de una modalidad a otra, a su voluntad.

Técnicas para cumplir el criterio

- En JavaScript, usa solo manejadores de eventos de alto nivel, independientes de la modalidad de entrada (*focus*, *blur*, *click*)
- En JavaScript, usa manejadores de eventos redundantes para teclado, puntero y pantalla táctil (por ejemplo, activa la misma funcionalidad en *pointerdown*, *touchstart* o *mousedown*).

¡No lo hagas!

- No uses eventos de teclado, de puntero o de pantalla táctil diferentes para la misma funcionalidad.
- No determines cuál será el principal mecanismo de entrada, asumiendo que no se añadirán o usarán otros.

Principio 3. Comprensible

Si nuestros usuarios no comprenden lo que les estamos diciendo, o les hacemos sentirse perdidos, tenemos un problema.

Debemos diseñar nuestro sitio web de forma sencilla, y esto incluye tanto la información como la interfaz de usuario.



Pauta 3.1

Fácil de leer y de comprender

Haz que los contenidos textuales resulten fáciles de leer y comprensibles.

El contenido es el rey en internet. Esta pauta está destinada en especial a los editores de páginas, para que准备n los textos de tal forma que cualquiera pueda comprenderlos.

Consejos, a modo general

- Evita los controles y elementos de interfaz innecesarios.
- Alinea el texto:
 - Evita el alineado central de texto.
 - No justifiques el texto (alineado tanto a los márgenes de izquierda y derecha) si causa un espaciado pobre entre palabras o caracteres.
 - Alinea el texto a la izquierda en idiomas que se escriban de izquierda a derecha; y alinea el texto a la derecha en idiomas que se escriban de derecha a izquierda.
- Aclara el contenido:
 - Fija expectativas sobre el contenido proveniente de fuentes no controladas.
 - Escoge el lenguaje más claro, más simple y más adecuado al contenido.
 - Complementa con imágenes, dibujos, vídeos, sonidos, símbolos o ejemplos prácticos.
 - Haz referencia a los encabezados incluyendo el texto completo del encabezado.
 - Evita palabras extranjeras inusuales.
 - Escribe con mayúsculas y minúsculas según las convenciones del idioma.
 - Ofrece información fácil de leer y en lengua de señas sobre temas básicos, incluyendo la información de contacto.
 - Ofrece una interpretación en lengua de señas para todos los contenidos (información, ideas y procesos) que deban ser comprendidos en orden para usar el contenido.
 - Si haces referencia a una parte determinada de la página web, enlaza directamente a esa parte mediante elementos *anchor*.

- Diseña el contenido:

- Limita el ancho de las columnas de texto.
- Evita los grandes trozos de texto en *italica*.
- Evita abusar de diferentes estilos en las páginas.
- Distingue visualmente los enlaces.
- Usa un fondo de color pastel claro mejor que fondo blanco detrás de un texto negro para crear un contraste suficiente pero no extremo.

Idioma de la página

El idioma predeterminado de cada página web puede ser determinado por software.

Técnicas para cumplir el criterio

Identifica el idioma o los idiomas predeterminados usando los atributos de idioma.

Por ejemplo, en HTML 5 se especificaría el español así:

```
<html lang="es">
```

En XHTML 1 debemos usar tanto el atributo *lang* como el atributo *xml:lang* juntos con el mismo valor cada vez que establezcamos el idioma.

```
<html lang="es" xml:lang="es" xmlns="http://www.w3.org/1999/xhtml">
```

Técnicas para mejorar la usabilidad

Especifica el idioma predeterminado de toda la página usando el encabezado *HTTP* de servidor.

Por ejemplo, en PHP se haría así:

```
<?php header ('Content-language: es') ; ?>
```

Idioma de las partes de la página

El idioma de cada pasaje o frase en el contenido puede ser determinado por software.

Excepciones: nombres propios, términos técnicos, palabras en un idioma indeterminado y palabras o frases que se han asimilado al idioma del texto que las rodea.

Técnicas para cumplir el criterio

- Identifica los cambios de idioma usando el atributo de idioma (*lang*, *xml:lang...*) en los elementos de HTML.

En las imágenes que incluyen texto en otro idioma, el atributo *alt* debe contar con esas palabras, y se debe usar el atributo de idioma en la imagen para que el lector de pantalla lo lea correctamente

Técnicas para mejorar la usabilidad

- Distingue visualmente los textos que no estén en el idioma predefinido.
- Da los nombres de todos los idiomas, distintos al predefinido, usados en pasajes y oraciones.
- Marca el idioma de los nombres propios de personas con el atributo de idioma (*lang*, *xml:lang...*) para facilitar a los lectores de pantalla su correcta pronunciación.



Puedes leer más sobre internacionalización en la web de la W3C.

<https://www.w3.org/International/questions/qa-html-language-declarations>

Palabras inusuales

Se proporciona un mecanismo para identificar las definiciones específicas de palabras o frases usadas de modo inusual o restringido, incluyendo las expresiones idiomáticas y la jerga.

Técnicas para cumplir el criterio

Si la frase o palabra inusual tiene un significado único dentro de la página web:

- Defínela la primera vez que aparezca:
 - enlaza a las definiciones: usando listas de definición o enlazando a un glosario externo; o
 - añade la definición en línea, en HTML con el elemento *dfn*
- Defínela cada vez que aparezca:
 - enlaza a las definiciones: usando listas de definición o enlazando a un glosario externo; o
 - proporciona un glosario propio; o
 - proporciona una funcionalidad de búsqueda en un diccionario *online*.

Si la frase o palabra inusual se usa con significados diferentes dentro de la página web:

- Defínela cada vez que aparezca en la página enlazando a las definiciones o añadiendo la definición en línea tal y como se ha especificado anteriormente.

Técnicas para mejorar la usabilidad

- Distingue visualmente las palabras inusuales para ayudar a los usuarios a reconocerlas.
- Ofrece una búsqueda en diccionario a través de voz para los usuarios con dificultades para teclear.
- Ofrece un diccionario en lengua de señas para ayudar a los usuarios sordos.
- Ofrece un mecanismo para encontrar definiciones para todas las palabras o para determinar su significado.
- Evita las palabras inusuales extranjeras.

Abreviaturas

Se proporciona un mecanismo para identificar la forma expandida o el significado de las abreviaturas.

Técnicas para cumplir el criterio

Si la abreviatura tiene un significado único dentro la página web:

- Explícala la primera vez que aparezca:
 - expandiendo el significado inmediatamente antes o después, por ejemplo, WAI (*Web Accessibility Initiative*); o
 - enlazando a las definiciones; o
 - usando el elemento *abbr*
- Explícala cada vez que aparezca:
 - enlazando a las definiciones; o
 - usando el elemento *abbr*; o
 - ofreciendo un glosario; o
 - proporcionando una funcionalidad de búsqueda en un diccionario *online*.

Si la abreviatura se usa con significados diferentes dentro de la misma página web:

- Expándela o explícala cada vez que aparezca:
 - enlazando a las definiciones, o
 - usando el elemento *abbr*

Técnicas para mejorar la usabilidad

- Utiliza abreviaturas únicas en cada página web.
- Ayuda a los usuarios a reconocer las abreviaturas mediante el formato visual.
- Ofrece acceso a un diccionario parlante para ayudar a los usuarios que puedan tener dificultades descodificando las definiciones escritas.
- Ofrece una búsqueda en diccionario a través de voz para los usuarios con dificultades para teclear.

Nivel de lectura

Cuando un texto requiere un nivel de lectura más avanzado que el nivel mínimo de educación secundaria, una vez que se han eliminado nombres propios y títulos, se proporciona un contenido suplementario o una versión que no requiere un nivel de lectura mayor a ese nivel educativo.

El nivel de educación que se establece en este criterio de conformidad es de entre 7 y 9 años de escolarización (*Lower secondary education* según el Estándar Internacional de Clasificación de la Educación de la UNESCO¹⁷). Según la clasificación de la educación en España, se corresponde con un nivel de educación entre 1º y 3º de la Educación Secundaria Obligatoria (ESO)¹⁸.

En el capítulo *Recursos y Herramientas* encontrarás una herramienta que te ayudará a evaluar la dificultad de lectura de un texto.

Técnicas para cumplir el criterio

- Ofrece un resumen que pueda ser comprendido por personas con un nivel lector de educación secundaria (entre 7 y 9 años de escolarización).
- Ofrece dibujos, fotografías y símbolos para ayuden a explicar ideas, eventos y procesos.
- Ofrece una versión hablada del texto.
- Haz el texto más fácil de leer.
- Ofrece una versión en lengua de señas de la información, ideas y procesos que deben ser comprendidos para poder usar la página.

Técnicas para mejorar la usabilidad

- Ofrece texto en las páginas de navegación, interiores y de destino (*landing pages*) que requieran una habilidad lectora menor que el nivel de educación secundaria (entre 7 y 9 años de escolarización).
- Incluye un resumen del contenido a través de los metadatos.
- Escribe con el lenguaje más claro y simple adecuado al contenido.
- Asocia el contenido principal con el contenido secundario mediante RDF.
- Ofrece una “imagen” representativa clara en la página de inicio del sitio.
- Indica claramente, con texto o un ícono, el contenido que ha sido optimizado para una fácil lectura.

¹⁷ http://www.unesco.org/education/information/nfsunesco/doc/isced_1997.htm

¹⁸ http://www.ine.es/daco/daco42/clasificaciones/cned14/CNED2014_capitulo0.pdf

- Escribe oraciones sin palabras superfluas, es decir, que no cambian el significado de la oración.
- Escribe oraciones que contengan dos conjunciones como máximo.
- Escribe oraciones que no sean más largas que la longitud aceptada para educación secundaria (en inglés 25 palabras)
- Escribe oraciones que no contengan palabras complejas que podrían ser reemplazadas por palabras más comunes sin que cambie el significado de la oración.
- Ofrece resúmenes para las diferentes secciones de texto.
- Asocia las versiones alternativas a diferentes niveles de lectura con metadatos.
- Asocia el contenido de texto con los contenidos textuales, gráficos o sonidos complementarios mediante los elementos de accesibilidad de los estándares *Dublin Core*, ISO AfA o IMS.
- Haz visibles los metadatos a los usuarios.



Puedes encontrar recomendaciones para elaborar documentos fáciles de leer en:

Lectura fácil: Métodos de redacción y evaluación de Oscar García Muñoz
www.plenainclusion.org/sites/default/files/lectura-facil-metodos.pdf

Norma UNE 153101 de “Lectura fácil”
<https://www.aenor.com/normas-y-libros/buscador-de-normas/UNE?c=N0060036>

Pronunciación

Cuando el significado de una palabra resulte ambiguo si no se conoce su pronunciación dentro del contexto, se proporciona un mecanismo para indicar su pronunciación.

Técnicas para cumplir el criterio

- Ofrece la pronunciación inmediatamente después de la palabra.
- Enlaza a las pronunciaciaciones.
- Ofrece un glosario que incluya la pronunciación de la palabra.
- Ofrece la pronunciación usando el elemento *ruby* de HTML, el cual añade pequeños textos de ayuda junto a un texto base¹⁹. El elemento *ruby* se utiliza para indicar la pronunciación o para hacer pequeños comentarios, como en el siguiente ejemplo.

```
<ruby>  
<rb>WCAG</rb>  
<rp>(</rp><rt>Wuh-KAG</rt></rp>)</rp>  
</ruby>
```



- Ofrece la pronunciación usando marcas diacríticas estándares, pero que puedan ser apagadas. En algunos idiomas, la translación escrita del lenguaje oral es un reto y se puede presentar de diversas maneras, como el ‘*pinyin*’ para el chino o los marcados ‘*okina*’ y ‘*kabako*’ para el hawaiano. Las marcas diacríticas pueden ser completamente necesarias para la comprensión de los textos, pero en ocasiones la fuente elegida no soporta dichos caracteres. Por ello, se recomienda utilizarlas, pero siempre dando la posibilidad al usuario de no mostrarlas si su ordenador no da soporte a la fuente utilizada.

Técnicas para mejorar la usabilidad

- Ofrece la pronunciación en un archivo de sonido para que los usuarios puedan escuchar la palabra.
- Ofrece un mecanismo para determinar las pronunciaciaciones de todas las palabras o frases extranjeras en el contenido textual.

¹⁹ No lo confundas con el lenguaje de programación del mismo nombre. Puedes leer la especificación completa de este elemento en <http://www.w3.org/TR/ruby>.

Pauta 3.2

Haz tu página **predecible**, no reinventes los estándares de navegación

Haz que las páginas web aparezcan y se manejen de manera predecible.

Si una página está continuamente refrescándose, cambiando el contenido, abriendo nuevas ventanas... sin que el usuario lo controle, puede confundirle, tanto si tiene como si no tiene alguna discapacidad.

Sé coherente y consistente en el diseño y permite acceder a la información de forma fácil.

Consejos, a modo general

- Posiciona correctamente las etiquetas para maximizar la predictibilidad de las relaciones entre los elementos y los eventos.

Al recibir el foco

Cuando cualquier componente recibe el foco, no se inicia ningún cambio en el contexto.

Un **cambio de contexto** es un cambio importante en el contenido de una página web que, cuando se hace sin el consentimiento del usuario, puede desorientar a quienes no pueden ver toda la página al mismo tiempo. Son cambios de contexto:

- **Cambiar de aplicación de usuario:** por ejemplo, abrir el gestor de correo o un lector de documento PDF;
- **Cambiar de vista:** el área donde se visualiza el contenido;
- **Cambiar de foco:** mover el foco de un elemento a otro;
- **Cambiar el contenido** de tal modo que cambie el significado de la página.

Un **cambio de contenido** no siempre es un cambio de contexto. Los cambios en el contenido como consecuencia de mostrar contenido nuevo u oculto, -como un acordeón de preguntas y respuestas frecuentes desplegable, un menú dinámico o un control de pestañas-, no cambian necesariamente el contexto, a menos que produzcan también algún otro cambio que sí sea considerado un cambio de contexto, como:

- abrir una nueva ventana;
- mover automáticamente el foco a otro componente (muy habitual al llenar campos de tipo cuenta bancaria o de tipo fecha);
- ir a otra página (incluyendo hacer creer al usuario que ha ido a otra página); o
- reorganizar el contenido de una página de forma significativa.

Técnicas para cumplir el criterio

- Usa una acción de tipo *activate* (presionar un botón, hacer clic en un enlace) para realizar un cambio de contexto en lugar de hacerlo cuando el elemento recibe el foco. Observa que se habla de cambios de contexto, no de cambios de contenido.

Técnicas para mejorar la usabilidad

- Evita cambiar estados o valores de forma permanente cuando un componente recibe el foco, o bien ofrece un mecanismo para restablecer cualquier cambio.
- Abre ventanas nuevas y pestañas sólo cuando sea necesario y avisa a los usuarios.

¡No lo hagas!

- No abras una ventana nueva tan pronto como se cargue la página.
- No quites el foco de un elemento justo cuando lo recibira.

Al recibir entradas

El cambio de estado en cualquier componente de la interfaz 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.

Técnicas para cumplir el criterio

- Para iniciar un cambio de contexto, ofrece un botón que el usuario pueda pulsar.
- Si un cambio en un control de formulario causa un cambio de contexto, describe antes qué sucederá.
- Usa el evento *onchange* en el elemento *select* sin provocar un cambio de contexto

Técnicas para mejorar la usabilidad

- Avisa que se va a abrir una nueva ventana.

¡No lo hagas!

- No envíes sin previo aviso un formulario automáticamente, presentando contenidos nuevos, cuando se rellena el último campo del formulario.
- No abras una ventana nueva -sin antes avisar- cuando cambia el estado de un botón de radio, una casilla de verificación o una lista de selección.

Navegación coherente

Los mecanismos de navegación que se repiten en múltiples páginas web dentro de un conjunto de páginas web aparecen siempre en el mismo orden relativo cada vez que se repiten, a menos que el cambio sea provocado por el propio usuario.

El objetivo de este criterio es que se pueda predecir la ubicación del contenido que se está buscando, muy útil para las personas con baja visión.

El “mismo orden” no significa que no se puedan añadir o eliminar elementos en la secuencia de una página a otra, por ejemplo, un bloque de navegación secundaria, sino que, de hacerse, debemos mantener el orden relativo del resto de los elementos.

Técnicas para cumplir el criterio

Presenta los componentes repetidos en varias páginas web en el mismo orden relativo cada vez que aparezcan.

Técnicas para mejorar la usabilidad

Asegura la consistencia de las páginas web del sitio mediante plantillas, diseños, ubicaciones y alineaciones de contenido uniformes. Esto no implica utilizar una única plantilla, sino que las diferentes plantillas deben ser coherentes entre sí.

¡No lo hagas!

No cambies el orden de los enlaces de navegación en páginas distintas.

Identificación consistente

Los componentes que tienen la misma funcionalidad dentro de un conjunto de páginas web son identificados de manera consistente.

El objetivo de este criterio es asegurar la consistencia en la identificación de componentes funcionales que aparecen repetidamente a lo largo del sitio web. Por ejemplo, si se usa un ícono de flecha con el texto alternativo “Descargar” más el nombre del fichero que descarga, utiliza siempre la misma fórmula. O si en una página un ícono significa una cosa, que en otra página no signifique algo diferente.

Técnicas para cumplir el criterio

- Usa etiquetas, nombres y alternativas textuales de forma consistente (lo cual no significa que sean idénticos) para contenidos que tengan la misma funcionalidad y sigue las mismas técnicas para cumplir el criterio que las enumeradas en los Criterios de Conformidad 1.1.1 (Contenido no textual) y 4.1.2 (Nombre, función y valor).

Técnicas para mejorar la usabilidad

- Comprueba que la alternativa textual describe la funcionalidad del componente y qué sucederá cuando el usuario lo active.
- Usa el mismo contenido no textual para una determinada funcionalidad cuando sea posible.

¡No lo hagas!

- No utilices dos etiquetas diferentes para la misma funcionalidad en diferentes páginas del sitio.

Cambios a petición

Los cambios en el contexto son iniciados únicamente a solicitud del usuario o se proporciona un mecanismo para detener tales cambios.

Técnicas para cumplir el criterio

- Si la página web **permite las actualizaciones automáticas**, ofrece un mecanismo para pedir al usuario que lo haga en lugar de hacerlo automáticamente.
- Si la página web **puede redirigir automáticamente**, hazlo en el lado de servidor en lugar del lado de cliente; o hazlo en el lado de cliente con una etiqueta *meta-refresh* con tiempo nulo o “0”.
- Usa el evento *onchange* en el elemento *select* **sin provocar un cambio de contexto**.
- Si se debe **abrir una ventana nueva**,
 - impleméntalo a través del atributo *target* y avisa al usuario en el texto del enlace; o utiliza un script respetando el principio de mejora progresiva y avisando también al usuario de si se abre en ventana nueva (por ejemplo, si la DTD no acepta el atributo *target* como en HTML 4.01 *Strict* o XHTML 1.0 *Strict*).

Un ejemplo de cómo utilizar un script para abrir correctamente nuevas ventanas es el siguiente:

1. Incluye este enlace al archivo *popup.js* en la cabecera de tu página web

```
<script type="text/javascript" src="popup.js"></script>
```

2. Introduce el enlace con un id en tu página web

```
<a href="enlace.html" id="nuevaVentana">Texto del enlace</a>
```

3. Ahora, en el archivo *popup.js* incluye:

```
//capturamos el evento
window.onload = addHandlers;

function addHandlers() {
// capturamos el enlace
    var objAnchor = document.getElementById('nuevaVentana');
    if (objAnchor){

// le añadimos el aviso de apertura en nueva ventana
        objAnchor.firstChild.data = objAnchor.firstChild.data + ' (abre
en nueva ventana)';

// lo lanzamos tanto en evento de puntero y de teclado
        objAnchor.onclick = function(event){return VentanaNueva(this,
event)};
```

```

        objAnchor.onkeypress = function(event){return VentanaNueva
(this, event);}
    }

// función de apertura
function VentanaNueva(objAnchor, objEvent)
{
    var iKeyCode, bSuccess=false;

    // solo queremos que funcione si las teclas pulsadas son espacio
    (32) o return (13)
    if (objEvent && objEvent.type == 'keypress'){
        if (objEvent.keyCode)
            iKeyCode = objEvent.keyCode;
        else if (objEvent.which)
            iKeyCode = objEvent.which;
        if (iKeyCode != 13 && iKeyCode != 32)
            return true;
    }

    // abrimos ventana y le decimos la url
    bSuccess = window.open(objAnchor.href);

    // Si no se abre en nueva ventana, le decimos al navegador que abra
    el enlace en la misma ventana
    if (!bSuccess)
        return true;

    // Parar de aplicar el código
    return false;
}

```

Técnicas para mejorar la usabilidad

Dado que muchos agentes de usuario ya permiten a los usuarios abrir enlaces en otra ventana o pestaña, no abras ventanas o pestañas nuevas o hazlo sólo cuando sea necesario.

¡No lo hagas!

- No cambies el contenido principal mediante actualizaciones automáticas que el usuario no pueda deshabilitar con un mecanismo dentro del contenido.
- No provoques un cambio de contexto cuando el usuario retire el foco de un elemento de formulario.
- No abras ventanas no solicitadas por el usuario, ni que se abran tan pronto como el contenido se cargue, o cuando el usuario introduce texto en un campo de entrada.
- No refresques la página usando la etiqueta *meta-refresh* con un límite de tiempo.

Pauta 3.3

Ayuda en la introducción de datos. *Errare humanum est*

Ayuda a los usuarios a evitar y corregir los errores.

Todos cometemos errores, y no iba a ser diferente en internet. Como diseñadores, debemos prevenir que los usuarios cometan errores, y, si los cometen, ayudarles a superarlos y continuar con su actividad.

La mayoría de los errores se producen en los formularios, así que pon especial atención cuando los diseñas.

Identificación de errores

Si se detecta automáticamente un error en la entrada de datos:

- se identifica el elemento erróneo y
- se describe el error al usuario mediante un texto.

Técnicas para cumplir el criterio

Si el usuario no ha llenado los campos obligatorios de un formulario:

- Usa el atributo *aria-invalid* para indicar los campos que han dado error.
- Proporciona una descripción textual que identifique los campos obligatorios que no se han completado.
- Valida el formulario en el lado de cliente y muestra un *alert()*.

Si la información proporcionada por el usuario debe seguir un formato concreto o debe estar en un rango de valores determinado:

- Utiliza ARIA (*live regions, role="alertdialog", role="alert"*) para identificar la zona de notificación de los errores.
- Usa el atributo *aria-invalid* para indicar los campos que han dado error.
- Avisa al usuario con un mensaje de texto explicando el error.
- Valida el formulario en el lado del cliente y añade un mensaje de error con el DOM o con un *alert()*.

Técnicas para mejorar la usabilidad

- Ofrece un mecanismo para que los usuarios puedan saltar del mensaje de error al campo y del mensaje de error al listado de errores.
- Valida el envío del formulario en el lado de cliente y también en el lado del servidor.
- Si hay varios errores, vuelve a mostrar el formulario con un resumen de los errores
- Advierte de los errores al momento de que el usuario introduzca la información
- Incluye la información de notificación de errores en el título de la página.
- Enfatiza visualmente los errores donde sucedan.
- Complementa el texto de los errores con contenido no textual.
- Avisa al usuario tanto si el formulario ha sido enviado con éxito como si no.
- Utiliza sonidos para enfocar la atención del usuario

Etiquetas o instrucciones

Se proporcionan etiquetas o instrucciones cuando el contenido requiere la introducción de datos por parte del usuario.

Técnicas para cumplir el criterio

- Ofrece etiquetas descriptivas y una de las siguientes opciones:
 - usa *aria-describedby* para describir los controles de la interfaz.
 - usa *aria-labelledby* para definir la etiqueta concatenando textos de varios nodos.
 - usa roles de ARIA para identificar controles de formulario relacionados.
 - indica el formato esperado de los datos y pon un ejemplo.
 - da instrucciones al comienzo del formulario o de un grupo de campos.
 - ubica las etiquetas de tal manera que ayude a predecir las relaciones.
 - añade una descripción para identificar los campos obligatorios no llenados.
 - indica los campos obligatorios usando la etiqueta *label* o *legend*.
- Usa el elemento *label* para asociar las etiquetas con los controles.
- Usa un botón adyacente para etiquetar el objetivo de ese campo.
- Describe los grupos de campos usando los elementos *fieldset* y *legend*.

Técnicas para mejorar la usabilidad

- Si hay un cambio de contexto cuando el usuario cambia un control de formulario, avisa antes al usuario.
- Ofrece un diseño lineal del formulario y agrupa elementos similares.

¡No lo hagas!

- No formatee visualmente un grupo de campos destinado a introducir un número de teléfono (por ejemplo, prefijo, número y extensión) sin utilizar una etiqueta de texto para cada campo.

Sugerencias ante errores

Si se detecta automáticamente un error en la entrada de datos y se dispone de sugerencias para hacer la corrección, entonces se presentan las sugerencias al usuario, a menos que esto ponga en riesgo la seguridad o el propósito del contenido.

Técnicas para cumplir el criterio

Si el usuario no ha llenado los campos obligatorios de un formulario:

- Proporciona descripciones textuales para identificar qué campos obligatorios no se han llenado.
- Identifica los campos obligatorios con el atributo *aria-required*. En el capítulo dedicado a ARIA tienes un ejemplo de cómo hacerlo.

Si la información proporcionada por el usuario debe seguir un formato concreto o debe estar en un rango de valores determinado:

- Usa *role=alertdialog* de ARIA para identificar la zona de notificación de errores.
- Valida el formulario en el lado de cliente y muestra un *alert()*.
- Valida el formulario en el lado de cliente y añade un mensaje de error en el DOM.
- Avisa al usuario con un mensaje de texto explicando el error.
- Sugiere una forma de llenar correctamente el campo.

Técnicas para mejorar la usabilidad

- Indica el número de errores, identifica los campos, aclara con sugerencias y ejemplos cómo corregirlos e indica cómo debe continuar. Los ejemplos deben mostrar formas correctas de introducir los contenidos.
- Sitúa la sugerencia y el ejemplo en el contexto del campo erróneo. Si no es posible, vuelve a mostrar el formulario original y sitúalos junto al primer elemento (o uno de los primeros elementos) de contenido.
- Escribe mensajes de error que sean fáciles de distinguir, de leer y de comprender.
- Si hay varios errores o sugerencias, ofrece un mecanismo para que los usuarios puedan saltar de uno a otro.
- Acepta datos en diversos formatos. Por ejemplo, si piden un número de teléfono, acepta paréntesis, guiones y espacios en la introducción del número, y luego modifícalo por programación para adecuarlo a los requisitos de la base de datos.
- Valida el envío del formulario en el lado de cliente y también en el de servidor.
- Tanto si el formulario ha sido enviado con éxito como si no, avisa al usuario.

Prevención de errores en páginas legales, financieras y de datos

Para las páginas web que

- representan para el usuario compromisos legales o transacciones financieras;
- modifican o eliminan datos controlables por el usuario en sistemas de almacenamiento de datos; o
- envían las respuestas del usuario a una prueba,

se cumple al menos uno de los siguientes casos:

- El usuario puede deshacer o cancelar el envío.
- El sistema detecta errores en la entrada de datos y se permite al usuario corregirlos.
- Se proporciona un mecanismo para que el usuario revise, confirme y corrija la información antes de enviar los datos.

Técnicas para cumplir el criterio

Si una acción del usuario provoca una transacción legal o comercial, elige una opción:

- **Antes del envío**, el usuario puede revisar y corregir sus respuestas; o debe seleccionar un *checkbox* indicando que ha revisado los datos y quiere efectuar la transacción, lo cual se hará con un botón de envío.
- **Después del envío**, se da un tiempo para que el usuario pueda modificar o cancelar la acción.

Si una acción del usuario va a provocar un borrado de información, elige una opción:

- **Antes del envío**, el usuario debe confirmar la acción para continuar; o debe seleccionar un *checkbox* para indicar que han revisado los datos y están seguros de querer borrar la información, lo cual se hará con un botón de envío.
- **Después del envío**, el usuario puede recuperar la información borrada.

Si la página web incluye una prueba o examen:

- **Antes del envío**, el usuario puede revisar y corregir sus respuestas; o se le pide confirmación para continuar con la acción seleccionada.

Técnicas para mejorar la usabilidad

- Avisa al usuario si va a suceder una acción irreversible.
- Valida el formulario en el lado de cliente y avisa al usuario de los errores con un *alert()*.
- Si un campo tiene un error, sitúa el foco en él.
- En una lista desplegable, no repitas palabras o letras al inicio de cada elemento.
- Tanto si el formulario ha sido enviado con éxito como si no, házselo saber al usuario.

Ayuda

Se proporciona ayuda dependiente del contexto.

Técnicas para cumplir el criterio

Si es necesario que el usuario interactúe con un formulario:

- Ofrece instrucciones textuales al inicio del formulario o grupo de campos que describan la introducción necesaria.
- Ofrece un enlace de ayuda o ayuda mediante un asistente en cada página web.
- Para cada campo de entrada, comprueba la ortografía y sugiere cómo solucionarlo. Por ejemplo “¿Quisiste decir...?”.

Si es necesario que el usuario introduzca datos en un formato determinado en un campo del formulario:

- Avisa del dato esperado y ofrece un ejemplo.
- Ofrece instrucciones textuales al inicio del formulario o de un grupo de campos que describan la entrada de datos necesaria.

Técnicas para mejorar la usabilidad

- Usa el atributo *title* para presentar ayuda en el contexto.
- En los campos de texto, si aplica, chequea el carácter y conviértelo al deseado.

Prevención de errores en todo tipo de páginas

Para las páginas web que requieren que el usuario envíe información, se cumple al menos uno de los siguientes casos:

- El usuario puede deshacer o cancelar el envío.
- El sistema verifica la información para detectar errores en la entrada de datos y se proporciona al usuario una oportunidad de corregirlos.
- Se proporciona un mecanismo para que el usuario revise, confirme y corrija la información antes de enviar los datos.

Técnicas para cumplir el criterio

Simplemente sigue las técnicas del Criterio de Conformidad 3.3.4 para todos los formularios en los que el usuario envíe información.

Principio 4.

Robusto

Este es el principio más dependiente de la tecnología. Hace referencia a la capacidad del sitio web para ser interpretado por los agentes de usuario, por los productos de apoyo y por dispositivos de todo tipo, actuales y en sus futuras versiones.



Pauta 4.1

Respetá el código para que sea **compatible** con el mayor número de dispositivos y programas

Maximiza la compatibilidad con las aplicaciones de usuario actuales y futuras, incluidos los productos de apoyo.

Nuestro código debe ser tan limpio como sea posible, y también debemos respetar los estándares. De esta forma, los navegadores y otros agentes de usuario serán capaces de reproducir la página correctamente.

Esta pauta y sus puntos de control están directamente destinados a los desarrolladores.

Procesamiento, código limpio

En los contenidos implementados mediante el uso de lenguajes de marcas:

- los elementos tienen las etiquetas de apertura y cierre completas;
- los elementos están anidados de acuerdo con sus especificaciones;
- los elementos no contienen atributos duplicados; y
- los identificadores son únicos.

Excepto cuando las especificaciones permitan otras características.

Técnicas para cumplir el criterio

- Valida que cumples completamente la especificación. Puedes usar los validadores automáticos referenciados en el capítulo *Recursos y Herramientas*.
- Elige el estándar que deseas (HTML, XHTML, XML...), pero sigue su especificación, asegurando que la página se puede parsear porque está bien formada o porque al menos:
 - las etiquetas de inicio y cierre se usan según la especificación.
 - cada atributo *id* tiene un valor diferente en la página web.
 - los elementos no tienen atributos repetidos.

¡No lo hagas!

- No confundas al navegador con identificadores repetidos.
- No confundas al navegador con etiquetas de apertura o cierre o atributos incorrectos.

Nombre, función y valor

Para todos los componentes de la interfaz de usuario (como los elementos de formulario, enlaces y componentes generados por *scripts*):

- el nombre y la función pueden ser determinados por software;
- los estados, propiedades y valores que pueden ser asignados por el usuario pueden ser especificados por software; y
- los cambios en estos elementos se encuentran disponibles para su consulta por las aplicaciones de usuario, incluyendo los productos de apoyo.

Este criterio de conformidad se dirige principalmente a los desarrolladores que programan sus propios componentes de interfaz de usuario o modifican los controles estándares. Los controles estándares de HTML satisfacen automáticamente este criterio cuando se emplean de acuerdo con su especificación.

Técnicas para cumplir el criterio

Si usas componentes estándares de interfaz de usuario en un lenguaje de marcado:

- Utiliza el atributo *aria-label* para proporcionar una etiqueta invisible cuando no puedes usar una etiqueta visible.
- Usa el atributo *aria-labelledby* para proporcionar un nombre a los controles de la interfaz de usuario.
- Usa las características del marcado para exponer el nombre y la función; permite que las propiedades configurables las establezca directamente el usuario; y avisa de los cambios usando:
 - controles de formulario y enlaces en HTML.
 - elementos *label* para asociar etiquetas de texto a controles de formulario; o el atributo *title* para identificar los controles si no puedes usar el elemento *label*.
 - el atributo *title* de los elementos *iframe* y *frame*.
 - HTML según la especificación.

Si usas programación para modificar el comportamiento de un componente estándar de interfaz de usuario en un lenguaje de marcado:

- Expón el nombre y la función;
- Permite que las propiedades configurables las establezca directamente el usuario; y
- Avisa de los cambios usando *aria-labelledby* para proporcionar un nombre a los controles de la interfaz de usuario.

Si manipulas componentes estándares de interfaz de usuario con una tecnología de programación:

- Usa las características de accesibilidad de la API de esa tecnología para exponer nombres y funciones; y
- Permite que las propiedades configurables las establezca directamente el usuario; y
- Avisa de los cambios.

Si creas componentes de la interfaz de usuario con un lenguaje de programación:

- Hazlo usando una tecnología que soporte las funciones de la API de accesibilidad de las plataformas en las que los agentes de usuario se ejecutan para exponer los nombres y funciones; y
- Permite que las propiedades configurables las establezca directamente el usuario; y
- Avisa de los cambios usando:
 - los roles, estados y propiedades de ARIA para exponer la función y estado de los componentes de la interfaz de usuarios.
 - el atributo *aria-labelledby* para proporcionar un nombre a los controles de la interfaz de usuario.

Técnicas para mejorar la usabilidad

- Ofrece etiquetas en todos los controles de formulario que no tengan etiquetas identificativas implícitas.

¡No lo hagas!

- No programes de tal modo que los elementos *div* o *span* se comporten como controles de interfaz de usuario sin proporcionarles un rol.
- No implementes controles personalizados que no usen la API de accesibilidad de la tecnología, o lo hagan de manera incompleta.
- No te olvides de actualizar las alternativas textuales cuando cambie el contenido no textual.
- No asocies etiquetas y controles de interfaz de usuario en una forma que no pueda ser reconocibles mediante programación.
- No sitúes el foco en un componente de la interfaz de usuario de tal forma que no pueda ser determinable por software o en la que no esté disponible la notificación de cambio de estado del foco.
- Cuando la introducción de un dato se pide en varios campos separados, por ejemplo, el número de teléfono o de cuenta bancaria, no te olvides de ofrecer nombres para cada uno de estos campos.
- No te olvides de dar un nombre accesible a una imagen cuando sea el único contenido de un enlace. Puedes hacerlo con el atributo *alt*, *aria-label* o *aria-labelledby*.

Mensajes de estado [NUEVO 2.1]

En el contenido implementado en lenguajes de marcado, los mensajes de estado pueden ser determinados por software a través de su rol o propiedades, de tal modo que puedan ser presentados al usuario de productos de apoyo sin recibir el foco.

Los mensajes de estado informan al usuario sobre el éxito o el resultado de una acción, sobre el estado de espera de una aplicación, sobre el progreso de un proceso o sobre la existencia de errores. Algunos ejemplos son:

- Al pulsar en un botón de “Agregar al carrito de la compra” aparece un texto junto al carrito que reza “Carrito: 1 elemento”.
- Al equivocarse al introducir un email en un campo de formulario, aparece un mensaje de error que avisa de que “El formato de email es incorrecto”.
- Cuando un proceso tarda un tiempo, aparece un ícono de ‘ocupado’ en pantalla.
- Al enviar un formulario, el lector de pantalla anuncia “Formulario enviado”.
- La lista de resultados de una búsqueda no se considera una actualización de estado sin embargo el mensaje breve “Se devolvieron 18 resultados” sí.

El objetivo de este criterio es garantizar que los usuarios puedan ser conscientes de los cambios importantes en el contenido sin interrumpir excesivamente su trabajo o incluso que, mediante los productos de apoyo, el usuario pueda retrasar o suprimir esos mensajes de estado; o, por el contrario, pueda resaltar dichos mensajes cuando lo requiera. En cualquier caso, los **mensajes de estado no se entregan a través de un cambio en el contexto** y no capturan el foco.

Este criterio no afecta a informaciones importantes que se presentan en diálogos modales, que deben ser reconocidos por el usuario de forma inmediata y sí deben capturar el foco.

Técnicas para cumplir el criterio

Si el mensaje de estado advierte del éxito o el resultado de una acción:

- Indica que los datos se enviaron correctamente en combinación con el *role="status"*

Si el mensaje de estado transmite una sugerencia o una advertencia sobre la existencia de un error:

- Usa el *role="alert"* o *live regions* de ARIA para identificar los errores en combinación con alguna de estas técnicas:
 - describir en texto los campos obligatorios no completados, los campos que no se ajustan a los valores posibles, o a los campos llenos en formatos permitidos;
 - permitir saltar a los errores;
 - dar sugerencias de corrección; o
 - proporcionar una revisión ortográfica y sugerencias para la entrada de datos.

Si el mensaje de estado transmite información sobre el progreso de un proceso:

- Informa de los cambios usando los roles *log*, *status* o *progressbar*.

Técnicas para mejorar la usabilidad

- Usa *aria-live* en chats o en el contenido mostrado con *hover* y *focus*.
- Usa los roles *marquee* o *timer*.
- Mueve el foco al nuevo contenido incluso mediante el uso de *alertdialog* o *dialog*.
- Ofrece una opción a los usuarios para definir sus preferencias acerca del contenido que se actualiza automáticamente.

¡No lo hagas!

- No uses el rol *alert* o *aria-live=assertive* en contenido que no es crítico o sensible al tiempo.
- No uses un evento *visibilitychange* para ocultar o mostrar un documento sin cambiar las regiones que se modifican automáticamente de activas a inactivas o viceversa.

Documentos PDF accesibles

Los archivos enlazados en una página web para su consulta o descarga, como las hojas de cálculo o los archivos PDF, forman parte del contenido del sitio, y por tanto deben cumplir los mismos requisitos de accesibilidad que las páginas web.

Por su estandarización y popularidad, en este capítulo se explica cómo hacer un PDF accesible y cómo comprobarlo en una evaluación.



La accesibilidad de los documentos PDF

El formato PDF (Formato de Documento Portátil) permite combinar texto, vídeos, sonido, vínculos, formularios... Estos contenidos, como en una página web, deben cumplir los principios de las WCAG 2.1.

Como se explicó en la introducción del libro, la tecnología PDF es compatible con la accesibilidad, es decir, que los productos de apoyo, como un lector de pantalla o una línea braille, pueden comprender y proporcionar la información a sus usuarios si el documento PDF se ha elaborado de forma accesible. De esta manera, si se hace el documento PDF accesible de forma nativa, no habría que proporcionar una versión alternativa en HTML.

Por otra parte, en las declaraciones de conformidad no se pueden excluir tecnologías concretas, por ejemplo, no podemos decir que el sitio web es accesible doble-A respecto a las WCAG 2.1 salvo por los documentos PDF. Por lo tanto, si nuestros documentos PDF no son accesibles, nuestro sitio tampoco lo será.

Los criterios de conformidad de las WCAG 2.1 están redactados como enunciados verificables independientes de una tecnología concreta, de tal manera que pueden aplicarse a páginas web pero también a cualquier documento electrónico. Aunque los criterios de conformidad que deben cumplir los documentos PDF son los mismos que los que deben cumplir las páginas web, podemos distinguir entre:

- Los criterios **no aplicables** a los documentos PDF, como la necesidad de poder aumentar el tamaño de texto, puesto que la tecnología PDF admite *zoom*.
- Los criterios que se cumplen con **técnicas generales**, independientemente de la tecnología utilizada, como los requisitos de contraste de color, de no transmitir información solo por el color, de no dar instrucciones que dependan de características sensoriales, de presentación visual, de dividir el contenido en secciones con títulos o de que estos sean descriptivos, etc.
- Los criterios que se cumplen con **técnicas específicas** para la tecnología PDF. La mayoría de estas técnicas específicas **no producirán cambios visuales** en el documento, pues están ideadas para ayudar especialmente a las personas que acceden con un lector de pantalla o una línea braille. Sin embargo, estas técnicas tienen beneficios adicionales como mejorar la indexación y el **posicionamiento** del documento en los buscadores, así como su **usabilidad** para todos los usuarios.

Las técnicas específicas de PDF

Las WCAG 2.1 incluyen 24 técnicas específicas para la tecnología PDF (una de ellas nueva).

1. **Añade textos alternativos a las imágenes** para que las personas sin visión puedan percibirlas.
2. **Crea marcadores** para que todos los usuarios puedan desplazarse rápidamente.
3. **Garantiza el orden correcto de lectura y de tabulación**, de tal forma que, si se accede de forma lineal como con un lector de pantalla, el contenido tenga sentido.
4. **Oculta las imágenes decorativas** para no molestar o dificultar la compresión.
5. Indica los controles de formulario obligatorios para evitar errores a los usuarios.
6. **Utiliza el elemento *table* para maquetar datos tabulares** de tal manera que un lector de pantalla pueda leerlos de forma organizada.
7. **Procesa por OCR** un documento escaneado para obtener el texto real; de lo contrario se obtiene una imagen que los productos de apoyo no pueden interpretar.
8. **Define las abreviaturas** para que un lector de pantalla pueda leer su forma extendida.
9. **Incluye títulos** para poder entender la organización de los contenidos y que el usuario de lector de pantalla pueda “oír” el documento.
10. **Etiqueta los controles interactivos de formulario** de tal manera que se entienda para qué sirve cada control.
11. **Incorpora enlaces con un texto descriptivo** para que el usuario pueda decidir si pulsa o no, y pueda comprender adónde se dirige el enlace.
12. Indica el nombre, la función y el valor de los campos de formulario para que no queden dudas de cómo se rellena.
13. **Ofrece texto alternativo en los enlaces** cuando el texto del enlace no sea suficientemente autoexplicativo.
14. Introduce en cada página encabezados y pies que permitan al lector saber dónde está.
15. **Añade botones de “Enviar” a los formularios** para que sean los propios usuarios los que decidan cuándo se envía.
16. **Configura el idioma predeterminado** para que el lector de pantalla sepa cómo pronunciar, y los usuarios puedan comprenderlo.
17. **Numera las páginas coherentemente** para que los usuarios puedan ubicarse.
18. **Pon un título al documento**, es muy útil para entender el contenido del fichero.
19. **Especifica el idioma de un párrafo o frase** cuando sea diferente del idioma del documento, por el mismo motivo que es necesario configurar el idioma predeterminado.
20. Repara las tablas mal etiquetadas con el editor de tablas de “Adobe Acrobat Pro”, hasta el momento, la mejor herramienta.

21. **Utiliza las etiquetas *L* y *LI* para crear listas** y permitir al usuario desplazarse rápidamente por los contenidos.
22. **Avisa al usuario cuando se equivoque al llenar un campo** porque debe seguir un formato determinado o introducir valores dentro de un rango, de este modo, facilitas al usuario cumplimentar el formulario y le evitas cometer errores al llenar el formulario.
23. **Ofrece controles de formulario interactivos** asegurándote de que funcionan correctamente sólo con teclado.
24. [Nueva 2.1] **Utiliza el formato PDF/UA** al crear PDF. Se utiliza en el criterio 1.4.10 Reajuste de elementos.



Puedes encontrar más información de las técnicas PDF en
<http://www.w3.org/TR/WCAG20-TECHS/pdf.html>

Implementar las técnicas PDF

Implementar estas técnicas implica trabajar tanto en el documento de origen como en el propio PDF. Un **documento de origen** accesible ayudará a que, al guardarlo o exportarlo, el PDF resultante sea también mucho más accesible. Por desgracia, esto no siempre es suficiente, especialmente en documentos complejos, y a menudo es necesario **modificar el PDF** con una herramienta de edición específica para corregir determinados problemas. Estas modificaciones pueden ser mínimas, o incluso no ser necesarias, si se han seguido las buenas prácticas en el programa de origen y el documento es sencillo.

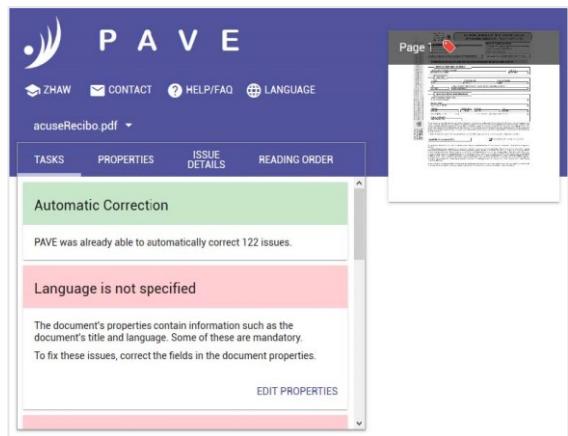
A continuación, se explica el proceso de cómo implementar las técnicas en el programa de origen y en un editor de PDF; luego, cómo revisar el documento PDF para comprobar su accesibilidad; y, por último, cómo exportar a PDF para no perder las mejoras de accesibilidad realizadas.

Para modificar el documento de origen hemos elegido Microsoft Word (aunque en ocasiones también se mencionan Open Office o Google Docs) y para modificar el propio PDF hemos elegido Adobe Acrobat Pro XI. Los pasos son comunes a otros programas (procesadores de texto, hojas de cálculo, programas de diseño...), pero se llevan a cabo desde diferentes opciones de menú.

Adobe Acrobat Professional²⁰ es la herramienta más recomendable actualmente para revisar y modificar la accesibilidad de un PDF, pero es un software propietario y de pago, diferente de Adobe Acrobat Reader. Aunque se puede trabajar la accesibilidad del PDF desde la versión de Adobe Acrobat Pro 6, se recomienda utilizar la versión XI o posteriores, por las mejoras significativas en sus herramientas específicas de accesibilidad.

Existe otra herramienta similar, *CommonLook PDF GlobalAccess*²¹, que es también local, de pago y bastante potente, y que permite revisar y modificar ciertos aspectos del PDF.

Como alternativa gratuita puedes probar PAVE²², una aplicación online que permite abrir y evaluar un PDF y corregir algunos de sus problemas de accesibilidad.



Captura de pantalla 1. PAVE

²⁰ <https://acrobat.adobe.com/es/es/acrobat/acrobat-pro.html>

²¹ <http://commonlook.com/accessibility-software/commonlook-pdf-globalaccess/>

²² <http://pave-pdf.org>

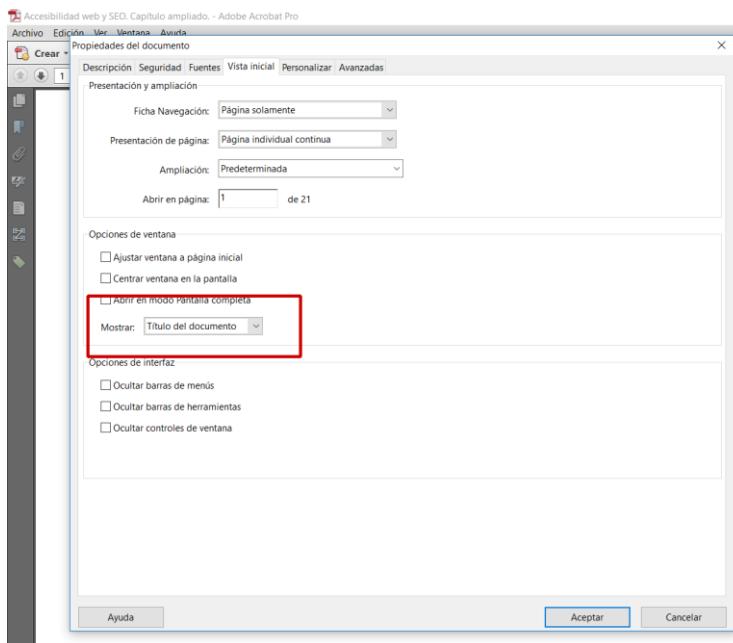
Título del documento

El título del documento es lo primero que anuncia el lector de pantalla al usuario, quien puede consultarla en cualquier momento con un atajo de teclado. En la mayoría de los programas de origen se puede indicar el título del documento en sus propiedades, de manera que este será exportado como título del PDF.

En Word, el título del documento se define en sus propiedades, en el menú “Archivo > Información > Propiedades > Título”.

En Acrobat, puedes hacerlo en el menú “Archivo > Propiedades > Título”.

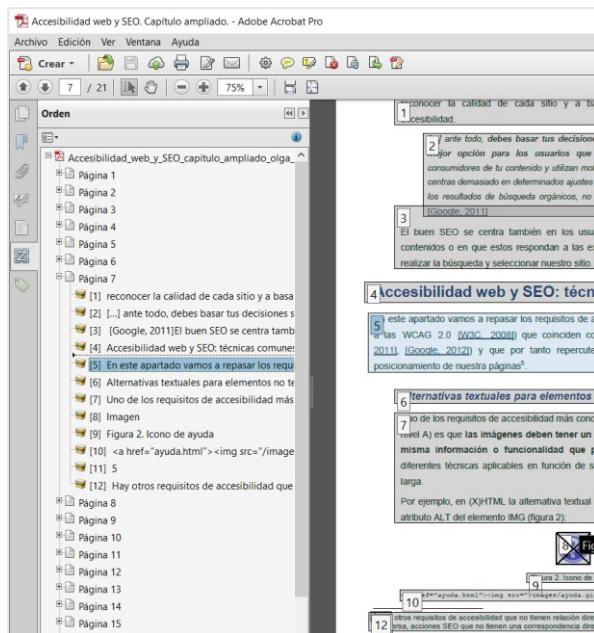
Además, debe indicarse en Acrobat que se muestre el título del documento en la cabecera del lector PDF, y no el nombre del fichero. Esto se define en el menú “Archivo > Propiedades > Vista inicial > Mostrar > Título del documento”.



Captura de pantalla 2. “Vista inicial” de las propiedades de un documento en Adobe Acrobat Pro XI

Orden de lectura

A menudo, el orden visual de los contenidos del documento no coincide con el orden de lectura del producto de apoyo. Cuando el contenido se lee en otro orden, el significado puede cambiar o resultar incomprendible. Para revisar el orden de lectura del contenido del PDF podemos escuchar el documento con un lector de pantalla. También podemos consultar el orden de lectura desde Acrobat con el panel “Orden” o con la herramienta “Retocar orden de lectura”, donde cada contenido de la página se muestra con un número. Este número indica el orden de lectura de cada elemento.



Captura de pantalla 3. Panel “Orden” de Adobe Acrobat Pro XI

El orden puede modificarse reordenando los contenidos en el árbol del panel “Orden”. Para conseguirlo, se selecciona un elemento y se arrastra a otra posición del árbol, de manera que la numeración de los contenidos cambiará para reflejar su nuevo orden de lectura. Estas modificaciones no deberían cambiar el aspecto visual del documento.

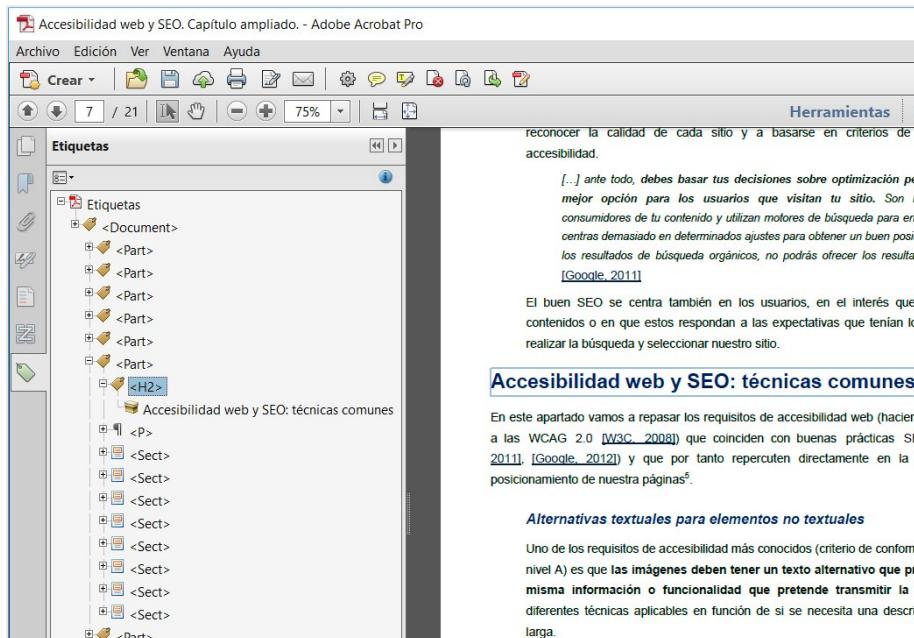
El orden de lectura incorrecto suele ser consecuencia de cómo se ha maquetado el contenido en el programa de origen. Una serie de **buenas prácticas** en Word para que el orden de lectura en el PDF sea el correcto y no haya que modificarlo son:

- No utilizar cuadros flotantes.
- Si se maqueta en varias columnas, no simularlas con tabuladores, cuadros flotantes o tablas sin bordes, sino utilizando la herramienta “Columnas”.
- Insertar las imágenes a medida que se redacta el texto, en línea con el contenido.

Etiquetado semántico

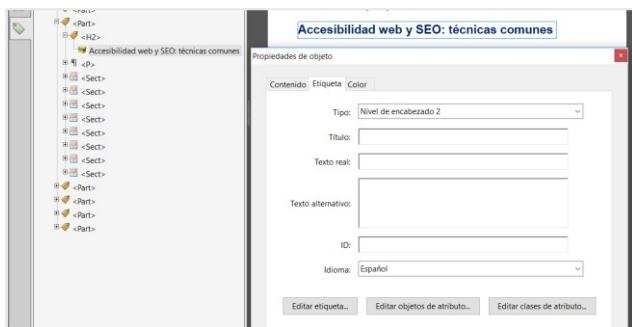
Cada contenido debe tener asociada internamente una etiqueta que indique qué tipo de contenido es: un encabezado, un párrafo, una lista, una tabla, una imagen, etc. Esto permite, entre otras cosas, que el lector de pantalla anuncie correctamente cada contenido según la etiqueta que tiene asociada, por ejemplo “Lista con cuatro elementos” o “Encabezado de primer nivel”. Además, facilita que los usuarios de lector de pantalla o de línea braille puedan “oír” el documento y saltar de unos contenidos a otros mediante los atajos del producto de apoyo.

En Acrobat Pro, el árbol de etiquetas se encuentra en el panel “Etiquetas”. Si el documento no se hubiera generado etiquetado, se puede crear el árbol de etiquetas automáticamente desde este panel.



Captura de pantalla 4. Panel “Etiquetas” de Adobe Acrobat Pro XI

Desde el árbol del panel “Etiquetas” puedes seleccionar cualquier etiqueta, acceder a sus propiedades e indicar, entre otras cosas, de qué tipo es.



Captura de pantalla 5. Propiedades de una etiqueta en Adobe Acrobat Pro XI



Descarga “*Cheat Sheet Etiquetas Estándar en PDF accesibles*”
https://www.usableyaccesible.com/archivos/AA_Cheatsheet_Etiquetas_Estandar_PDF_accesible.pdf
(PDF, 678 KB)

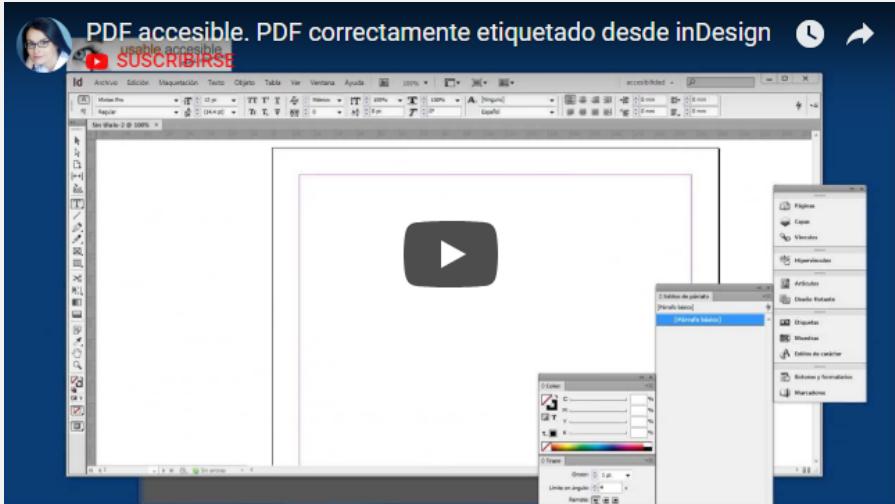
Si el documento se maqueta con las herramientas adecuadas del programa de origen, el etiquetado del documento será correcto y apenas será necesario retocarlo en Acrobat.

Las **buenas prácticas** a seguir son:

- Definir los títulos mediante estilos de título, nunca simularlos cambiando simplemente el tamaño o el color del texto.
- Asignar los títulos de forma coherente y no en base a su aspecto. No saltarse niveles.
- Definir las listas mediante la herramienta de lista con viñetas o lista numerada.
- No incluir retornos de carro para separar párrafos o saltar de página, ya que se exportan como párrafos vacíos que confunden a los usuarios de lector de pantalla. La separación entre los contenidos debe definirse mediante sus márgenes, espaciados e interlineados. De igual modo, los saltos de páginas deben incluirse con la herramienta “Salto de página”.
- Si se incluyen notas, un índice de contenidos, referencias bibliográficas, u otro contenido similar, se debe hacer siempre con la herramienta correspondiente.

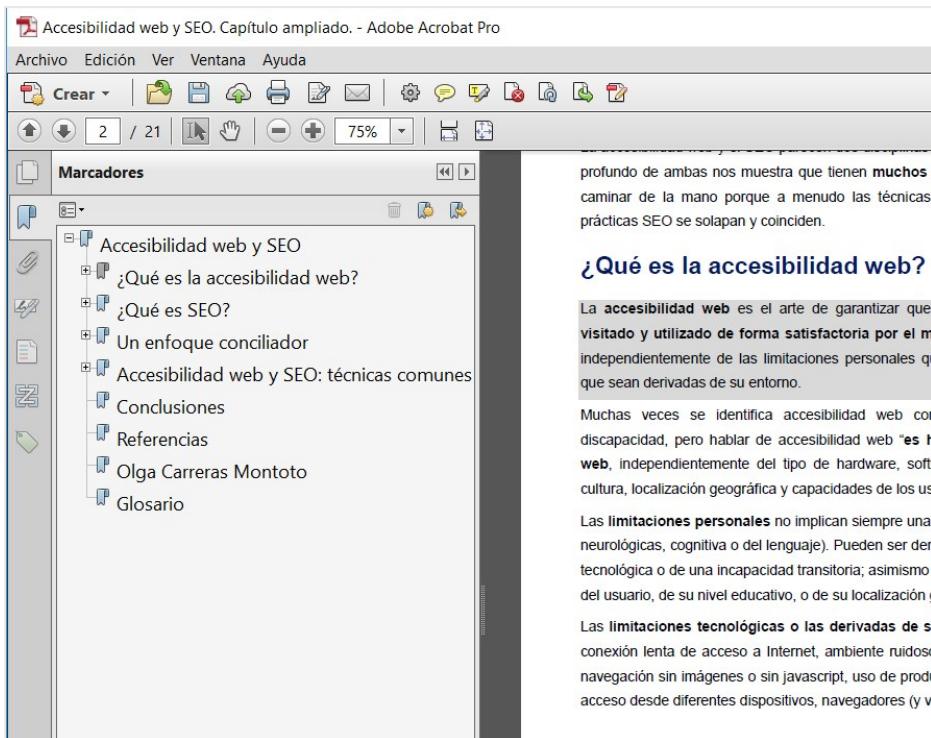


Si usas Adobe InDesign, consulta el video
“PDF accesible. PDF correctamente etiquetado desde InDesign”.
<https://olgacarreras.blogspot.com.es/2013/11/pdf-accesible-pdf-correctamente.html>



Marcadores

El panel “Marcadores” de Acrobat Pro incluye un índice del documento generado a partir de sus encabezados. Este panel es muy útil porque permite comprender la estructura del documento y saltar a cualquier sección del mismo.



Captura de pantalla 6. Panel “Marcadores” de Adobe Acrobat Pro XI

En las opciones de exportación del PDF se puede indicar que se genere el PDF con el índice de marcadores o se puede generar automáticamente desde el panel “Marcadores > Nuevos marcadores de estructura”.

Idioma del documento

Es muy importante indicar el idioma del documento y los cambios de idioma en el contenido para que el lector de pantalla pueda leer el documento adecuadamente.

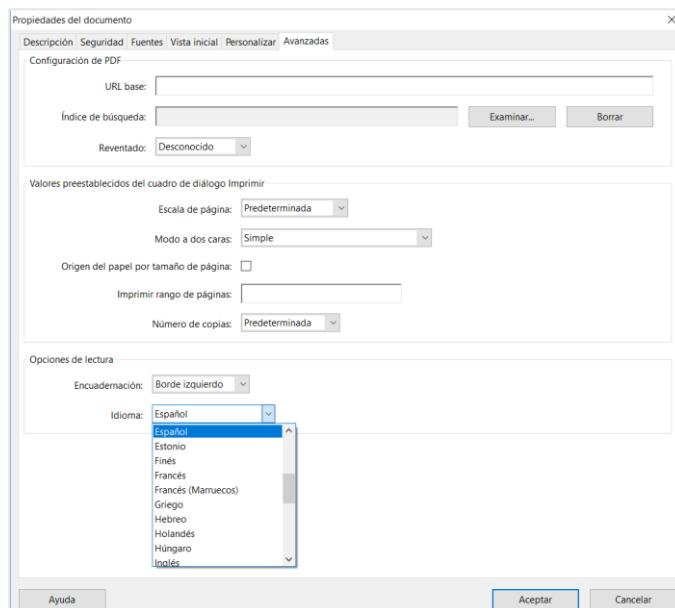
Para indicarlo en el programa de origen,

- en Microsoft Word se hace desde el menú “Revisar > Idioma”;
- en OpenOffice Writer desde el menú “Herramienta > Idioma”;
- en Google Docs desde el menú “File > Language”.

Sin embargo, es necesario revisar que esta información se ha exportado al PDF porque no siempre es así.

En Acrobat Pro, el idioma del documento se define en sus propiedades en el menú “Archivo > Propiedades > Avanzadas”. Si el idioma del documento no aparece en el desplegable, como ocurre por ejemplo con el catalán o el euskera, se puede escribir su código ISO, en este caso “ca” o “eu” respectivamente.

Si un contenido se encuentra en un idioma diferente al del idioma principal, se indicará en las propiedades de su etiqueta desde el árbol del panel “Etiquetas”.



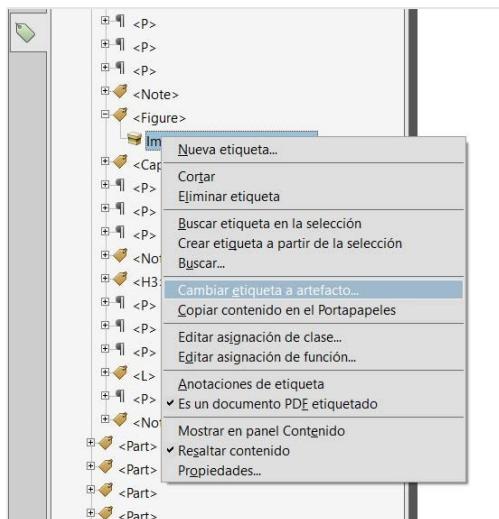
Captura de pantalla 7. Selección de idioma en “Propiedades del documento” con Adobe Acrobat Pro XI

Imágenes decorativas

Las imágenes decorativas son aquellas que no aportan ninguna información ni cumplen ninguna función y por tanto no es necesario que sean anunciadas por el producto de apoyo. Estas imágenes se etiquetan como “**artefacto**”.

En Acrobat Pro podemos convertir una imagen o cualquier otro contenido en artefacto, es decir, quitarlo del orden de lectura para que no sea anunciado, de dos maneras:

- seleccionando su etiqueta y pulsando la opción “Cambiar etiqueta a artefacto”.
- mediante la herramienta “Accesibilidad >Retocar orden de lectura”, seleccionando el contenido decorativo dentro del documento y pulsando “Fondo”.



Captura de pantalla 8. Convertir una etiqueta en “artefacto” con Adobe Acrobat Pro XI

Desde muy pocos programas se puede indicar que una imagen es decorativa, por eso en la mayor parte de los casos deberá establecerse desde Acrobat. Lo que sí puede hacerse en la mayoría de los programas es insertar la cabecera y el pie de los documentos con una opción de menú específica. Esto permitirá que sean exportados como “artefactos de página”, de tal manera que no sean anunciados en cada página sino solo bajo petición del usuario.



Nunca debemos eliminar un contenido del árbol de etiquetas.

Imágenes informativas

Las imágenes informativas son aquellas que tienen una función o comunican una información relevante. En estos casos debemos asociar a la imagen un texto alternativo que transmita la misma información o función a los usuarios de los productos de apoyo.

En Acrobat Pro, el texto alternativo de las imágenes se incluye en las propiedades de la etiqueta, en el campo “Texto alternativo”. Acrobat también dispone de una herramienta “Establecer texto alternativo” que permite recorrer todas las imágenes del documento para revisar y modificar su texto alternativo, o incluso indicar si son decorativas.

Desde la mayoría de los programas de origen se permite incluir textos alternativos a las imágenes y estos serán exportados con las imágenes al PDF. En Microsoft Office el texto alternativo de las imágenes se incluye en el campo “Texto alternativo > Descripción” de la ventana “Formato de la imagen”. Sin embargo, en OpenOffice se añade en el campo “Título” (no en el campo “Descripción”) de la ventana “Descripción” de la imagen. En Adobe InDesign se define en el menú “Objeto > Opciones de exportación de objetos > Texto alternativo > A medida”.

Paginación consistente

La paginación del documento, es decir, el número de página que aparece en la cabecera o el pie del documento debe coincidir con el número real de página. El objetivo es que la navegación por la barra de paginación de Adobe sea precisa, comprensible y consistente con la paginación visual y el posible índice de contenidos del documento.

Si ambas paginaciones no coinciden, se pueden cambiar los números de página desde el programa de origen, o bien modificar la paginación en Acrobat desde el panel “Miniaturas de páginas”. Para ello, se selecciona la página o rango de páginas a las que se desea cambiar la paginación y se accede a la opción “Numerar páginas”.

Destino de los enlaces

Es recomendable incluir los hipervínculos en el documento de origen para que se exporten al PDF. Los textos de los enlaces deben ser significativos fuera de contexto, y en caso de no serlo, deberemos clarificar su destino con un texto alternativo.

En Acrobat Pro, la manera de incluir un texto alternativo a un enlace es mediante el campo “Texto alternativo” de las propiedades de su etiqueta. Hay que tener en cuenta que **este texto alternativo sustituirá al texto del enlace**, al contrario que en una página web, donde el texto alternativo del enlace se lee complementariamente al texto del enlace. Es decir, en un PDF el producto de apoyo anunciará el texto alternativo del enlace en vez de su texto. Por esta razón, hay que asegurarse de que el texto alternativo tiene toda la información necesaria, y de que tiene sentido al ser leído junto al texto que le precede y le sigue.

Hay programas como Microsoft Word que permiten asociar un texto alternativo a un enlace (opción “Info. en pantalla” de la ventana “Modificar vínculo”). Sin embargo, esa información no se exporta al documento PDF, con lo cual hay que revisarlo en Adobe Acrobat.

Tablas

Las tablas de datos pueden resultar difíciles de comprender y de acceder con un producto de apoyo si su estructura es complicada. Por ello, siempre debemos preguntarnos primero si se pueden simplificar, dividirlas en varias más sencillas o presentar la información en otro formato. Por la misma razón es recomendable no combinar celdas, lo que hará que las tablas sean regulares y por tanto más fáciles de comprender y acceder.

Para que las tablas sean accesibles deben cumplir tres requisitos:

Requisito 1. Tener una descripción

Aunque la descripción no se visualiza, será anunciada por el lector de pantalla o la línea braille para que el usuario comprenda la estructura de la tabla.

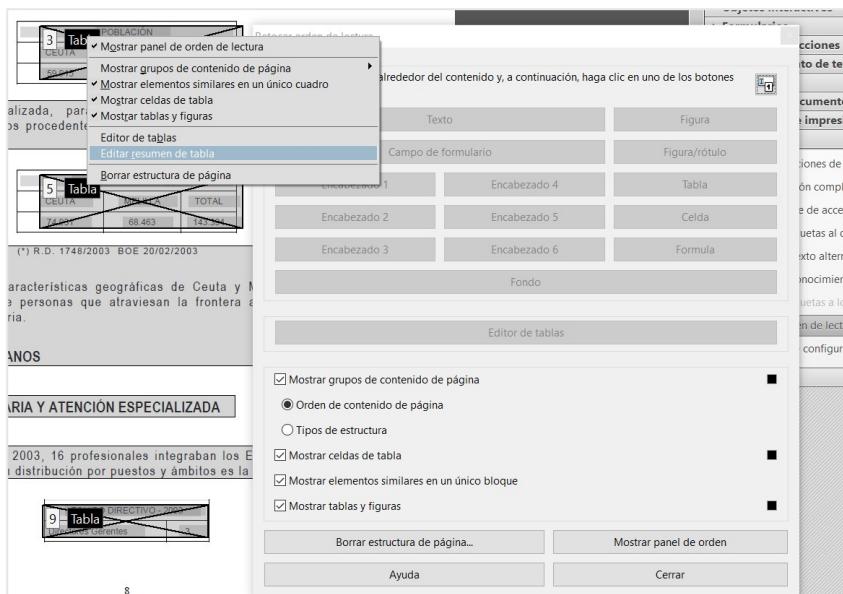
Por ejemplo, en la siguiente tabla, una descripción adecuada sería: “Tabla de tareas. Elige una tarea en la primera columna y en la misma fila encontrarás el número de horas y el precio en euros asociado a esa tarea. En la última fila se indica la suma total de horas y la suma total del precio.”

Tarea	Número de horas	Precio (Euros)
Tarea 1	8	400
Tarea 2	2	100
Tarea 3	4	200
Total	14	700

No es necesario indicar cuántas columnas o filas tiene la tabla puesto que esto ya lo anuncia por defecto el producto de apoyo.

En algunos programas puede asociarse una descripción a la tabla. En Microsoft Word se incluye desde la ventana “Propiedades” de la tabla, sin embargo, la descripción no se exporta al PDF.

En Acrobat Pro, la descripción de la tabla se incluye seleccionando la tabla con la herramienta “Accesibilidad > Retocar orden de lectura” y la opción “Editar resumen de tabla”.



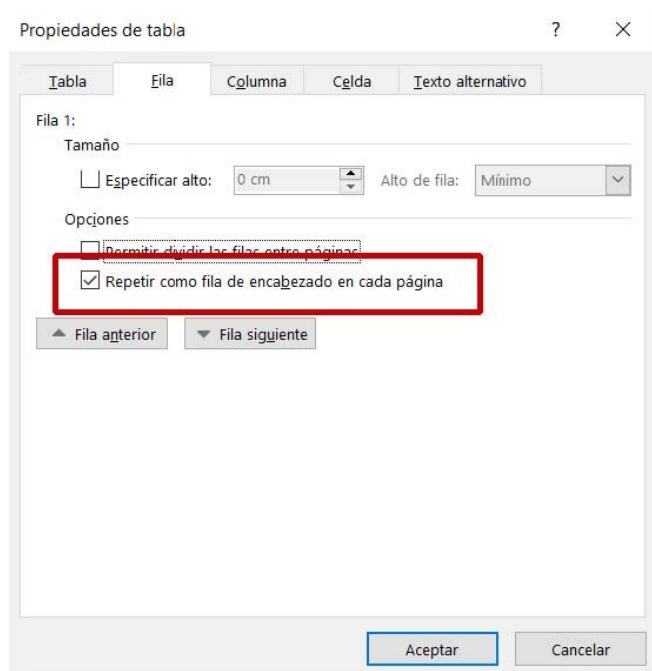
Captura de pantalla 9. “Editar resumen de tabla” en Adobe Acrobat Pro XI

Requisito 2. La tabla debe estar correctamente etiquetada

La tabla tiene que estar correctamente etiquetada, es decir, se debe indicar qué celdas son de encabezado (con la etiqueta TH) y qué celdas son de datos (con la etiqueta TD). Podemos revisar y modificar la estructura de la tabla desde el árbol de etiquetas o desde el “Editor de tablas”, al que se accede desde el menú contextual de la tabla.

Las buenas prácticas en el programa de origen para que las tablas se generen con un etiquetado correcto son:

- Insertar las tablas con la herramienta “Tabla”.
- Indicar la fila de encabezado. Por ejemplo, en Microsoft Word se define con la opción “Diseño > Fila de encabezado/ Primera columna” y en “Propiedades de la tabla > Fila > Repetir como fila de encabezado” se indica que se repita en todas las páginas cuando la tabla ocupe varias páginas. Sin embargo, no se pueden identificar varios niveles de celdas de encabezado, esto debería hacerse desde Acrobat o simplificar la tabla.



Captura de pantalla 10. Propiedades de la fila de una tabla en Word 2016

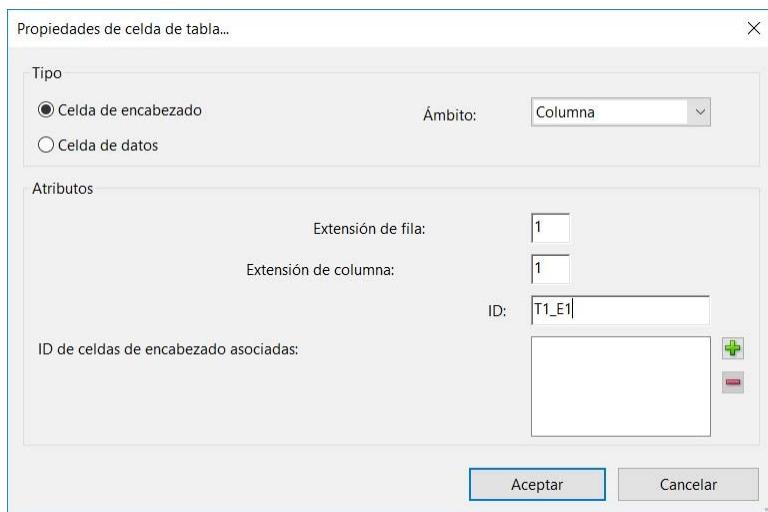
Requisito 3. Indicar la relación entre las celdas de encabezado y las celdas de datos

Si la tabla es compleja, con más de un nivel de encabezados, se debe asociar cada celda de datos con todas las celdas que la encabezan. Esto solo puede hacerse desde Acrobat y es bastante laborioso, así que se aconseja siempre intentar primero simplificar la tabla.

Definir la relación entre las celdas de datos y las celdas de encabezado se realiza con el “Editor de tablas”, al que se accede desde el menú contextual de la tabla.

Mediante la opción “Editor de tablas > Propiedades de celda de tabla” debemos:

- Dar a cada celda de encabezado un ID único en el documento.
- Indicar en cada celda de datos los ID de todas las celdas que la encabezan.



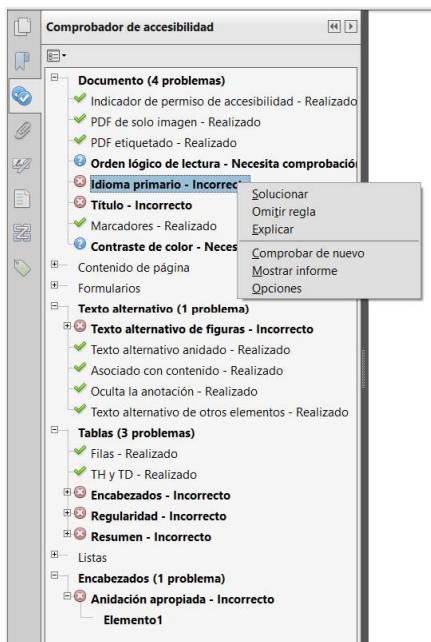
Captura de pantalla 11. “Editor de tablas” de Adobe Acrobat Pro XI

Comprobar el documento de origen y el PDF

Antes de exportar el documento a PDF hay que comprobar si está todo correcto o tiene algún problema de accesibilidad.

Desde la versión 2010 de **Microsoft Office** puedes revisar tu documento con el validador de accesibilidad que se encuentra en “Archivo > Comprobar si hay problemas > Comprobar accesibilidad”. Este validador detecta errores como imágenes sin texto alternativo, retornos de carro innecesarios o tablas en las que no se especifica la fila de encabezado. Es muy recomendable que verifiques si este validador detecta problemas en tu documento y los soluciones antes de exportarlo a PDF.

Adobe Acrobat Pro tiene un validador de accesibilidad denominado “Comprobación completa”. Aunque no puede evaluar todos los requisitos de accesibilidad que debe cumplir el PDF, es importante comprobar que no encuentra ningún error y corregir los detectados.



Captura de pantalla 12. Resultados del validador de accesibilidad de Adobe Acrobat Pro XI

Adobe Acrobat Pro tiene también la herramienta “Hacer accesible” dentro del “Asistente de acciones”. En ningún caso convierte el PDF en accesible, sino que, a modo de asistente, nos guía por las acciones que debemos realizar.



Otras herramientas que podemos encontrar son **validadores automáticos**, que detectan determinados errores de accesibilidad del PDF (nunca todos) pero no permiten corregirlos.

Aunque el más completo es el de Acrobat Pro, puedes probar **PAC 3** (*PDF Accessibility Checker*), un programa gratuito que facilita llevar a cabo evaluaciones de accesibilidad.

<http://www.access-for-all.ch/en/pdf-lab/pdf-accessibility-checker-pac.html>

Puedes consultar un listado de validadores de PDF en “Validadores y herramientas para consultorías de accesibilidad y usabilidad. Validar PDF” de Olga Carreras.

https://www.usableyaccesible.com/recurso_misvalidadores.php#accesibilidadpdf

Exportar a PDF

Una vez validado el documento se puede guardar o exportar como PDF. Dependiendo del programa y versión, las opciones de exportación que se deben marcar son:

- Exportar como “PDF etiquetado” o “PDF con etiquetas”, de modo que se genere con el árbol de etiquetas.
- “Exportar marcadores” o “Crear marcadores”, para que cree automáticamente el índice de marcadores a partir de los encabezados del documento.
- “Agregar vínculos” o “Convertir vínculos” para que conserve los enlaces del documento.

Se debe elegir una versión posterior a Acrobat 6.0 / PDF 1.5, a partir de la cual se incluyeron las opciones de etiquetado del contenido. Lo más habitual es seleccionar compatibilidad desde Acrobat 8 / PDF 1.7

Por último, como se indicaba anteriormente en las técnicas, **un documento escaneado como imagen nunca podrá ser un documento accesible**: no se puede buscar dentro de ese texto, los productos de apoyo no pueden interpretar su estructura ni acceder a su contenido, al aumentar el zoom el texto se verá *pixelado*, etc. Por eso es imprescindible convertirlo en texto con una herramienta de reconocimiento de caracteres (OCR). El propio Adobe Acrobat Pro incluye uno en el menú “Herramientas> Reconocimiento de texto> En este archivo”.

¿Qué tipos de formatos PDF usar?

Aunque el estándar PDF fue desarrollado por Adobe, desde 2008 es un estándar formal y abierto publicado por la ISO como ISO 32000²³. Existen diferentes subestándares de PDF: PDF/A, PDF/X, PDF/E y PDF/UA, que definen un PDF que debe cumplir ciertos requisitos.

Es una confusión habitual creer que un **PDF/A** es un PDF accesible. La opción de guardar o exportar un PDF como PDF/A es una opción habitual en muchos programas. Un PDF/A no es un PDF accesible, simplemente es un PDF que está pensado para que se conserve igual a largo plazo, es decir, que pueda ser reproducido con exactitud en el futuro. Por ello tiene que cumplir con ciertos requisitos, como estar etiquetado o ser autocontenido, es decir, que las fuentes, las imágenes, etc. estén embebidas en el documento.

Por el contrario, el estándar **PDF/UA** (*PDF/Universal Accessibility*) sí que define las características que debe cumplir un PDF para ser accesible. Los requisitos de accesibilidad del estándar PDF/UA son equivalentes a los definidos por las WCAG 2.1 más **tres requisitos adicionales**:

1. **Todas las fuentes deben estar embebidas** en el PDF, como es habitual y como son exportados por defecto los documentos.
2. **Todos los caracteres deben tener su correspondencia con caracteres Unicode**, lo cual asegurará que el lector de pantalla puede interpretar y leer todo el texto correctamente. Este requisito es verificado por el validador de accesibilidad de Adobe Acrobat Pro.

²³ <https://www.iso.org/standard/63534.html>

3. Las **opciones de seguridad del documento** no deben prohibir a los productos de apoyo acceder al contenido. En Acrobat Pro se especifica en “Archivo > Propiedades > Seguridad”. En las opciones de exportación de algunos programas, como Word o InDesign, también se puede definir la seguridad del documento. Este requisito es verificado por el validador de accesibilidad de Adobe Acrobat Pro.

Hoy en día no se puede guardar o exportar un PDF como PDF/UA, y aunque se pudiera, por sí mismo no podría garantizar que el PDF es efectivamente accesible, puesto que muchos de los requisitos no pueden comprobarse, ni mucho menos solucionarse, automáticamente.



Puedes ampliar información sobre el estándar PDF/UA en “PDF/UA. Descripción de la norma. Comparativa y relación con las WCAG” de Olga Carreras

<https://olgacarreras.blogspot.com.es/2012/09/pdfua-descripcion-de-la-norma.html>

Guías y enlaces de interés

“Guía para elaborar documentación digital accesible. Recomendaciones para Word, PowerPoint y Excel de Microsoft Office 2010” de Lourdes Moreno, Paloma Martínez y Yolanda González.

<http://www.centac.es/es/tecnologia/guia-acces-office>

“Accesibilidad en PDFs” del Observatorio de Accesibilidad.

https://administracionelectronica.gob.es/pae/Home/pae_Estrategias/pae_Accesibilidad/pae_documentacion/pae_eInclusion_Accesibilidad_de_PDF.html#.WvFrTEiFOUk

“Accesibilidad en documentos PDF, ePub, Office y OpenOffice” de Olga Carreras.
https://olgacarreras.blogspot.com.es/2009/04/dos-anos-de-usable-y-accesible.html#.accesibilidad_pdf

ARIA, el aliado (casi) desconocido

Los sitios y aplicaciones web cuentan muchas veces con controles no nativos, habitualmente desarrollados con JavaScript, que los lectores de pantalla no pueden interpretar correctamente, por lo que sus usuarios no logran comprenderlos y usarlos.

Para solucionar este problema, el W3C desarrolló la tecnología ARIA (*Accessible Rich Internet Application*), también conocida como WAI-ARIA, que permite añadir información semántica a cualquier elemento de la interfaz. De este modo el navegador transmite esa información al producto de apoyo, y este al usuario, para que así pueda interaccionar con el sitio o la aplicación web.



Qué es ARIA

En el mundo ideal, los elementos HTML se usan y funcionan de la manera en la que están pensados: un botón es un botón, y un enlace es un enlace. Sin embargo, HTML no tiene una etiqueta para definir un árbol desplegable, unas pestañas, una ventana modal, o un mensaje de alerta. Para crear estos nuevos componentes utilizamos elementos estándar de HTML más programación JavaScript (o AJAX o cualquier *framework* JavaScript).

Otras veces, utilizamos elementos HTML para una función que no es para la que fueron diseñados, por ejemplo, podemos hacer que un *div* se comporte como un botón dándole la apariencia de un botón con la CSS y añadiéndole eventos JavaScript.

Las personas que pueden ver la pantalla comprenderán, por la representación visual del elemento, que el *div* es un botón, o verán que una lista es un árbol desplegable que está abierto o cerrado.

Pero el usuario de producto de apoyo, como un lector de pantalla o una línea braille, no lo tiene tan fácil. Para comprender el por qué, es necesario que expliquemos primero brevemente **cómo funciona un producto de apoyo**.

El navegador expone a la API de accesibilidad del sistema operativo la información sobre la interfaz, y a su vez, el lector de pantalla extrae de la API de accesibilidad esta información y se la anuncia al usuario. Si lo que tenemos es una lista de elementos (*ul*), por mucho que se esté comportando como un árbol desplegable, el navegador la expondrá a la API como una lista, y a su vez el producto de apoyo tomará esta información y le anunciará al usuario que es una lista de elementos. En este ejemplo, el lector de pantalla NVDA diría “Lista con 3 elementos”.

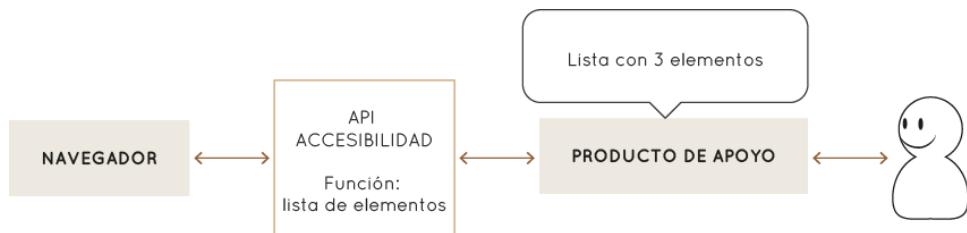


Ilustración 12. Esquema de comunicación entre el navegador y el producto de apoyo a través de la API de accesibilidad. Ejemplo de lista de elementos.

¿Cómo hacemos que el producto de apoyo anuncie al usuario que la lista es un árbol desplegable? Para ello se creó WAI-ARIA, o ARIA a secas (*Accessible Rich Internet Applications*), una taxonomía de roles, estados y propiedades que ayudan a definir los elementos de la interfaz.

En el caso del árbol desplegable, gracias a ARIA podemos definir cuál es la nueva función de la lista y podemos añadirle diferentes propiedades, por ejemplo, el nombre con el que queremos que se anuncie el árbol, que se indique si está abierto o cerrado, etc.

Listado de ficheros

- ⊕ Proyectos
- ⊕ Recursos
- ⊕ Presupuestos

```
<ul role="tree" aria-label="Listado de ficheros">
<li role="treeitem" aria-expanded="false"
    aria-posinset="1" aria-setsize="3" aria-level="1"
    tabindex="0">
    Proyectos
    [...]
</ul>
```

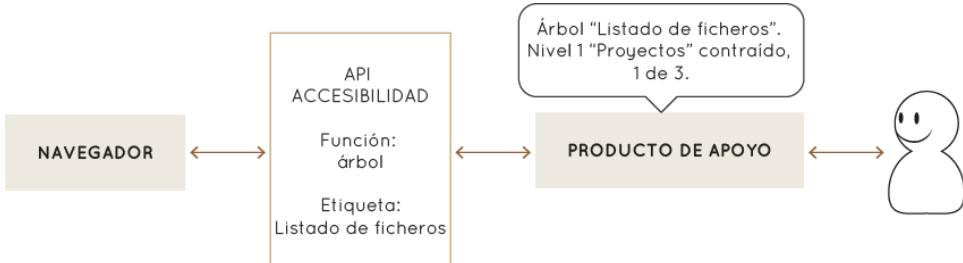


Ilustración 13. Esquema de comunicación entre el navegador y el producto de apoyo a través de la API de accesibilidad. Ejemplo de árbol definido con ARIA.

De esta manera, el usuario de producto de apoyo tendrá toda la información para poder comprender y manejar el árbol. En este caso, el lector de pantalla NVDA diría “Árbol Listado de ficheros Nivel 1 Proyectos contraído, 1 de 3”.

Por tanto, se podría definir ARIA como un **conjunto de atributos que se añaden a las etiquetas HTML** para que los agentes de usuario (navegadores y productos de apoyo) comprendan que estas etiquetas tienen un comportamiento diferente al habitual y actúen en consecuencia.

ARIA nació en 2008 y actualmente la última recomendación es WAI-ARIA 1.1 de 2017. Es una tecnología poco conocida e implementada por los desarrolladores web, posiblemente porque, como hemos visto, solo conociendo la problemática de las personas que acceden con productos de apoyo y utilizando un lector de pantalla, puedes realmente comprender su importancia e implementarla correctamente.

Actualmente su soporte es alto por parte de los navegadores y los productos de apoyo²⁴. Además, las WCAG 2.1 proponen usar ARIA como una forma óptima para cumplir con sus criterios de conformidad, ya que, bien utilizada, permite que diseños y funcionalidades muy complejos sean perfectamente accesibles.

Con ARIA se pueden crear controles complejos, como barras de progreso, deslizadores, desplegables, *tooltips* flotantes, alertas, *pop-hover* con opciones de diálogo, ordenación de listas de elementos, árboles de contenido contraíbles y desplegables, elementos sobre los que se puede hacer *drag-and-drop*. También se puede indicar la obligatoriedad de un campo de formulario o que este ha dado error de validación.

²⁴ Consulta el estado actual de soporte de esta tecnología en <https://caniuse.com/#feat=wai-aria>

Con ARIA puedes definir áreas de contenido que se actualizan sin intervención del usuario (como un *timeline* de Twitter, un reloj que indique el tiempo que nos queda para completar un formulario, o un mensaje de error al llenar un formulario) de tal forma que las actualizaciones no le pasen desapercibidas al usuario que no puede verlas. Además, podemos configurar cómo queremos que le sean anunciadas al usuario.

ARIA, aunque no lo tratamos en el libro, también permite dotar de accesibilidad a los contenidos SVG.

ARIA puede ser usado tanto en HTML 5 como en XHTML o HTML4, solo que, en estos dos últimos casos, si no quieres que el validador de sintaxis del W3C dé error de validación, debes usar un DOCTYPE específico:

En XHTML:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML+ARIA 1.0//EN"
"http://www.w3.org/WAI/ARIA/schemata/xhtml-aria-1.dtd">
```

En HTML 4:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML+ARIA 1.0//EN"
"http://www.w3.org/WAI/ARIA/schemata/html4-aria-1.dtd">
```



Puedes leer la especificación completa WAI-ARIA 1.1 en
<https://www.w3.org/TR/wai-aria>

Ejemplo 1. Cambiar el funcionamiento de un objeto: una capa que se comporta como un botón

Imagina que al final de un formulario de nuestro sitio web, en lugar de poner un botón de “Enviar” queremos poner una imagen.

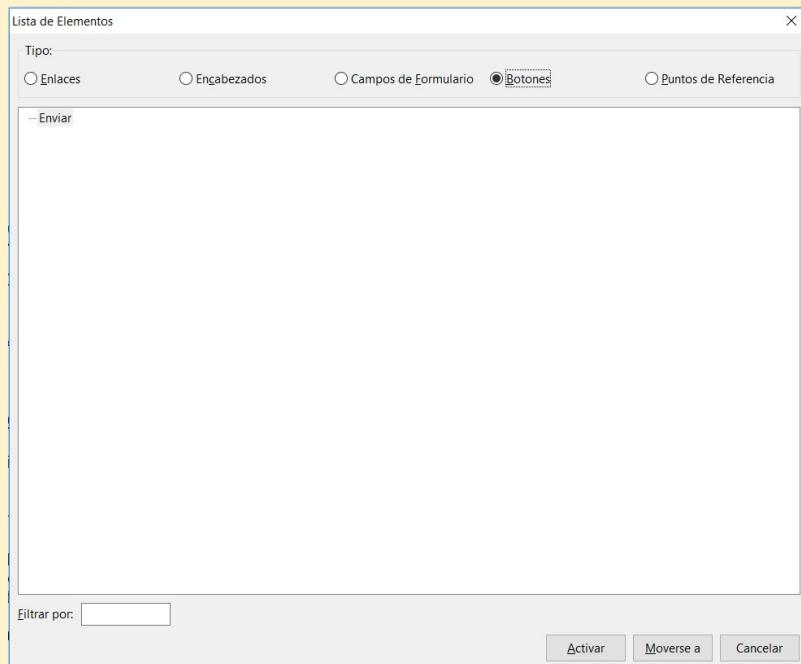
```
<div></div>
```

En vez de usar la etiqueta nativa `<button>` vamos a utilizar la etiqueta `<div>` y tenemos que hacer que se comporte como lo haría un `<button>`.

Para informar al navegador de ese cambio de funcionalidad del `<div>`, y que éste lo traslade al lector de pantalla, debemos añadirle el atributo `role=button`.

```
<div role="button">
    
</div>
```

A partir de este momento, para el lector de pantalla es un botón: lo anuncia como tal (“Botón gráfico Enviar”), aparecerá en el listado de campos de formulario del lector, o podremos llegar hasta él pulsando el atajo de teclado correspondiente del lector (habitualmente la letra “b”).



Captura de pantalla 13. Listado de botones de NVDA (insert+F7)

Sin embargo, solo por decirle que es un botón, no va a comportarse como tal, tendremos que añadir su comportamiento incluyendo los eventos correspondientes con *JavaScript* no intrusivo.

Aunque le asociemos un evento *onclick* o *onkeypress*, el *<div>* no cogerá el foco cuando naveguemos con el tabulador del teclado, como sí lo haría un *<button>*. Por tanto, no será accesible por teclado, solo por ratón.

Para conseguir que coja el foco tendremos que añadirle el atributo *tabindex=0*, que indica al navegador que incluya el *<div>* en la lista de elementos con los que el usuario puede interactuar. De este modo, podemos utilizar la tecla TAB para llegar hasta el elemento *<div>* (en el orden que le corresponda en el DOM), y la tecla ESPACIO o ENTER para pulsarlo.

El código final mejorado quedaría de la siguiente manera:

```
<div role="button" tabindex="0" id="buttonEnviar">
    
</div>
```

El ejemplo que acabamos de ver es un ejemplo de algo que se puede hacer, pero que ¿debería hacerse? Si podemos usar la etiqueta nativa, lo más conveniente es usarla. ARIA es un complemento que sirve para mejorar la accesibilidad de las páginas, no un sustituto *per se* de los controles nativos.

Los cuatro casos en los que se recomienda utilizar ARIA en lugar de elementos HTML nativos son:

1. La característica no está disponible en HTML.
2. La característica está disponible en HTML, pero no está implementada en los agentes de usuario.
3. La característica está disponible e implementada en HTML, pero el agente de usuario no proporciona el soporte para la accesibilidad de ese elemento.
4. El diseño visual ‘obliga’ a determinado estilo, pero no podríamos decorar un elemento nativo con ese diseño visual.

Roles

Los roles en ARIA permiten indicar la función de un elemento de la interfaz. La función se asigna con el atributo *role*. Hay 81 roles categorizados de la siguiente manera:

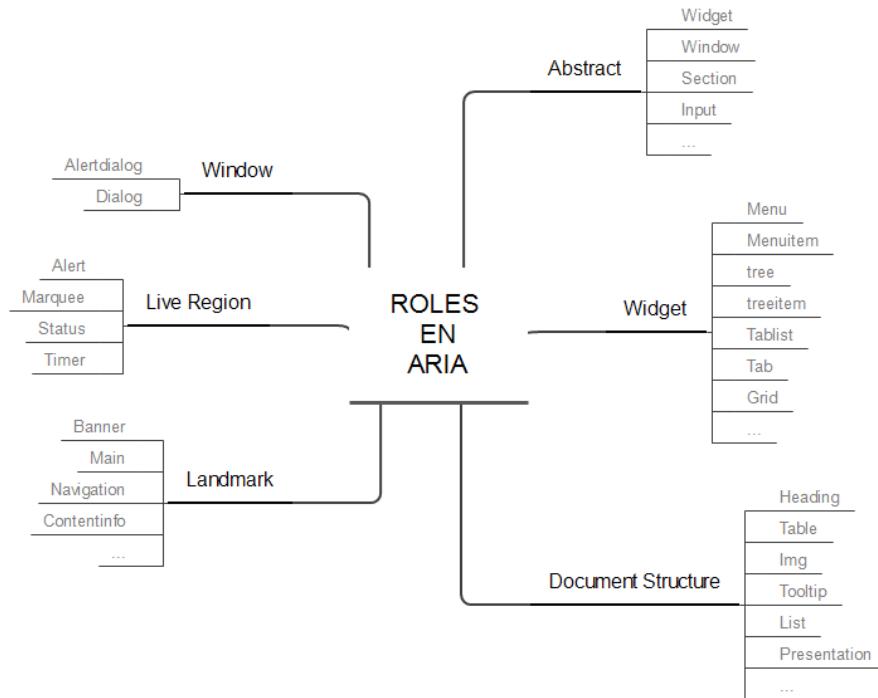
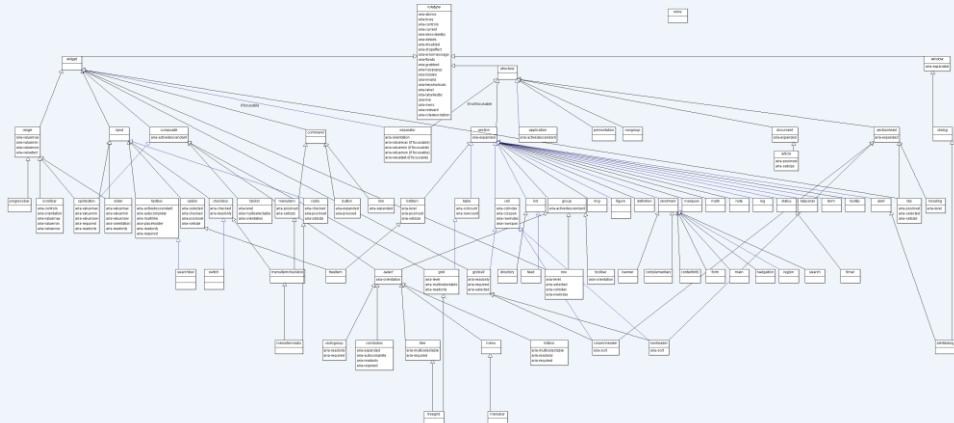


Ilustración 14. Esquema de Roles en Aria

- **Abstract:** *widget, window, section, input*, etc. No se usan en el contenido, sino para definir dentro de la ontología tipos de roles generales.
- **Widget:** *menu, menuitem, tree, treeitem, tablist, tab, tabpanel, button, grid*, etc.
- **Document Structure:** *heading, table, img, tooltip, list, presentation*, etc. No suelen ser interactivos.
- **Landmark:** *banner, main, navigation, contentinfo*, etc. Permiten definir las grandes zonas de la página igual que las etiquetas semánticas de HTML5 (*header, main, nav, footer*, etc.). Los usuarios de lector de pantalla tienen atajos para saltar de zona en zona o sacar un árbol de la estructura de la página generado a partir de esta información.
- **Live Region:** *alert, log, marquee, status* y *timer*. Definen la función de las zonas “vivas” de la página, es decir, que cambian automáticamente sin intervención del usuario.
- **Window:** *alertdialog* y *dialog*, para las capas que abrimos a modo de ventanas.

Puedes acceder a la **definición de cada rol** y su detalle en el apartado “*Definition of Roles*” de la especificación WAI-ARIA 1.1 http://www.w3.org/TR/wai-aria-1.1/#role_definitions



También puedes descargar el gráfico esquemático de **relaciones entre los roles**.

http://www.w3.org/TR/wai-aria-1.1/img/rdf_model.html

Por último, puedes aprender más sobre los **roles *landmark*** y su convivencia con las **etiquetas semánticas** de HTML 5 en <https://olgacarreras.blogspot.com.es/2014/03/navegacion-mas-accesible-y-semantica-en.html>

Estados y propiedades

Además de los roles, ARIA define los estados y propiedades de los diversos controles. La diferencia conceptual entre “estado” y “propiedad” es muy sutil: las propiedades suelen cambiar menos (aunque no siempre) que los estados, que cambian con frecuencia debido a la interacción del usuario.

Pero en la práctica no es necesario diferenciarlos y todos ellos comenzarán por *aria*.

Hay 48 estados y propiedades y se dividen en cuatro categorías:

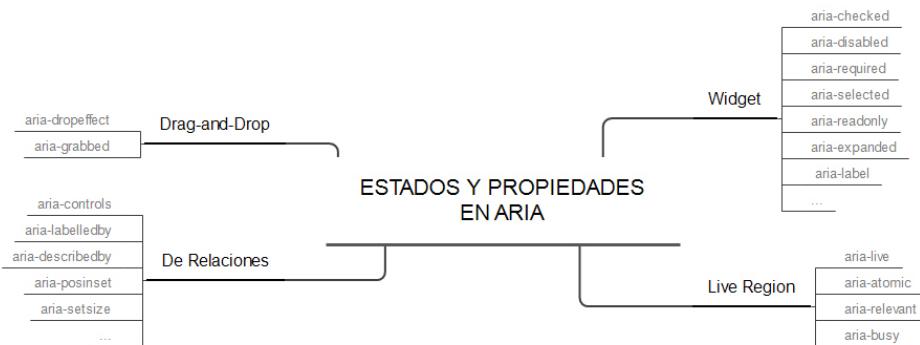


Ilustración 15 Esquema de estados y propiedades en ARIA

- **Atributos de Widget:** *aria-checked*, *aria-disabled*, *aria-required*, *aria-selected*, *aria-readonly*, *aria-expanded*, *aria-label*, etc.
- **Atributos de Live Region:** *aria-live*, *aria-atomic*, *aria-relevant* y *aria-busy* que permiten definir cuándo se anunciarán al usuario de producto de apoyo los cambios producidos en las zonas que se actualizan solas, qué parte se anunciará, qué tipo de actualización queremos que se anuncie o si queremos que temporalmente dejen de anunciarse²⁵.
- **Atributos de Drag-and-Drop:** *aria-dropeffect* y *aria-grabbed*
- **Atributos de relaciones:** es decir, que expresan relaciones o asociaciones entre los elementos que no se pueden determinar fácilmente a partir de la estructura del documento: *aria-controls*, *aria-labelledby*, *aria-describedby*, *aria-posinset*, *aria-setsize*, etc.

Los estados y propiedades pueden cambiarse con *JavaScript*, de hecho, será muy importante cambiarlos cuando el usuario interactúe con ellos. Si el usuario abre o cierra el árbol desplegable gracias a los eventos *JavaScript* asociados, las funciones deben incluir también el cambio dinámico del atributo *aria-expanded="false"* a *aria-expanded="true"*, para que el cambio de estado sea anunciado correctamente al usuario. Por el contrario, el valor del atributo *role* debe permanecer fijo.

²⁵ Puedes ampliar información y consultar ejemplos en el artículo “Live Regions y WAI-ARIA”, <https://olgacarreras.blogspot.com.es/2013/11/live-regions-y-wai-aria-como-mejorar-la.html>

Puedes acceder a la definición de cada estado o propiedad, y de su detalle, en el apartado “*Definitions of States and Properties (all aria-* attributes)*” de la especificación WAI-ARIA 1.1.

http://www.w3.org/TR/wai-aria-1.1/#state_prop_def

Hay tres propiedades fundamentales, a las que hacen referencia muchas de las técnicas ARIA de las WCAG 2.1, que debemos utilizar continuamente y hacerlo bien. Son aquellas que permiten etiquetar o asociar descripciones a los elementos:

- *aria-label*,
- *aria-labelledby* y
- *aria-describedby*.

Las veremos en profundidad en las próximas páginas.

aria-label

Con este atributo indicamos directamente **el contenido de la etiqueta del elemento**:

```
<button aria-label="Cerrar">X</button>
```

Es muy importante saber que *aria-label* anula la etiqueta nativa. Es decir, el lector de pantalla no leerá “botón X” ni “botón X Cerrar” sino “botón Cerrar”, que en este caso es lo queremos.

También podemos usarlo para **ampliar la información** de un enlace que en principio podría ser confuso:

```
<a href="" aria-label="Leer más sobre accesibilidad web">Leer más</a>
```

Otro uso habitual es para **distinguir zonas de la página** (*landmark roles*) del mismo tipo, por ejemplo, si tenemos dos zonas de navegación:

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

Los tres ejemplos expuestos son algunos de los casos para los que las WCAG 2.1 admiten el uso de *aria-label* como técnica suficiente.

Sin embargo, no debemos utilizar este atributo en un enlace a modo de *title*:

Forma incorrecta

```
<a href="" aria-label="Se abre en ventana nueva" target="_blank">  
Leer más sobre accesibilidad web</a>
```

aria-labelledby

Tanto *aria-label* como *aria-labelledby* se usan para indicar **el contenido de la etiqueta del elemento**. Sin embargo, si el nombre de la etiqueta está visible en la página, es mejor usar *aria-labelledby*, referenciando el id del elemento (o ids de elementos) que contiene dicha información.

Los atributos *aria-label* y *aria-labelledby* se anulan entre sí y no tiene sentido usarlos juntos. Además, al igual que *aria-label*, *aria-labelledby* también anula la etiqueta nativa.

En este ejemplo, el lector anunciará el enlace como “Informe anual 2017 Descargar PDF (25 KB)” en vez de anunciarlo como “enlace PDF (25 KB)”, es decir, concatenará la etiqueta de los dos elementos referenciados.

```
<h2 id="informe">
    Informe anual 2017
</h2>
<p><a aria-labelledby="informe pdf" href="" id="pdf">
    Descargar PDF (25 KB)
</a></p>
```

Del mismo modo podemos usar el atributo *alt* de una imagen para conseguir el mismo efecto:

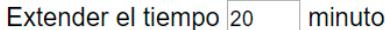
```
<h2 id="informe">
    Informe anual 2017
</h2>
<p><a aria-labelledby="informe pdf" href="" id="pdf">
    
    (25 KB)
</a></p>
```

Otro ejemplo de uso habitual se da en los **formularios de búsqueda**. En el siguiente caso, el lector anunciará el campo como “Buscar” aprovechando una etiqueta ya disponible en la página, esto es, la del botón adyacente.

A visual representation of a search form. It consists of a rectangular input field on the left and a rectangular button labeled "Buscar" on the right.

```
<input name="searchtxt" type="text" aria-labelledby="btn">
<input name="searchbtn" id="btn" type="submit" value="Buscar">
```

También podemos usar *aria-labelledby* con el **valor de los campos de un formulario**. En este ejemplo el lector anunciará el campo como “Extender el tiempo a 20 minutos” (en vez de “Extender el tiempo a 20”).



A visual representation of a form element. It shows the text "Extender el tiempo" followed by a text input field containing the number "20" and the word "minutos" to its right.

```
<label for="duration" id="timeout">
    Extender el tiempo a
</label>
<input type="text" id="duration" value="20" aria-labelledby="timeout
duration unit">
<span id="unit">
    minutos
</span>
```

Además, podemos usar *aria-labelledby* para **etiquetar zonas de la página**, o también para etiquetar **imágenes** como en el siguiente ejemplo:



```
<div role="img" aria-labelledby="puntos">
  
  
  
  
  
</div>
<div id="puntos" aria-hidden="true">
  <span class="oculto">puntuación </span>
  <span>4 de 5</span>
</div>
```

En este caso el lector ignorará las estrellas (porque tienen *alt=""*) y anunciará la capa como una imagen con la etiqueta asociada “puntuación 4 de 5”. Se utiliza el atributo *aria-hidden* para que el producto de apoyo no anuncie después el contenido del *div*, pues sería innecesario y redundante. Además, se oculta visualmente el texto “puntuación”, en este caso da igual con qué técnica, porque no será anunciado en ningún caso por el lector de pantalla.

Este ejemplo nos permite además indicar que, aunque los elementos a los que referencia el atributo *aria-labelledby* estén ocultos con cualquier técnica, no afecta a su comportamiento, seguirá anunciando “puntuación 4 de 5”.

Estos son varios de los ejemplos que incluyen las WCAG 2.1 para el uso de *aria-labelledby* como técnica suficiente.

aria-describedby

Permite referenciar el id del elemento que queremos que funcione como descripción de otro, es decir, que proporcione una **información adicional** a la de su etiqueta.

```
<button aria-label="Cerrar" aria-describedby="descClose">  
  X  
</button>  
  
<div id="descClose">  
  Cerrar esta ventana descartará cualquier cambio realizado y regresará  
  a la página principal.  
</div>
```

Imaginemos una ventana con un botón cerrar, cierto contenido y al final de la ventana un mensaje de aviso. Asociándolo de esta manera, el lector anunciará: “Botón Cerrar. Cerrar esta ventana descartará cualquier cambio realizado y regresará a la página principal.”, es decir, como un añadido a la etiqueta, evitando que la información le pase desapercibida al usuario de lector de pantalla.

También puede usarse *aria-describedby* para la **descripción de imágenes**:

```
<img src="" alt="Esquema WCAG 2.1" aria-describedby="descWCAG" />  
  
<div id="descWCAG">  
  <p>Las pautas WCAG 2.1 se componen de 4 principios:</p>  
  <ol>  
    <li>Perceptible</li>  
    <li>Operable</li>  
    <li>Comprensible</li>  
    <li>Robusto</li>  
  </ol>  
</div>
```

Otra forma de usar *aria-describedby* es para **asociar un campo de texto a su ayuda contextual**:

Contraseña (obligatorio):

La contraseña debe tener mínimo 6 caracteres

```
<label for="contra">  
  Contraseña (obligatorio) :  
</label>  
<input name="contra" id="contra" type="password" aria-  
describedby="descripcionContra" aria-required="true" />  
  
<p id="descripcionContra" class="ayuda">  
  La contraseña debe tener mínimo 6 caracteres  
</p>
```

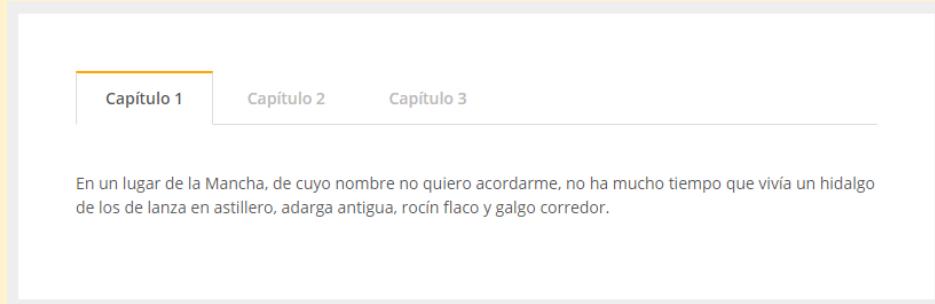
El lector anuncia el campo con la etiqueta “Contraseña (obligatorio)” y le añade la descripción “La contraseña debe tener mínimo 6 caracteres.”²⁶, de esta manera, los requisitos del campo no le pasarán desapercibidos al usuario de lector de pantalla cuando vaya saltando de campo en campo por el formulario.

Estos son varios de los ejemplos que incluyen las WCAG 2.1 para el uso de *aria-describedby* como técnica suficiente.

²⁶ NVDA anuncia exactamente ese campo como: “Contraseña obligatorio. Campo de edición contraseña. Requerido. La contraseña debe tener mínimo 6 caracteres. En blanco”. Según la combinación de lector y navegador, el anuncio puede ser diferente.

Ejemplo 2. Mejorar el uso accesible: navegación en pestañas

Supongamos que tenemos este diseño, definido con el código que se muestra a continuación:



Captura de pantalla 14. Contenido organizado en tres pestañas

```
<!-- pestanas -->
<ul>
<li><a href="#cap1">Capítulo 1</a></li>
<li><a href="#cap2">Capítulo 2</a></li>
<li><a href="#cap3">Capítulo 3</a></li>
</ul>

<!-- contenidos -->
<section id="cap1">
    En un lugar de la Mancha...
</section>
<section id="cap2">
    ...
</section>
<section id="cap3">
    ...
</section>
```

Cuando se pulsa en las pestañas, mediante *JavaScript* no intrusivo y CSS, se muestra el contenido del capítulo correspondiente. Sin embargo, los productos de apoyo no entenderán la verdadera función de la lista ni su relación con las secciones. Vamos a mejorar su accesibilidad con ARIA.

- En primer lugar, debemos indicar que la lista se va a comportar como una lista de pestañas (*role=tablist*) y darle una etiqueta con *aria-label* o *aria-labelledby*.

- Luego, debe indicarse que cada elemento de la lista cumple el rol de presentación de contenidos (*role=presentation*) para que se anuncie su contenido pero no su rol nativo²⁷.
- A los enlaces se les tienen que añadir varios atributos:
 - Primero, el rol de pestaña (*role=tab*).
 - A continuación, qué contenido (sección) controla (*aria-controls=cap1*).
 - Si está seleccionado o no (*aria-selected=true*) que deberá cambiarse dinámicamente por JavaScript cada vez que el usuario pulse una pestaña.
 - Por último, *aria-posinset* y *aria-setsize* para indicar qué posición ocupa y cuántas pestañas hay.
- Por último, a cada sección se le deben añadir los atributos:
 - Primero cuál es su rol (*role=tabpanel*)
 - A continuación, haremos que pueda coger el foco con *tabindex=0*, para facilitar a los usuarios de productos de apoyo moverse fácilmente hasta ella, especialmente si dentro del panel no hay un elemento que pueda coger el foco.
 - Si el método utilizado para ocultar las pestañas solo las oculta visualmente (por ejemplo con un *text-indent:-1000px*) necesitaremos el atributo *aria-hidden=true* que indica que se oculte también al lector de pantalla, y deberá cambiarse también dinámicamente su valor por JavaScript a medida que el usuario pulse las pestañas.
 - Por último, tendremos que añadirlle el atributo *aria-labelledby* para etiquetarla con el nombre de su pestaña.

De este modo, el código final mejorado quedaría de la siguiente manera:

```
<!-- pestañas -->
<ul role="tablist" aria-label="Capítulos del Quijote">

  <li role="presentation"><a href="#cap1" tabindex="0" role="tab" aria-controls="cap1" aria-selected="true" aria-posinset="1" aria-setsize="3"> Capítulo 1</a></li>

  <li role="presentation"><a href="#cap2" tabindex="-1" role="tab" aria-controls="cap2" aria-posinset="2" aria-setsize="3"> Capítulo 2</a></li>

  <li role="presentation"><a href="#cap3" tabindex="-1" role="tab" aria-controls="cap3" aria-posinset="3" aria-setsize="3">Capítulo 3</a></li>

</ul>
```

²⁷ Cuando *role=presentation* se aplica a una imagen, es como aplicar un *alt=""* o *aria-hidden*, el contenido es ignorado por el lector de pantalla. Pero cuando se aplica a otro elemento, el lector anuncia su contenido textual pero eliminado la información semántica de su rol, es decir, sin anunciar que es una lista, una tabla o un encabezado.

```

<!-- contenidos -->
<section id="cap1" role="tabpanel" tabindex="0" aria-
labelledby="cap1">
En un lugar de la Mancha...
</section>

<section id="cap2" role="tabpanel" aria-hidden="true"
tabindex="0" aria-labelledby="cap2">
...
</section>

<section id="cap3" role="tabpanel" aria-hidden="true"
tabindex="0" aria-labelledby="cap3">
...
</section>

```

¿Por qué se añade `tabindex=0` y `tabindex=-1` a los enlaces de las pestañas si son elementos que ya de por sí cogen el foco?

La razón es por la manera en que deben comportarse las pestañas en el acceso mediante teclado. Si en cada sitio web se comportaran de una manera diferente, los usuarios que acceden solo con el teclado tendrían que aprender su uso en cada página.

En el caso de las pestañas, una vez que coge el foco la primera, la manera de moverse a las demás pestañas debe ser con la flecha derecha e izquierda del teclado; con la tecla Inicio y Fin debes poder ir a la primera y última pestaña; al pulsar la tecla Enter o Espacio en una pestaña tienes que activarla, etc.

Por eso a la primera pestaña se le da `tabindex=0`, para que coja el foco, y a las demás se las elimina de la secuencia del tabulador con `tabindex=-1` (atributo que indica que el elemento solo puede coger el foco por programación). A medida que el usuario selecciona las pestañas con las flechas, se va cambiando dinámicamente por JavaScript el atributo de `tabindex=-1` a `tabindex=0`. Es lo que se llama el *roving tabindex*²⁸.

Una vez mejorado el código, el lector anunciará que hay tres pestañas, y que la seleccionada es la primera²⁹. Además, el usuario podrá navegar de una a otra mediante el teclado teniendo perfecto conocimiento de dónde está y qué está pasando en la página en todo momento.

²⁸ Más información en https://www.w3.org/TR/wai-aria-practices-1.1/#kbd_roving_tabindex

²⁹ Cuando la primera pestaña del ejemplo coge el foco, NVDA anuncia exactamente: “Capítulos del Quijote, pestañas. Capítulo 1 pestaña seleccionada 1 de 3.” Según la combinación de lector y navegador, el anuncio puede ser diferente.

Buenas prácticas

Mal usado, ARIA es un peligro. ARIA tiene el poder de dotar de semántica a la interfaz, de describir casi cualquier componente de manera que los productos de apoyo los puedan interpretar... pero anulando la semántica original, lo cual, si se hace inadvertidamente o por una incorrecta aplicación del estándar, puede confundir más que ayudar.

Para mitigar este problema, el W3C recomienda:

1. **No utilices ARIA si no es necesario.** Usa siempre que puedas etiquetas estándar de HTML de manera estándar. Si puedes usar `<input type="checkbox">` o `<button>` úsalos en vez de `<div role="checkbox">` o `<div role="button">`. Recuerda que el rol ARIA anula el rol nativo.
2. **Un rol es una promesa.** Si indicas que un `<div>` es un botón (`role=button`) como hemos visto en el primer ejemplo de este capítulo, esto no hace que los navegadores proporcionen a ese elemento el estilo o el comportamiento de un botón, solo conseguimos que sea anunciado como un botón, pero será responsabilidad tuya que se comporte como tal.
3. **Usa los roles y las propiedades según la especificación.** Recuerda que el rol no debe cambiar dinámicamente, solo se cambian las propiedades y estados. Debes marcar la estructura del widget (`menubar`, `tablist`...) y las relaciones entre sus elementos (`aria-labelledby`, `aria-control`...).
4. **Evita los conflictos.** No añadas ARIA a etiquetas si pueden entrar en conflicto con su propia semántica. Por ejemplo, si redefinimos el comportamiento de un *radio button* como si fuera un *checkbox* (`<input type="radio" role="checkbox" />`), cada agente de usuario podría implementarlo de una forma diferente, el comportamiento sería caótico, y se perdería la robustez del código.
5. **Evita la redundancia.** No añadas ARIA a los controles nativos con el mismo valor, ya que es redundantes. Por ejemplo, estos ejemplos serían absurdos:
`<input type="checkbox" role="checkbox" />`
``.
6. **Cambia los estados y las propiedades en respuesta a los eventos.** Si indicamos que un árbol está abierto con `aria-expanded=true`, cuando el usuario lo cierre, debemos cambiar la propiedad por *JavaScript* y ponerla a `false`, para que el producto de apoyo pueda anunciar adecuadamente que ahora está cerrado.
7. **Accesible por teclado.** En HTML los elementos que pueden coger el foco por defecto son los elementos de formulario, los botones y enlaces. A cualquier otro elemento que queramos dotar de interacción (un div, un elemento de lista, etc.) deberemos incluirle el atributo `tabindex=0` (o `tabindex="-1"` si solo queremos que coja el foco por programación).
8. **Sincroniza la interfaz visual con la interfaz accesible.** Si un elemento cambia su estado de seleccionado a no seleccionado, cambia también su estilo:

```
.treeitem[role="treeitem"] [aria-selected="true"] {  
    color: #000;  
    background-color: #f2f2f2;  
}  
  
.treeitem[role="treeitem"] [aria-selected="false"] {  
    color: #f2f2f2;  
    background-color: #000;  
}
```

9. Aunque ya no es necesario que la página funcione sin JavaScript activo, sino que basta hacerlo accesible de forma nativa, programa utilizando **JavaScript no intrusivo** y sigue el principio de **mejora progresiva**.
10. Si aplicas el estándar ARIA **revísalo** siempre con algunas de las herramientas que te indicamos en el apartado “Cómo revisar ARIA”.

Puedes leer más sobre estas buenas prácticas en **WAI-ARIA Authoring Practices 1.1**.

Este documento también explica por cada widget cuál tiene que ser su comportamiento por teclado, cuál es la correcta aplicación de cada rol, ofrece uno o varios ejemplos de cada uno y podemos descargar su JS asociado, lo que ahorra mucho tiempo de programación.

<https://www.w3.org/TR/wai-aria-practices/>

Ejemplo 3. Validar un campo obligatorio

Imaginemos que validamos los campos de nuestro formulario cuando pierden el foco y, si detectamos un error de validación, mostramos una capa con un mensaje de alerta. Para que sea lo más accesible posible podemos utilizar ARIA de la siguiente manera:

```
<!-- Capa de error -->
<div id="error" aria-live="assertive" role="alert">

    <p>
        <svg role="img" aria-label="error:> [...] </svg>
        El campo Nombre es obligatorio.
    </p>
</div>

<!-- Campo que ha provocado el error-->
<label for="c1">Nombre*</label>
<input type="text" id="c1" name="c1" aria-invalid="true"
aria-required="true" value="">
```

La capa de error tiene el *rol="alert"* y el atributo *aria-live="assertive"*, lo que provocará que en cuanto se modifique el contenido de la capa el lector anuncie “Alerta” y lea su contenido (sin que el foco se mueva a la capa).

El mensaje del error va precedido de un ícono vectorial en formato SVG. Este elemento tiene los atributos *rol="img"* y *aria-label="error:"*. Esto hará que el ícono sea anunciado como una imagen con la etiqueta “error:”.

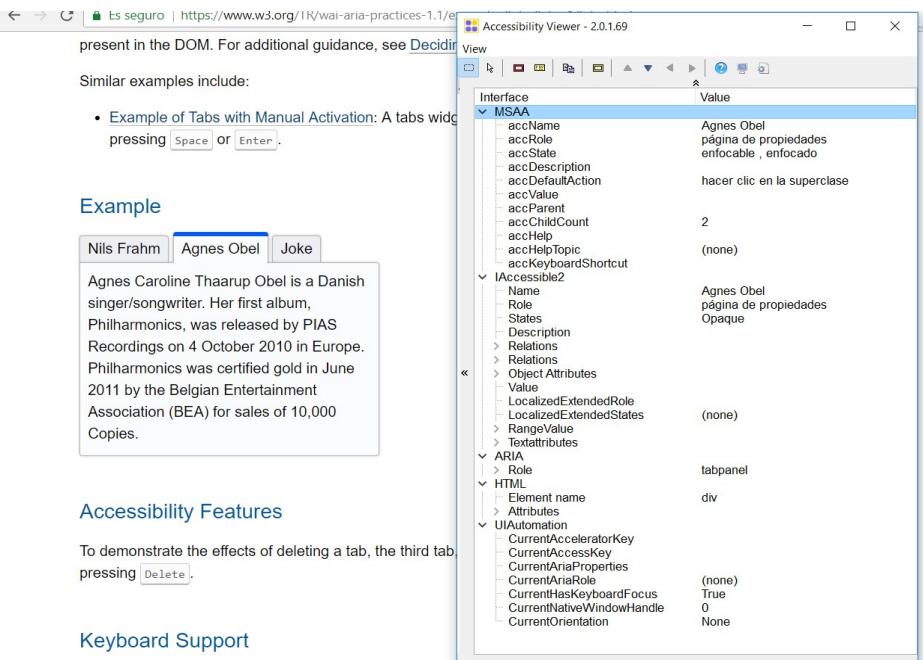
En cuanto al campo, vemos que tiene los atributos *aria-required="true"* y *aria-invalid="true"* para que, cuando coja el foco, el lector anuncie que el campo es obligatorio y que su entrada es inválida.

Cómo revisar ARIA

Existen diversas maneras de revisar tu código ARIA.

La manera más recomendable de revisar nuestra página es **acceder con un lector de pantalla** como NVDA o *VoiceOver*. Solo así podemos escuchar cómo se anuncia cada elemento, si es comprensible, si necesita más información, y las diferencias al incluir roles y atributos ARIA.

Otro tipo de herramienta muy útil es aquella que nos permite **conocer la información expuesta por el navegador a la API de accesibilidad**. Por ejemplo, tenemos la herramienta gratuita *Accessibility Viewer*³⁰, para Windows, que permite consultar cómo se expone la información a diferentes API de accesibilidad e inspeccionar específicamente los roles y atributos ARIA.



Captura de pantalla 15. *Accessibility Viewer*

Otra opción es **auditar** si estamos aplicando correctamente ARIA según la especificación. En la pestaña *Audits* de las “Herramientas para desarrolladores” del navegador Chrome, hay un apartado de “Accesibilidad”, y dentro de este apartado el epígrafe *ARIA Attributes Follow Best Practices* que valida aspectos formales específicos de ARIA: si estamos utilizando un atributo no permitido para un rol determinado, si el rol tiene sus atributos obligatorios, si los atributos tienen valores correctos, si el rol tiene los padres o hijos necesarios, etc.

³⁰ <https://developer.paciellogroup.com/resources/ariewner/>

The screenshot shows the Chrome DevTools interface with the 'Audits' tab selected. The title bar says 'www.w3.org 28/5/2018 0:44:11'. The main content area displays a report titled 'ARIA Attributes Follow Best Practices'. It includes a summary: 'These are opportunities to improve the usage of ARIA in your application which may enhance the experience for users of assistive technology, like a screen reader.' Below this is a list of 10 items, each with a green checkmark indicating they are good practices:

- ▶ [aria-*] attributes match their roles
- ▶ [role]s have all required [aria-*] attributes
- ▶ Elements with [role] that require specific children [role]s, are present
- ▶ [role]s are contained by their required parent element
- ▶ [role] values are valid
- ▶ [aria-*] attributes have valid values
- ▶ [aria-*] attributes are valid and not misspelled

Captura de pantalla 16. *ARIA Attributes Follow Best Practices* en Chrome

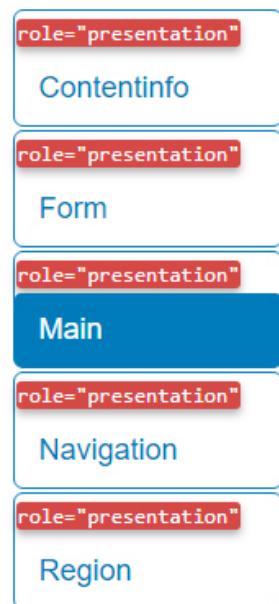
Por último, tenemos muchas extensiones de navegador que nos permiten **inspeccionar** los roles y atributos ARIA que se han aplicado en la página, como: tota11y³¹ y Visual ARIA³²; o la *Web Developer Toolbar* para los navegadores Chrome, Firefox y Opera³³. En las siguientes capturas de pantalla se muestran cómo presentan la información cada uno.



Captura de pantalla 17
Tota11y resaltando roles



Captura de pantalla 18
Visual ARIA resaltando roles,
estados y propiedades



Captura de pantalla 19
Web Developer Toolbar
resaltando roles

³¹ <http://khan.github.io/tota11y>

³² <http://whatsock.com/training/matrices/visual-aria.htm>

³³ <https://chrispederick.com/work/web-developer/>

Las técnicas específicas de ARIA

Las WCAG 2.1 incluyen 20 técnicas³⁴ específicas para la tecnología ARIA:

1. Usa la propiedad *aria-describedby* para **describir los controles** de interfaz de usuario.
2. Identifica **los campos obligatorios** con la propiedad *aria-required*.
4. Utiliza los roles para **informar del rol** de cada componente de interfaz de usuario.
5. Utiliza los **estados y propiedades** para informar del estado de cada componente de interfaz de usuario.
6. **Etiqueta** objetos con *aria-label*.
7. Define el **objetivo del enlace** con *aria-labelledby*.
8. Define el **objetivo del enlace** con *aria-label*.
9. Crea una etiqueta **concatenando** varios nodos de texto con *aria-labelledby*.
10. Usa *aria-labelledby* para dar un **texto alternativo** a contenido no textual.
11. Usa los roles *landmarks* para identificar las **zonas** de la página.
12. Identifica **encabezados** con *role=heading*.
13. Nombra **regiones** y *landmarks* con *aria-labelledby*.
14. Provee **etiquetas invisibles** con *aria-label* cuando no puedes usar etiquetas visibles.
15. Describe **imágenes** con *aria-describedby*.
16. Nombra **controles de interfaz** de usuario con *aria-labelledby*.
17. Usa roles de **agrupación** para identificar controles de formulario relacionados.
18. Identifica **errores** con *role=alertdialog*.
19. Identifica **errores** con *role=alert* o con *live regions*.
20. Identifica **regiones** de la página con *role=region*.
21. Identifica los **campos con errores** con *aria-invalid*



Puedes encontrar más información de las técnicas ARIA en
<https://www.w3.org/TR/WCAG20-TECHS/aria.html>

³⁴ La técnica 3 de ARIA ya no existe porque se integró dentro de la técnica 5 en 2014. Consistía en identificar un rango válido de información en un campo con las propiedades *valuemin* y *valuemax*.

Guías y enlaces de interés

"WAI-ARIA. Introducción, referencias, ejemplos, herramientas" de Olga Carreras. <http://olgacarreras.blogspot.com.es/2007/02/ajax-accesible-ii-wai-aria.html>

“Novedades WAI-ARIA 1.1” de Olga Carreras
<https://olgacarreras.blogspot.com.es/2017/12/novedades-wai-aria-11.html>

Recursos y herramientas

No existen herramientas que detecten todos los errores de accesibilidad de un sitio web, pero existen herramientas que ayudan a registrar los datos de la evaluación manual, que evalúan automáticamente determinados criterios, que simulan discapacidades o que ayudan a revisar de forma general el sitio web.



WCAG-EM Report Tool - Generador de Informes

Es una herramienta online del propio W3C que permite al auditor registrar los datos de la evaluación siguiendo la metodología WCAG-EM, y generar un informe automáticamente.

Aunque funciona online, se pueden guardar los datos en local y realizar la auditoría en diferentes momentos. Además, el informe generado se puede descargar en formato PDF y JSON.

De momento sólo está disponible en inglés para las WCAG 2.0, su adaptación a las WCAG 2.1 se espera para finales de 2018.

<https://www.w3.org/WAI/eval/report-tool/#/>

The screenshot shows the 'Step 1: Define the Evaluation Scope' page of the WCAG-EM Report Tool. At the top, there's a navigation bar with tabs: 'Start' (highlighted), 'Define Scope' (with a step indicator '1'), 'Explore Website' (with '2'), 'Select Sample' (with '3'), 'Audit Sample' (with '4'), 'Report Findings' (with '5'), and 'View Report'. Below the navigation is a section titled 'Step 1: Define the Evaluation Scope' with a sub-instruction: 'Define the overall parameters and scope of the evaluation. Ideally, this is done with the person who commissioned the evaluation (who may or may not be the website owner), to ensure common expectations about the scope of the evaluation. More guidance on this step is provided in [WCAG-EM Step 1: Define the Evaluation Scope](#)'. There are four input fields: 'Website name' (with placeholder 'http://'), 'Scope of the website' (with a large text area), 'Conformance target' (set to 'Level AA'), and 'Accessibility support baseline' (with a large text area). At the bottom, there are buttons for 'Previous step: Start' and 'Next step: Explore Website'.

Captura de pantalla 20. WCAG-EM Report Tool

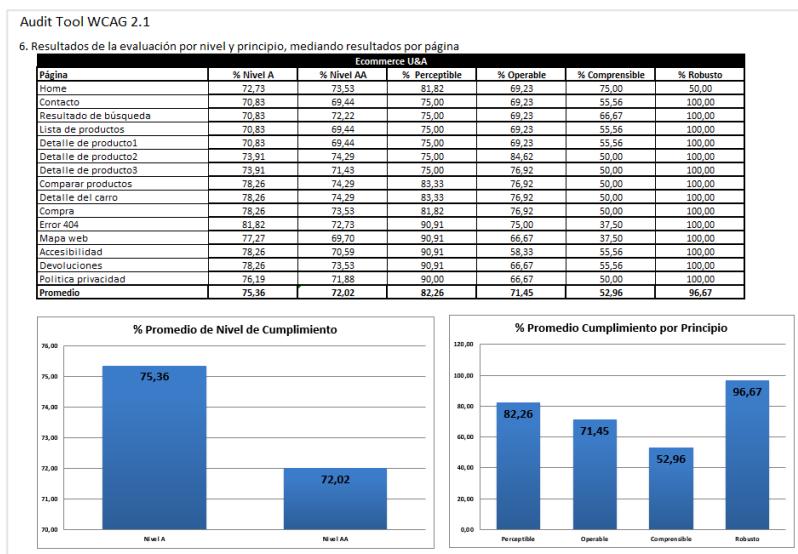
Audit Tool WCAG 2.1

Es una herramienta en formato excel que permite ir recogiendo los datos obtenidos durante la revisión automática y manual del sitio de acuerdo a las WCAG 2.1. Te genera gráficas y estadísticas de cumplimiento e incumplimiento por página, nivel, principio o criterio de conformidad. También incluye una comparativa del cumplimiento de la muestra respecto a las WCAG 2.0 frente a las WCAG 2.1

Esta información es útil para realizar el resumen ejecutivo, comparar sitios o resultados en el tiempo, identificar las páginas y criterios que más problemas presentan, o valorar el esfuerzo necesario para cumplir con la nueva versión del estándar.

Disponible en castellano y próximamente en inglés.

<https://olgacarreras.blogspot.com/2018/06/audit-tool-wcag-21.html>

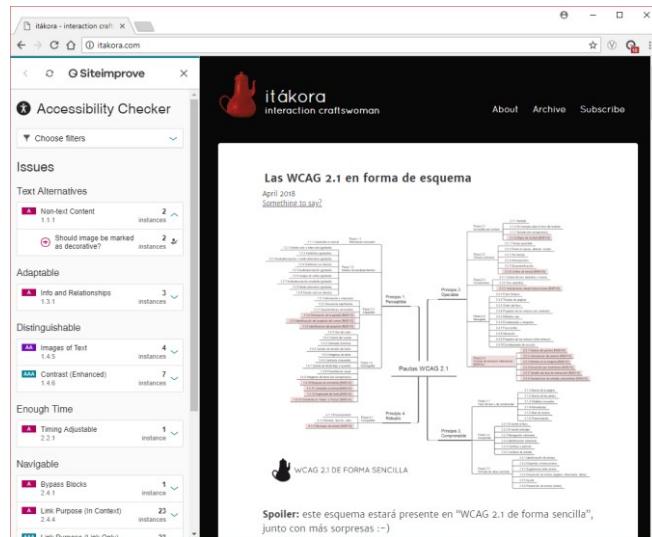


Captura de pantalla 21. Audit Tool WCAG 2.1

Validación automática de accesibilidad

La extensión **SiteImprove** del navegador Chrome es un validador automático y gratuito de accesibilidad que valida **algunos criterios (no todos)** de acuerdo, de momento, a las WCAG 2.0. Permite filtrar los resultados por nivel (A, AA, AAA), tipo (error, advertencia y revisión manual) y perfil de la persona que debe realizar los cambios (diseñador, *webmaster* y desarrollador). Es muy fiable en los resultados, sin falsos positivos. Solo está disponible en inglés.

<https://chrome.google.com/webstore/detail/siteimprove-accessibility/efcfolpjihicnikpmhnmphjhhpiellje>



Captura de pantalla 22. Extensión *SiteImprove*

SiteImprove también existe como aplicación online profesional de pago que permite programar evaluaciones periódicas de sitios completos (incluidos documentos PDF), integrarse con diversos gestores de contenidos, generar informes detallados, definir análisis personalizados, además de tener gestión de usuarios, histórico de acciones y otros tipos de análisis (ortografía, SEO, inventarios, etc.)

<https://siteimprove.com/en/content-accessibility/>

INFORME

WCAG 2.0 AA Detallado

SITIO
www.usableyaccesible.com

Visión general



Páginas con problemas

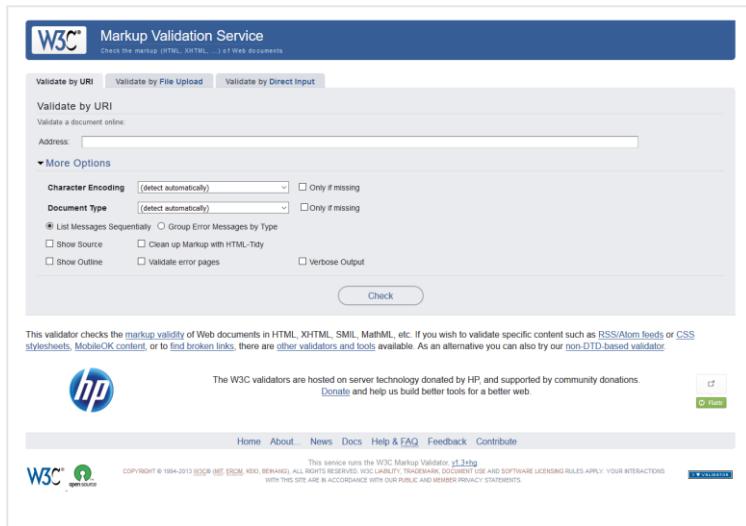
Los datos de esta tabla pueden distorsionar la problemática del sitio. El número de errores que se indica en cada página es el sumatorio de todas las instancias de todos los errores, advertencias y 'necesita revisión manual' de esa página, muchos de los cuales no serán realmente problemas.

Título de página	A	AA	AAA	Nivel de página
 Curriculum de Olga Carreras http://www.usableyaccesible.com/index.php	12	10	69	1
 Términos y condiciones de Usable y accesible http://www.usableyaccesible.com/aviso.php	10	0	7	2
 Acessibilidad: estilos de fuentes y opciones de personalización de Usable y accesible http://www.usableyaccesible.com/accesibilidad.php	11	10	15	2
 Mapa web de Usable y accesible http://www.usableyaccesible.com/mapa_web.php	16	8	28	2
 Enlaces de interés sobre usabilidad y accesibilidad web http://www.usableyaccesible.com/recurso_enlaces.php	16	8	124	2
 Validadores de accesibilidad y usabilidad web http://www.usableyaccesible.com/recurso_misValidadores.php	31	26	565	2
 Libros UX y accesibilidad recomendados y pon reseña de Olga Carreras http://www.usableyaccesible.com/recurso_libreria.php	48	26	63	2
 Glosario de usabilidad y accesibilidad web http://www.usableyaccesible.com/recurso_glosario.php	141	52	1.380	2
 Herramientas y material de apoyo para consultorías de accesibilidad o usabilidad http://www.usableyaccesible.com/recurso_descargas.php	14	51	98	2
 Consultoría y servicios de accesibilidad y usabilidad web http://www.usableyaccesible.com/servicios.php	10	8	8	2

Captura de pantalla 23. [Informe generado por la aplicación de pago SiteImprove](#) (PDF, 453KB)

Validación automática de código

Para cumplir con el criterio 4.1.1 (A) es necesario validar el código de nuestra página, para ello te recomendamos el **Servicio de Validación del W3C**, que permite validar el código HTML, XHTML, SMIL, MathML o SVG de tu web, <https://validator.w3.org/>



The W3C validators are hosted on server technology donated by HP and supported by community donations. [Donate](#) and help us build better tools for a better web.

This service runs the W3C Markup Validator v1.3.1b. Copyright © 1994-2013 W3C® (MIT, ERCIM, Keio). BEHIND THE SCENES. ALL RIGHTS RESERVED. W3C™, LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

Captura de pantalla 24. Servicio de validación de código del W3C

Si quieras comprobar el código de tus hojas de estilo CSS, el W3C tiene otro validador aparte, también muy completo.

<https://jigsaw.w3.org/css-validator/>



The W3C validators are developed with assistance from the Mozilla Foundation, and supported by community donations. [Donate](#) and help us build better tools for a better web.

Nota: Si deseas validar tu hoja de estilo CSS incrustada en un documento (X)HTML, deberías antes [comprobar que el DHTML utilizado es válido](#).

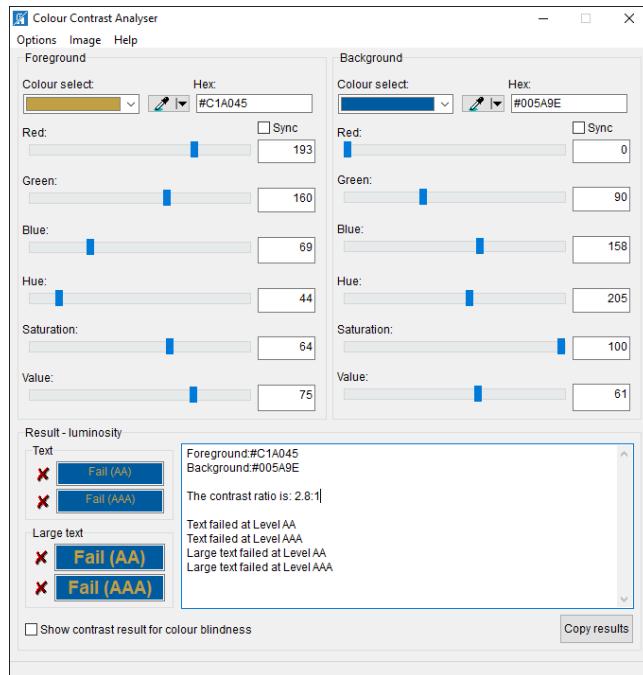
Copyright © 1994-2009 W3C® (MIT, ERCIM, Keio). ALL RIGHTS RESERVED. W3C™, LIABILITY, TRADEMARK, DOCUMENT USE AND SOFTWARE LICENSING RULES APPLY. YOUR INTERACTIONS WITH THIS SITE ARE IN ACCORDANCE WITH OUR PUBLIC AND MEMBER PRIVACY STATEMENTS.

Captura de pantalla 25. Servicio de validación de CSS del W3C

Validación de contraste de colores

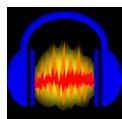
Para probar las ratios de contraste entre dos colores (criterios 1.4.3, 1.4.6 y 1.4.11), la herramienta más sencilla es **Colour Contrast Analyser** de *The Paciello Group*. Simplemente elige la combinación de colores y comprueba los resultados del contraste. Con la herramienta de cuentagotas puedes seleccionar colores de cualquier parte de la pantalla: páginas web, documentos PDF o Word, o cualquier programa en general.

<https://developer.paciellogroup.com/resources/contrastanalyser/>



Captura de pantalla 26. *Color Contrast Analyzer*

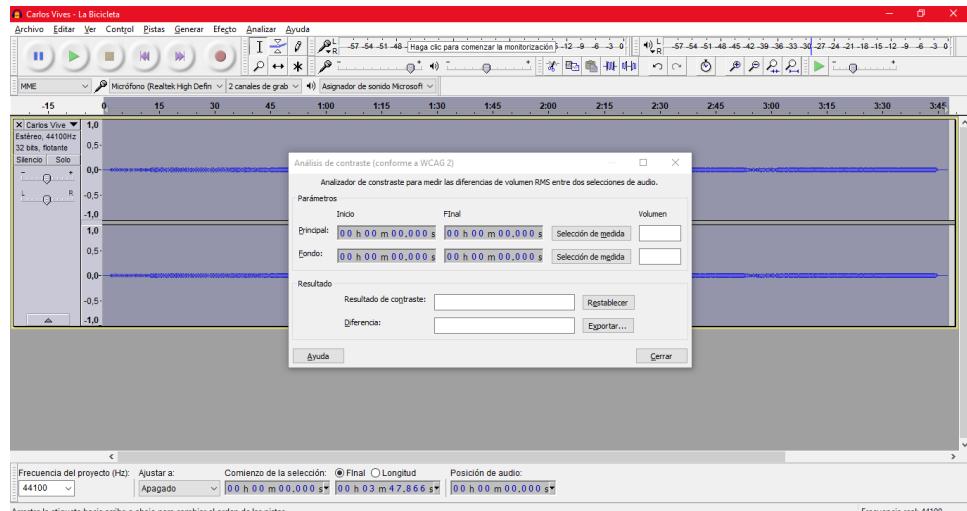
Validación del contraste de sonidos



Para comprobar si los sonidos de fondo y de primer plano se diferencian 20 decibelios se puede usar Audacity®, una herramienta de escritorio gratuita (disponible para Windows, Mac y Linux) que incluye una opción para verificar el contraste de acuerdo con el criterio 1.4.7 (AAA) de las WCAG 2.1

Dentro del menú “Analizar” tienes la opción “Contraste”. Selecciona las partes de la pista de audio que quieras verificar y obtendrás el resultado.

- Información sobre Audacity <https://www.audacityteam.org/>
- Descargar la herramienta: <https://sourceforge.net/projects/audacity/>
- Tutorial de cómo comprobar el contraste:
http://www.eramp.com/WCAG_2_audio_contrast_tool_help.htm

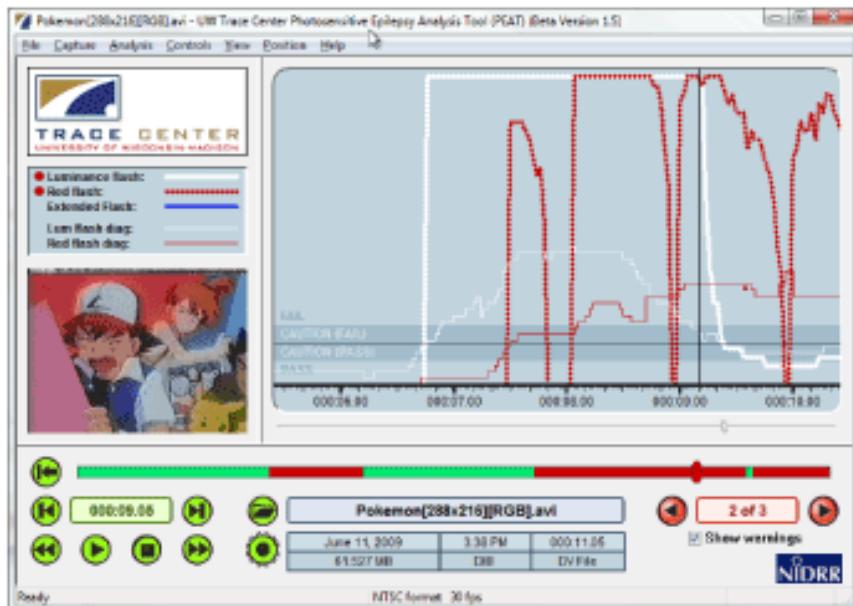


Captura de pantalla 27. Análisis de contraste con Audacity

Validación de destellos

De acuerdo a los criterios 2.3.1 (A) y 2.3.2 (AAA), para probar si un video contiene destellos que pueden causar convulsiones, recomendamos la herramienta **PEAT - Photosensitive Epilepsy Analysis Tool**, que muestra visualmente los momentos conflictivos.

<http://trace.wisc.edu/peat>

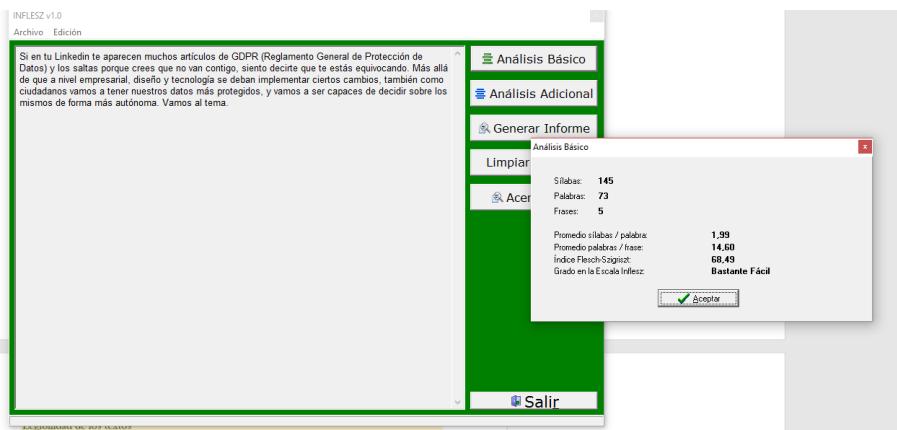


Captura de pantalla 28. PEAT

Validación de la legibilidad de los textos

El programa *freeware* INFLESZ para *Windows* calcula 9 parámetros que facilitan estimar la legibilidad de un texto escrito en español para evaluar el criterio 3.1.5 (AAA).

<https://legibilidad.blogspot.com.es/2015/01/el-programa-inflesz.html>



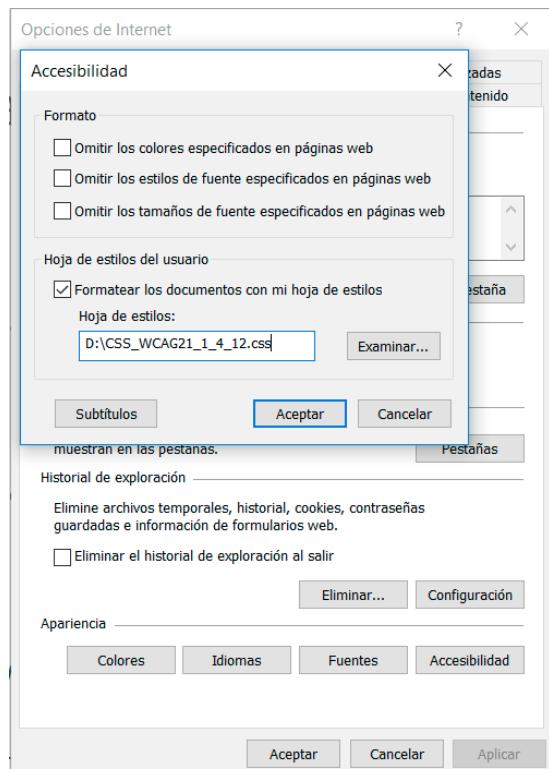
Captura de pantalla 29. Análisis con INFLESZ

Validación del espaciado de texto

El nuevo criterio "1.4.12 Espaciado del texto" (AA) pide verificar la página con determinados valores de espaciado para comprobar que no se produce pérdida de contenido ni de funcionalidad.

Para ello, Olga Carreras ha preparado una hoja de estilos propia que permite verificar si se cumple o no este criterio. Puedes descargarla y leer el tutorial en su blog

<https://olgacarreras.blogspot.com/2018/06/wcag-21-criterio-1412-aa-herramienta.html>



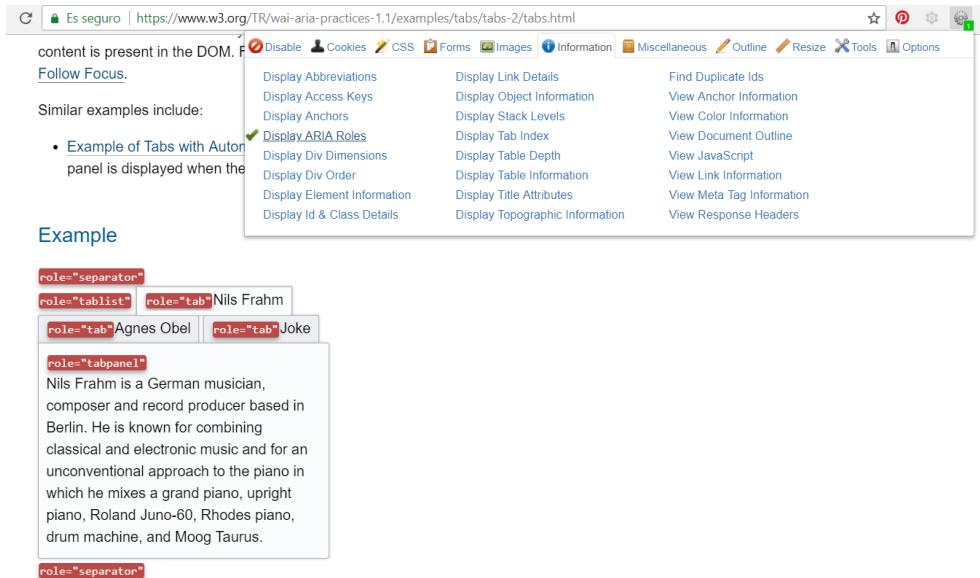
Captura de pantalla 30. Usar hoja de estilos propia

Barras de navegador

Existen diferentes *toolbars* o barras que se instalan dentro del navegador y ayudan a revisar manualmente muchos aspectos de accesibilidad, ya que permiten deshabilitar las imágenes o las CSS, resaltar los atributos de los elementos, marcar la estructura de la página, o mostrar los roles ARIA, entre otras opciones.

Para los navegadores Chrome, Firefox y Opera recomendamos la *Web Developer Toolbar*, y para Internet Explorer la *Web Accessibility Toolbar*, aunque su funcionamiento es limitado debido a que dejó de mantenerse hace un par de años.

- **Web Developer Toolbar** <https://chrispederick.com/work/web-developer/>
- **Web Accessibility Toolbar** <http://developer.paciellogroup.com/resources/wat/>

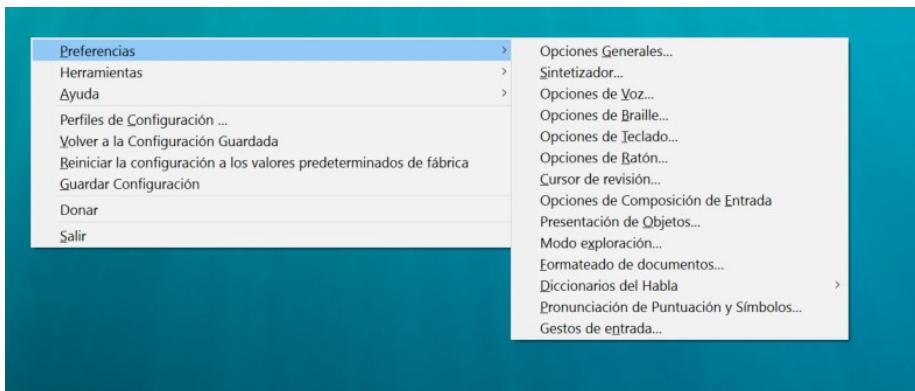


Captura de pantalla 31 *Web Developer Toolbar*

Lectores de pantalla

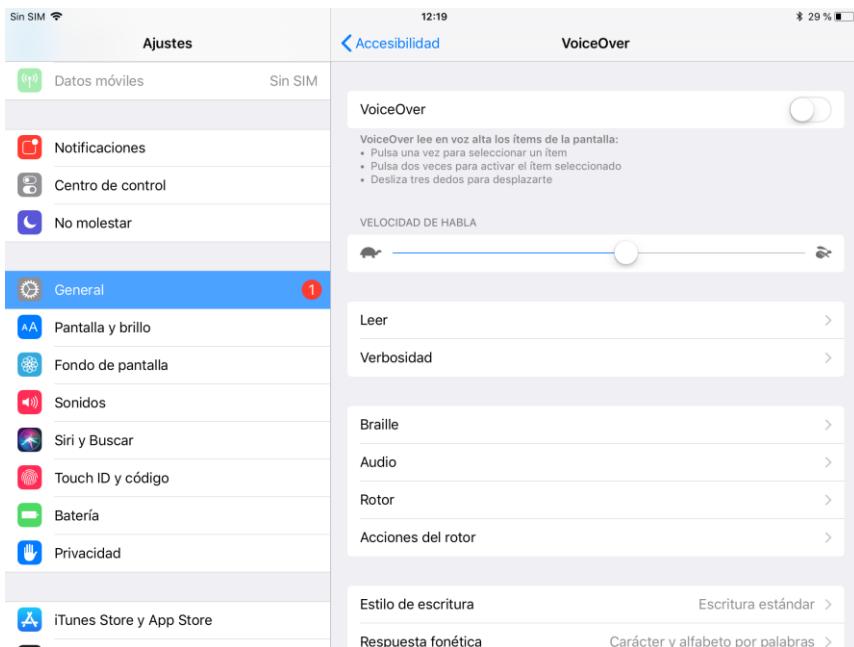
Existen muchos lectores de pantalla, pero nuestra recomendación para *Windows* es **NVDA**, que es gratuito y ofrece muchas opciones de configuración.

<https://www.nvaccess.org>



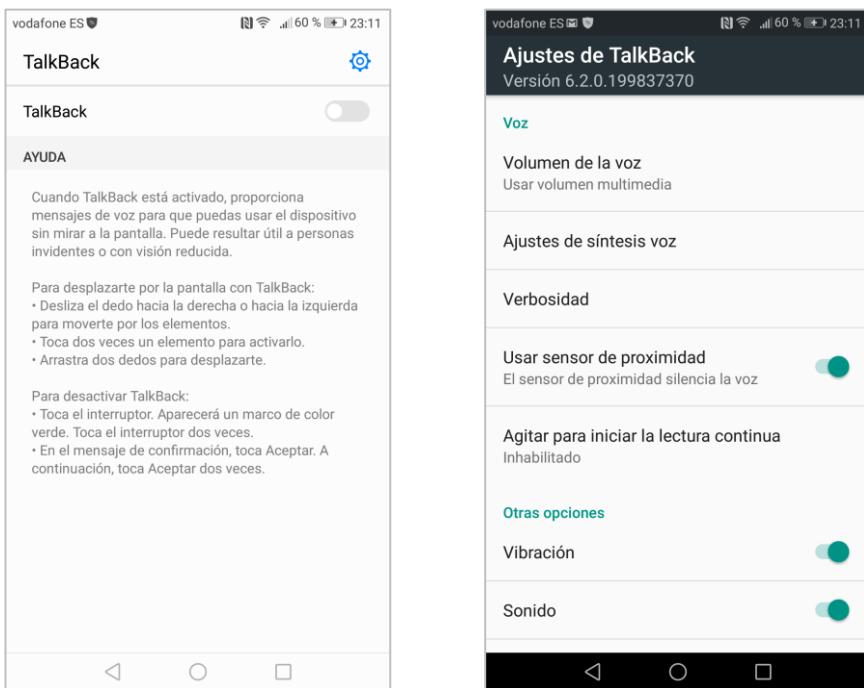
Captura de pantalla 32. Opciones de NVDA en desktop

Los usuarios de iOS disponen del lector de pantalla **VoiceOver** integrado en el sistema.



Captura de pantalla 33. VoiceOver en iOS para Tablet

En Android se puede activar **TalkBack** dentro de Ajustes.



Captura de pantalla 34. Funcionalidad de Talkbak y sus ajustes en un móvil Android

Los lectores de pantalla de escritorio se manejan con atajos de teclado, y los lectores de pantalla de dispositivos móviles mediante gestos táctiles. Las opciones son parecidas entre los diferentes lectores de pantalla, pero cada uno tiene sus peculiaridades y no siempre enuncian los contenidos de la misma manera. Por ello, es importante ajustarse lo máximo posible a los estándares y a la norma, para que puedan interpretar las páginas y aplicaciones con la mayor fidelidad posible.

Reproductores de video accesibles

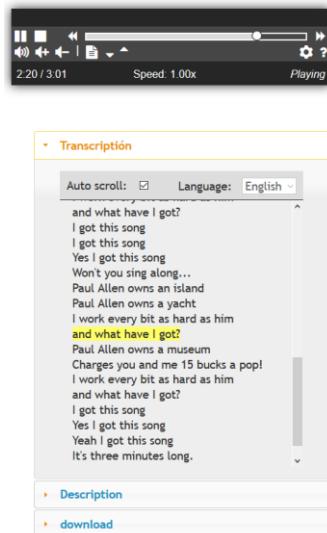
HTML5 permite incluir audio y vídeo de forma nativa con los elementos `<audio>` o `<video>`. Aunque están bien soportados por todos los navegadores actuales, no pasa lo mismo con su elemento `<track>`, que debería permitir asociarles subtítulos, capítulos o transcripciones. Por otra parte, no todos los reproductores nativos son completamente accesibles con el teclado y con los productos de apoyo. Además, no admiten audiodescripción o sincronización en lengua de signos.

Por ello es muy recomendable utilizar otros reproductores HTML5, basados en los elementos `<audio>` o `<video>`, que sí son completamente accesibles y admiten todas las alternativas necesarias para cumplir con los criterios de la Pauta 1.2. Nuestras recomendaciones son Able Player y OzPlayer. Ambos admiten vídeos de YouTube, siendo más recomendable utilizar estos reproductores por su apoyo ampliado a la accesibilidad que el propio reproductor de YouTube.

- **Able Player** <http://sidar.org/ableplayer2/index.html>
- **OzPlayer** <https://www.accessibilityoz.com/ozplayer>



Captura de pantalla 35. Vídeo con transcripción en OZ Player



Captura de pantalla 36. Audio con transcripción en Able Player

Puedes consultar una tabla comparativa de diferentes reproductores y sus características en HTML5 Video Player Comparison³⁵

³⁵ <http://videosws.praegnanz.de/#sws>

Simulación de discapacidades

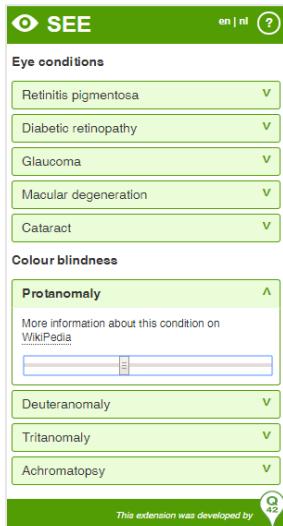
En el navegador Chrome podemos instalar extensiones que modifican las páginas web de tal manera que permiten ponernos en los zapatos de personas con discapacidad y comprobar cómo perciben, manejan y operan la web.

- **See:** glaucoma, cataratas, degeneración macular o daltonismo, entre otras.

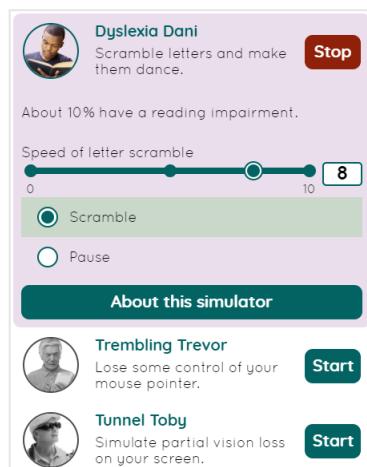
<https://chrome.google.com/webstore/detail/see/dkihccbkakkbzikjmpnhamkobjfdcn>

- **Funkify:** dislexia, temblores o dificultades cognitivas, entre otras.

<https://chrome.google.com/webstore/detail/funkify-disability-simula/ojcijidchelkddboickejbnbdpeajdijg>



Captura de pantalla 37. Extensión *See*



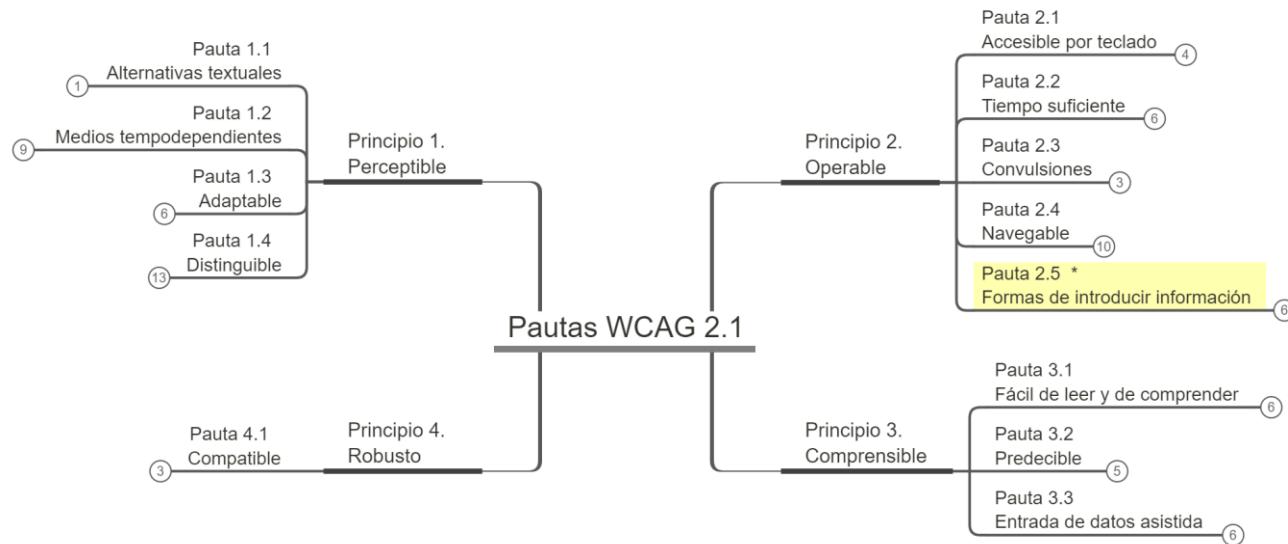
Captura de pantalla 38. Extensión *Funkify*

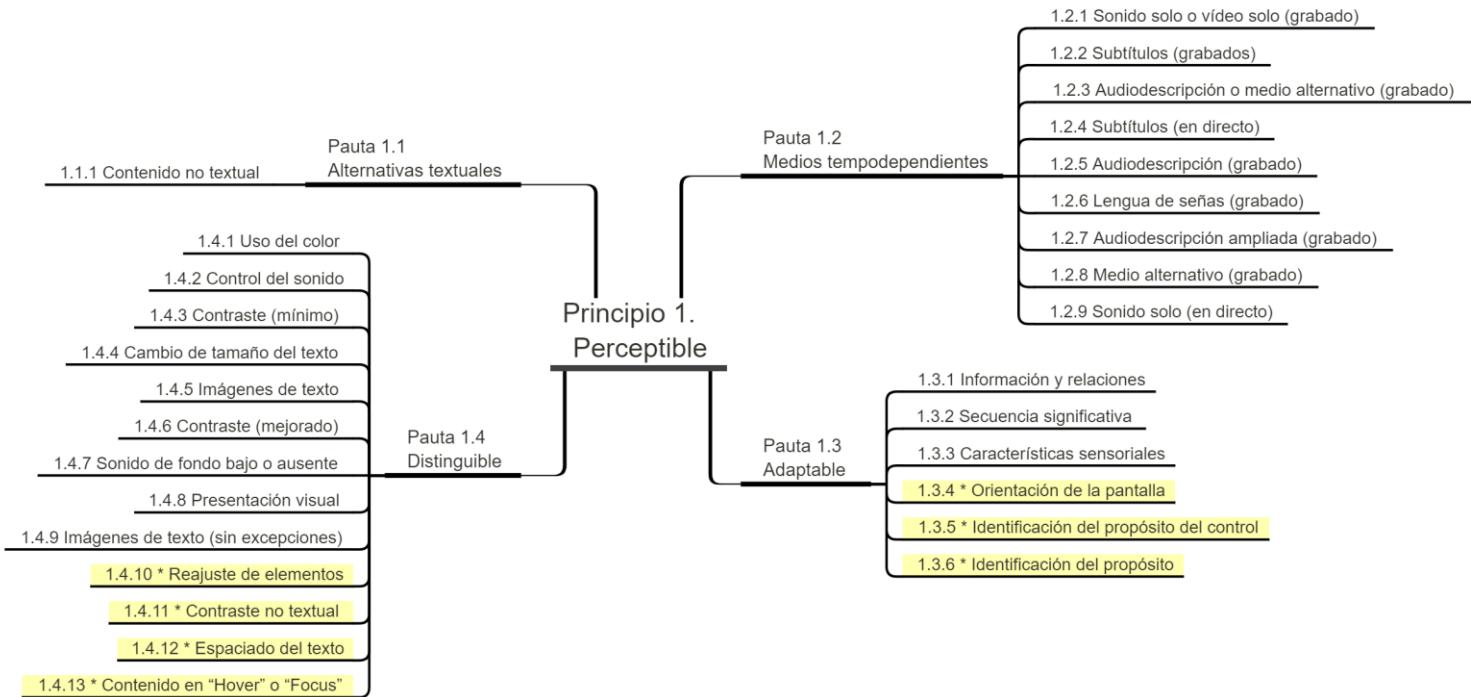
Resúmenes y esquemas

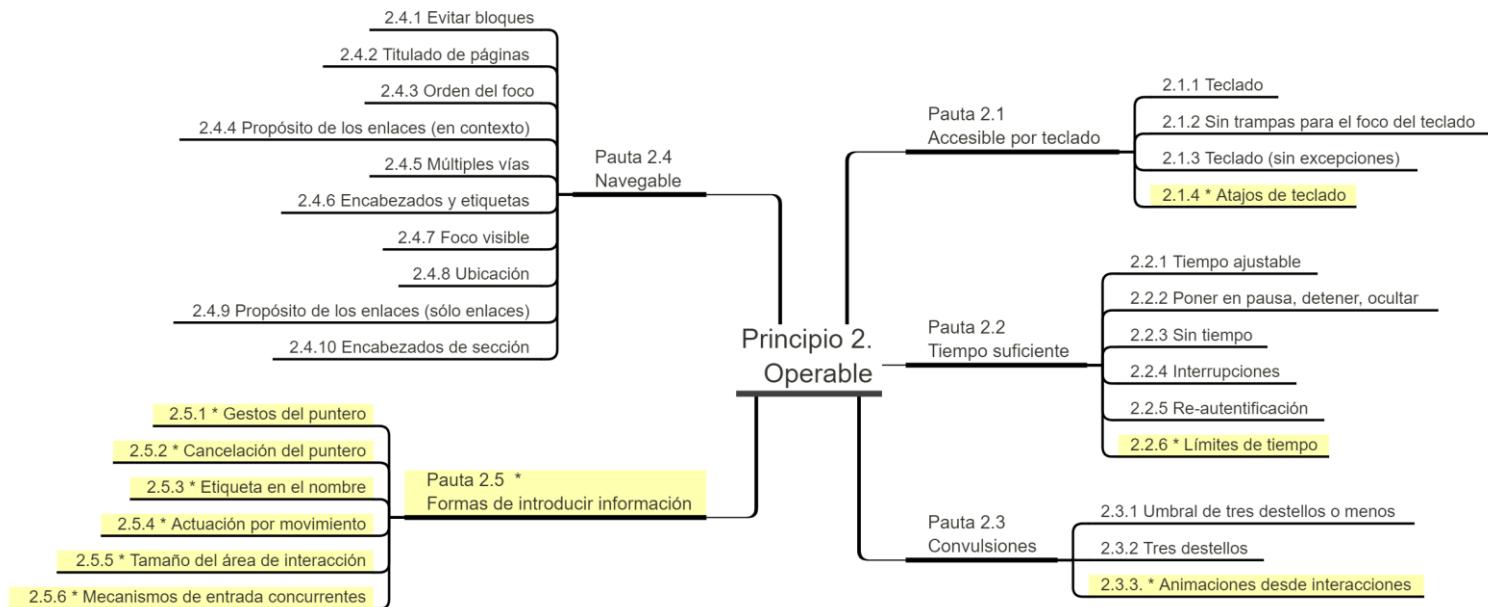


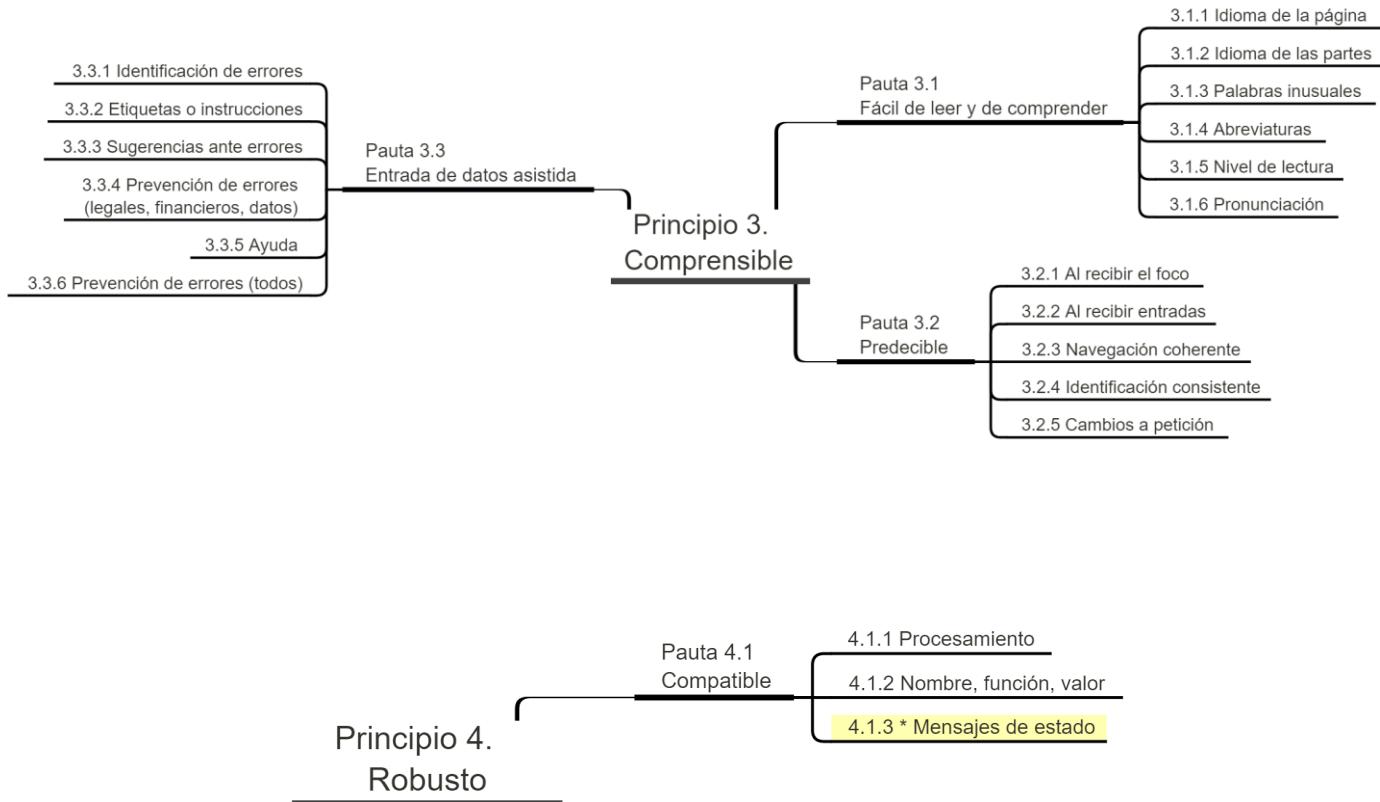
Diagramas conceptuales de las WCAG 2.1

* Nuevos 2.1.









Principios, Pautas y Criterios de Conformidad

* Nuevos 2.1

Principio 1. Perceptible

Pauta 1.1 Alternativas textuales

Criterio	Nivel	Descripción
1.1.1	A	Contenido no textual

Pauta 1.2 Medios tempodependientes

Criterio	Nivel	Descripción
1.2.1	A	Sonido solo o vídeo solo (grabado)
1.2.2	A	Sonido sincronizado con subtítulos (grabado)
1.2.3	A	Vídeo con audiodescripción o medio alternativo (grabado)
1.2.4	AA	Sonido sincronizado con subtítulos (en directo)
1.2.5	AA	Vídeo con audiodescripción (grabado)
1.2.6	AAA	Sonido sincronizado con lengua de signos (grabado)
1.2.7	AAA	Vídeo con audiodescripción ampliada (grabado)
1.2.8	AAA	Vídeo solo o medio sincronizado con un medio alternativo (grabados)
1.2.9	AAA	Sonido solo (en directo)

Pauta 1.3 Adaptable

Criterio	Nivel	Descripción
1.3.1	A	Información y relaciones
1.3.2	A	Secuencia significativa
1.3.3	A	Características sensoriales
1.3.4*	AA	Orientación de la pantalla
1.3.5*	AA	Identificación del propósito del campo
1.3.6*	AAA	Identificación del propósito

Pauta 1.4 Distinguible

Criterio	Nivel	Descripción
1.4.1	A	Uso del color
1.4.2	A	Control del sonido
1.4.3	AA	Contraste mínimo
1.4.4	AA	Cambio de tamaño del texto
1.4.5	AA	Imágenes de texto
1.4.6	AAA	Contraste mejorado
1.4.7	AAA	Sonido de fondo bajo o ausente
1.4.8	AAA	Presentación visual
1.4.9	AAA	Imágenes de texto (sin excepciones)
1.4.10*	AA	Reajuste de elementos
1.4.11*	AA	Contraste no textual
1.4.12*	AA	Espaciado del texto
1.4.13*	AA	Contenido en <i>hover</i> o <i>focus</i>

Principio 2. Operable

Pauta 2.1 Manejable por teclado

Criterio	Nivel	Descripción
2.1.1	A	Teclado
2.1.2	A	Sin trampas para el foco del teclado
2.1.3	AAA	Teclado (sin excepciones)
2.1.4*	A	Atajos de teclado

Pauta 2.2 Tiempo suficiente

Criterio	Nivel	Descripción
2.2.1	A	Tiempo ajustable
2.2.2	A	Poner en pausa, detener, ocultar
2.2.3	AAA	Sin tiempo
2.2.4	AAA	Interrupciones
2.2.5	AAA	Re-autentificación
2.2.6*	AAA	Límites de tiempo

Pauta 2.3 Reacciones físicas y psíquicas negativas

Criterio	Nivel	Descripción
2.3.1	A	Umbral de tres destellos o menos
2.3.2	AAA	Tres destellos
2.3.3*	AAA	Animaciones desde interacciones

Pauta 2.4 Navegable

Criterio	Nivel	Descripción
2.4.1	A	Evitar bloques
2.4.2	A	Titulado de páginas
2.4.3	A	Orden del foco
2.4.4	A	Propósito de los enlaces (en contexto)
2.4.5	AA	Múltiples vías
2.4.6	AA	Encabezados y etiquetas
2.4.7	AA	Foco visible
2.4.8	AAA	Ubicación
2.4.9	AAA	Propósito de los enlaces (sólo enlaces)
2.4.10	AAA	Encabezados de sección

Pauta 2.5 Introducción de la información *

Criterio	Nivel	Descripción
2.5.1*	A	Gestos del puntero
2.5.2*	A	Cancelación del puntero
2.5.3*	A	Etiqueta en el nombre
2.5.4*	A	Actuación por movimiento
2.5.5*	AAA	Tamaño del área de interacción
2.5.6*	AAA	Mecanismos de entrada concurrentes

Principio 3. Comprensible

Pauta 3.1 Fácil de leer

Criterio	Nivel	Descripción
3.1.1	A	Idioma de la página
3.1.2	AA	Idioma de las partes de la página
3.1.3	AAA	Palabras inusuales
3.1.4	AAA	Abreviaturas
3.1.5	AAA	Nivel de lectura
3.1.6	AAA	Pronunciación

Pauta 3.2 Predecible

Criterio	Nivel	Descripción
3.2.1	A	Al recibir el foco
3.2.2	A	Al recibir entradas
3.2.3	AA	Navegación coherente
3.2.4	AA	Identificación consistente
3.2.5	AAA	Cambios a petición

Pauta 3.3 Ayuda

Criterio	Nivel	Descripción
3.3.1	A	Identificación de errores
3.3.2	A	Etiquetas o instrucciones
3.3.3	AA	Sugerencias ante errores
3.3.4	AA	Prevención de errores en páginas legales, financieras y de datos
3.3.5	AAA	Ayuda
3.3.6	AAA	Prevención de errores en todo tipo de páginas

Principio 4. Robusto

Pauta 4.1 Compatible

Criterio	Nivel	Descripción
4.1.1	A	Procesamiento, código limpio
4.1.2	A	Nombre, función y valor
4.1.3*	AA	Mensajes de estado

Criterios de conformidad por niveles

* Nuevos 2.1

Nivel A

1.1.1	Contenido no textual
1.2.1	Sonido solo o vídeo solo (grabado)
1.2.2	Subtítulos (grabados)
1.2.3	Audiodescripción o medio alternativo (grabado)
1.3.1	Información y relaciones
1.3.2	Secuencia significativa
1.3.3	Características sensoriales
1.4.1	Uso de color
1.4.2	Control del sonido
2.1.1	Teclado
2.1.2	Sin trampas para el foco del teclado
2.1.4*	Atajos de teclado
2.2.1	Tiempo ajustable
2.2.2	Poner en pausa, detener, ocultar
2.3.1	Umbral de tres destellos o menos
2.4.1	Evitar bloques
2.4.2	Titulado de páginas
2.4.3	Orden del foco
2.4.4	Propósito de los enlaces (en contexto)
2.5.1*	Gestos del puntero
2.5.2*	Cancelación del puntero
2.5.3*	Etiqueta en el nombre
2.5.4*	Actuación por movimiento
3.1.1	Idioma de la página
3.2.1	Al recibir el foco
3.2.2	Al recibir entradas
3.3.1	Identificación de errores
3.3.2	Etiquetas o instrucciones
4.1.1	Procesamiento
4.1.2	Nombre, función, valor

Nivel AA

- | | |
|----------------|---|
| 1.2.4 | Subtítulos (en directo) |
| 1.2.5 | Audiodescripción (grabado) |
| 1.3.4* | Orientación de la pantalla |
| 1.3.5* | Identificación del propósito del campo |
| 1.4.3 | Contraste (mínimo) |
| 1.4.4 | Cambio de tamaño del texto |
| 1.4.5 | Imágenes de texto |
| 1.4.10* | Reajuste de elementos |
| 1.4.11* | Contraste no textual |
| 1.4.12* | Espaciado del texto |
| 1.4.13* | Contenido en <i>hover</i> o <i>focus</i> |
| 2.4.5 | Múltiples vías |
| 2.4.6 | Encabezados y etiquetas |
| 2.4.7 | Foco visible |
| 3.1.2 | Idioma de las partes |
| 3.2.3 | Navegación coherente |
| 3.2.4 | Identificación consistente |
| 3.3.3 | Sugerencias ante errores |
| 3.3.4 | Prevención de errores (legales, financieros, datos) |
| 4.1.3* | Mensajes de estado |

Nivel AAA

- 1.2.6** Lengua de señas (grabado)
- 1.2.7** Audiodescripción ampliada (grabada)
- 1.2.8** Medio alternativo (grabado)
- 1.2.9** Sólo sonido (en directo)
- 1.3.6*** Identificación del propósito
- 1.4.6** Contraste (mejorado)
- 1.4.7** Sonido de fondo bajo o ausente
- 1.4.8** Presentación visual
- 1.4.9** Imágenes de texto (sin excepciones)
- 2.1.3** Teclado (sin excepciones)
- 2.2.3** Sin tiempo
- 2.2.4** Interrupciones
- 2.2.5** Re-autentificación
- 2.2.6*** Límites de tiempo
- 2.3.2** Tres destellos
- 2.3.3*** Animaciones desde interacciones
- 2.4.8** Ubicación
- 2.4.9** Propósito de los enlaces (sólo enlaces)
- 2.4.10** Encabezados de sección
- 2.5.5*** Tamaño del área de interacción
- 2.5.6*** Mecanismos de entrada concurrentes
- 3.1.3** Palabras inusuales
- 3.1.4** Abreviaturas
- 3.1.5** Nivel de lectura
- 3.1.6** Pronunciación
- 3.2.5** Cambios a petición
- 3.3.5** Ayuda
- 3.3.6** Prevención de errores (todos)

Criterios para diseñadores

* Nuevos 2.1

Concepto	Criterio	Nivel	Descripción
Asegurar el contraste	1.4.3	AA	Contraste mínimo
	1.4.6	AAA	Contraste mejorado
	1.4.11*	AA	Contraste no textual
No usar sólo el color para informar	1.4.1	A	Uso del color
Elementos interactivos fáciles de identificar	2.4.7	AA	Foco visible
	3.2.4	AA	Identificación consistente
Navegación clara	2.4.5	AA	Múltiples vías
	2.4.8	AAA	Ubicación
	2.4.4	A	Propósito de los enlaces (en contexto)
	2.4.9	AAA	Propósito de los enlaces (sólo el enlace)
	3.2.3	AA	Navegación coherente
	3.3.5	AAA	Ayuda
Formularios claros	2.4.6	AA	Encabezados y etiquetas
	3.3.2	A	Etiquetas o instrucciones
	3.3.4	AA	Prevención de errores en páginas legales, financieras y de datos
	3.3.6	AAA	Prevención de errores en todo tipo de páginas
Feedback claro	3.3.1	A	Identificación de errores
	3.3.3	AA	Sugerencias ante errores
Organización del contenido clara	1.4.8	AAA	Presentación visual
	2.4.6	AA	Encabezados y etiquetas
	2.4.10	AAA	Encabezados de sección
Diseña para diferentes tamaños	1.3.4*	AA	Orientación de la pantalla
	1.4.4	AA	Cambio de tamaño del texto
	1.4.10*	AA	Reajuste de elementos
	1.4.12*	AA	Espaciado del texto
	2.5.5*	AAA	Tamaño del área de interacción
Imágenes, videos y audios	1.1.1	A	Contenido no textual
	1.4.2	A	Control del audio
	2.2.2	A	Poner en pausa, detener, ocultar

Concepto	Criterio	Nivel	Descripción
Manejo sencillo	1.4.13*	AA	Contenido en <i>hover</i> o <i>focus</i>
	2.5.1*	A	Gestos del puntero
	2.5.2*	A	Cancelación del puntero
	2.5.4*	A	Actuación por movimiento
	2.5.6*	AAA	Mecanismos de entrada concurrentes
Sin agobios de tiempo	2.2.1	A	Tiempo ajustable
	2.2.3	AAA	Sin tiempo
	2.2.6*	AAA	Límites de tiempo
No incomodar al usuario	2.2.4	AAA	Interrupciones
	2.2.5	AAA	Re-autentificación
No tomes decisiones por el usuario	3.2.1	A	Al recibir el foco
	3.2.2	A	Al recibir entradas
	3.2.5	AAA	Cambios a petición
Molestias físicas	2.3.1	A	Umbral de tres destellos o menos
	2.3.2	AAA	Tres destellos
	2.3.3*	AAA	Animaciones desde interacciones

Criterios para creadores de contenido

* Nuevos 2.1

Concepto	Criterio	Nivel	Descripción
Imágenes, vídeos y audios	1.1.1	A	Contenido no textual
	1.2.1	A	Audio solo o vídeo solo (grabado)
	1.2.2	A	Audio sincronizado con subtítulos (grabado)
	1.2.3	A	Vídeo con audiodescripción o medio alternativo (grabado)
	1.2.4	AA	Audio sincronizado con subtítulos (en directo)
	1.2.5	AA	Vídeo con audiodescripción (grabado)
	1.2.6	AAA	Audio sincronizado con lengua de signos (grabado)
	1.2.7	AAA	Vídeo con audiodescripción ampliada (grabado)
	1.2.8	AAA	Vídeo solo o medio sincronizado con un medio alternativo (grabados)
	1.2.9	AAA	Audio solo (en directo)
Redacción sencilla	1.4.2	A	Control del audio
	1.3.3.	A	Características sensoriales
	2.4.2	A	Titulado de páginas
	2.4.4	A	Propósito de los enlaces (en contexto)
	2.4.9	AAA	Propósito de los enlaces (sólo enlaces)
	2.4.6	AA	Encabezados y etiquetas
	2.4.10	AAA	Encabezados de sección
	3.2.4	AA	Identificación consistente
	3.3.2	A	Etiquetas o instrucciones
	3.3.5	AAA	Ayuda
Lenguaje sencillo	3.1.3	AAA	Palabras inusuales
	3.1.4	AAA	Abreviaturas
	3.1.5	AAA	Nivel de lectura
Formateo sencillo	1.3.1	A	Información y relaciones
	1.4.4	AA	Cambio de tamaño del texto
	1.4.5	AA	Imágenes de texto
	1.4.9	AAA	Imágenes de texto (sin excepciones)
	1.4.10*	AA	Reajuste de elementos
	2.4.1	A	Evitar bloques
Idioma	3.1.1	A	Idioma de la página
	3.1.2	AA	Idioma de las partes de la página
	3.1.6	AAA	Pronunciación

Concepto	Criterio	Nivel	Descripción
Contraste auditivo	1.4.7	AAA	Sonido de fondo bajo o ausente
Asegurar el contraste	1.4.3	AA	Contraste mínimo
	1.4.6	AAA	Contraste mejorado
	1.4.11*	AA	Contraste no textual
No usar sólo el color para informar	1.4.1	A	Uso del color
Molestias físicas	2.3.1	A	Umbral de tres destellos o menos
	2.3.2	AAA	Tres destellos
	2.3.3*	AAA	Animaciones desde interacciones



Si el contenido está en otros formatos, como Word o PDF por ejemplo, debes integrar también la revisión de los criterios de diseño.

Criterios para desarrolladores

* Nuevos 2.1

Concepto	Criterio	Nivel	Descripción
Código válido	4.1.1	A	Procesamiento, código limpio
Estructura la información de forma semántica	1.3.1	A	Información y relaciones
	1.3.2	A	Secuencia significativa
	1.4.13*	AA	Contenido en <i>hover</i> o <i>focus</i>
	3.3.2	A	Etiquetas o instrucciones
	4.1.3*	AA	Mensajes de estado
Identifica los controles y su propósito	1.3.5*	AA	Identificación del propósito del campo
	1.3.6*	AAA	Identificación del propósito
	4.1.2	A	Nombre, función y valor
	2.4.4	A	Propósito de los enlaces (en contexto)
	2.4.9	AAA	Propósito de los enlaces (sólo enlaces)
	2.5.3*	A	Etiqueta en el nombre
Separa el contenido de la presentación	1.1.1	A	Contenido no textual
	1.4.5	AA	Imágenes de texto
	1.4.9	AAA	Imágenes de texto (sin excepciones)
Uso mediante teclado	2.1.1	A	Teclado
	2.1.2	A	Sin trampas para el foco del teclado
	2.1.3	AAA	Teclado (sin excepciones)
	2.1.4*	A	Atajos de teclado
	2.4.3	A	Orden del foco
	2.4.7	AA	Foco visible
Facilitar la navegación	2.4.1	A	Evitar bloques
	2.4.2	A	Titulado de páginas
	2.4.8	AAA	Ubicación
Idioma	3.1.1	A	Idioma de la página
	3.1.2	AA	Idioma de las partes de la página
Ayuda a llenar formularios	3.3.1	A	Identificación de errores
	3.3.2	A	Etiquetas o instrucciones
	3.3.3	AA	Sugerencias ante errores
	3.2.4	AA	Identificación consistente
No incomodar al usuario	2.2.4	AAA	Interrupciones
	2.2.5	AAA	Re-autentificación
	2.2.1	A	Tiempo ajustable

Concepto	Criterio	Nivel	Descripción
Sin agobios de tiempo	2.2.3	AAA	Sin tiempo
	2.2.6*	AAA	Límites de tiempo
Maqueta para diferentes tamaños	1.3.4*	AA	Orientación de la pantalla
	1.4.4	AA	Cambio de tamaño del texto
	1.4.10*	AA	Reajuste de elementos
	1.4.12*	AA	Espaciado del texto

Glosario

AfA – Access For All. Es un marco diseñado para definir y describir recursos de accesibilidad. El estándar oficial de AENOR, ISO e IEC es el 24751.

<http://dublincore.org/accessibilitywiki/AccessForAll>

AENOR - Asociación Española de Normalización y Certificación. Es una entidad dedicada al desarrollo de la normalización y la certificación en todos los sectores industriales y de servicios. <http://www.aenor.es>

API - Application Programming Interface. Es una interfaz implementada por un software que le permite interactuar con otro software. <http://es.wikipedia.org/wiki/API>

API de accesibilidad - API que expone información sobre los objetos y eventos a los productos de apoyo. Algunas API de accesibilidad son MSAA, IAccessible2 o UI Automation https://www.usableyaccesible.com/recurso_glosario.php#api_accesibilidad

ARIA - Accessible Rich Internet Applications. Es una especificación técnica que proporciona formas de marcar el código para mejorar la accesibilidad e interoperabilidad de los contenidos y las aplicaciones. <http://www.w3.org/WAI/PF/aria/>

ASCII - American Standard Code for Information Interchange. Es un esquema de codificación de caracteres basado en el orden del alfabeto inglés.

<http://es.wikipedia.org/wiki/ASCII>

CAPTCHA - Completely Automated Public Turing test to tell Computers and Humans Apart. Es una prueba para determinar si quien está usando el sistema es una persona o una máquina. <http://es.wikipedia.org/wiki/CAPTCHA>

CEN – Comité Europeo de Normalización es una organización no lucrativa privada cuya misión es fomentar la economía europea, el bienestar de ciudadanos europeos y el medio ambiente. <https://www.cen.eu/>

CSS - Cascading Style Sheets. Es un mecanismo sencillo para añadir estilo a las páginas web. <http://es.wikipedia.org/wiki/CSS>

dB – Decibelio. Es una medida de la presión del sonido. <http://es.wikipedia.org/wiki/Decibelio>

Determinado o determinable por software: información suministrada por el autor de tal modo que las aplicaciones de usuario, incluyendo los productos de apoyo, pueden extraer y presentar esta información a los usuarios de distintas maneras. En un lenguaje de marcas se hace a partir de los elementos y atributos a los que acceden los productos de apoyo. En una tecnología que no es un lenguaje de marcas se hace a través de una API de accesibilidad que intercambia una estructura de datos con los productos de apoyo.

DOM - Document Object Model. Es una API que proporciona un conjunto estándar de objetos para representar documentos HTML y XML, un modelo estándar sobre cómo pueden combinarse dichos objetos, y una interfaz estándar para acceder a ellos y manipularlos. http://es.wikipedia.org/wiki/Document_Object_Model

DTD - Document Type Definition. Es un documento que incluye la estructura y sintaxis de un tipo de documento específico. <http://es.wikipedia.org/wiki/DTD>

Dublin Core. Es un modelo de metadatos que transmite un significado semántico a los elementos que define. <http://www.dublincore.org>

ETSI - European Telecommunications Standards Institute. Es una organización no lucrativa compuesta por más de 800 organizaciones mundiales que ayuda a los miembros a desarrollar, testar y ratificar estándares. <http://www.etsi.org/about>

Flash. Es una plataforma multimedia propietaria de Adobe usada para añadir animación, vídeo e interactividad a las páginas web. http://es.wikipedia.org/wiki/Adobe_Flash

Flexbox. Es una propiedad de CSS para adaptar el tamaño y posición de los elementos a diferentes tamaños de pantalla y diferentes dispositivos.

https://developer.mozilla.org/es/docs/Web/CSS/CSS_Flexible_Box_Layout/Usando_las_cajas_flexibles_CSS

HTML - HyperText Markup Language. Es el lenguaje de marcado predominante para páginas web. <https://www.w3.org/TR/html/>

HTTP - HyperText Transfer Protocol. Es un protocolo de red para sistemas de información distribuidos, colaborativos e hipermedia. <http://es.wikipedia.org/wiki/HTML>

IEC - International Electrotechnical Commission. Es una organización que publica estándares relacionados con la electricidad, electrónica y similares. <http://www.iec.ch>

Imagen rasterizada. Es una imagen compuesta por un conjunto de píxeles o puntos para ser mostrados en una pantalla o una impresora.

IMS - Global Learning Consortium. Es una organización internacional no lucrativa que promueve la tecnología en la educación. <http://www.imsglobal.org>

Interfaz de teclado. Es una interfaz usada por un programa para obtener pulsaciones de teclas incluso cuando la tecnología nativa no contiene un teclado como una pantalla táctil o aplicaciones de reconocimiento de voz. <http://www.w3.org/TR/2008/REC-WCAG20-20081211/#keybd-interfacedef>

ISO - International Organization for Standardization. Es una organización internacional que publica estándares para todo tipo de productos y servicios. <http://www.iso.org>

JavaScript. Es un lenguaje de programación del lado de cliente orientado a objetos. <http://es.wikipedia.org/wiki/Javascript>

Leet. Es un tipo de escritura compuesta de caracteres alfanuméricos. Por ejemplo, la palabra "HOLA" se escribe "H0L4". <http://es.wikipedia.org/wiki/Leet>

OCR - Optical character recognition. Es un proceso de reconocimiento óptico de caracteres en documentos, para transformarlos en texto. <http://es.wikipedia.org/wiki/OCR>

PDF - Portable Document Format. Es un estándar abierto para el intercambio de documentos con un formato determinado. <http://es.wikipedia.org/wiki/PDF>

Píxeles CSS - son la unidad de medida canónica para las mediciones de CSS. Esta unidad es independiente de la densidad de pantalla, y diferente de los píxeles físicos de las pantallas.

<https://www.w3.org/TR/css3-values/#lengths>

RDF - Resource Description Framework. Es un marco de metadatos estándar para representar la información en internet. <http://www.w3.org/TR/rdf11-primer/>

Sección 508. Es la norma de Estados Unidos para hacer la tecnología y la información accesible a personas con discapacidad. <http://www.section508.gov>

SIDAR - Seminario Iberoamericano sobre Discapacidad y Accesibilidad en la Red. Es un grupo de trabajo permanente y voluntario, integrado por personas expertas en nuevas tecnologías y en su accesibilidad. <http://www.sidar.org/>

Silverlight. Es una plataforma multimedia propietaria de Microsoft usada para añadir animación, vídeo e interactividad a las páginas web.

http://es.wikipedia.org/wiki/Microsoft_Silverlight

SMIL - Synchronized Multimedia Integration Language. Es un estándar del W3C para utilizar el lenguaje de marcado XML para describir presentaciones multimedia.

<http://es.wikipedia.org/wiki/SMIL>

SVG - Scalable Vector Graphics. Es un formato de gráficos vectoriales bidimensionales, tanto estáticos como animados, en formato XML

https://es.wikipedia.org/wiki/Gr%C3%A1ficos_vectoriales_escalables

UNE - Una Norma Española. Es un documento aprobado por AENOR de aplicación voluntaria que contiene especificaciones técnicas basadas en los resultados de la experiencia y del desarrollo tecnológico. http://www.aenor.es/aenor/normas/normas/quees_norma.asp

Unicode. Es un estándar de la industria de la computación para codificar, representar y manejar el texto de forma consistente en los diversos sistemas. <http://unicode.org>

W3C - World Wide Web Consortium. Es una comunidad internacional donde las organizaciones, personal y el público trabajan juntos para desarrollar estándares web.

<http://www.w3.org>

WAI - Web Accessibility Initiative. Es un grupo del W3C que desarrolla estrategias, pautas y recursos para ayudar a que internet sea accesible a personas con discapacidad.

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

WCAG - Web Content Accessibility Guidelines. Son una serie de recomendaciones del W3C para hacer el contenido web accesible a personas con discapacidad.

<http://www.w3.org/WAI/intro/wcag>

XHTML - eXtensible HyperText Markup Language. Es una familia del lenguaje de marcado XML que supera las versiones del HTML.. <http://es.wikipedia.org/wiki/XHTML>

XML - Extensible Markup Language. Es un lenguaje de formateo de texto muy simple pero muy flexible diseñado para intercambiar una amplia variedad de datos.

<http://es.wikipedia.org/wiki/XML>

Índice Global

Introducción a la Accesibilidad Web	11
Qué es la Accesibilidad Web	12
Profesionales relacionados con la Accesibilidad Web	15
Qué debes saber si creas contenido, diseñas o desarrollas front	18
Implantar y gestionar la política de Accesibilidad Web de una organización	24
Las pautas WCAG 2.1	27
Qué son las Pautas WCAG	28
Qué novedades introducen las WCAG 2.1 con respecto de las WCAG 2.0	29
Cómo se organizan las WCAG 2.1	30
Los cuatro principios y sus pautas	31
Las técnicas y los errores	34
¿A qué tecnologías se aplican las WCAG 2.1?	35
Por qué seguir las WCAG 2.1	36
Requisitos de Conformidad	37
Los 5 requisitos	38
Niveles de conformidad de las páginas web	39
Declaraciones de conformidad	40
Conformidades parciales	42
Ejemplos de declaración de conformidad	43
Plantilla de declaración de conformidad	44

Evaluar la accesibilidad con WCAG-EM.....	45
La metodología WCAG-EM.....	46
Paso 1. Definir el alcance de la evaluación	47
Paso 2. Explorar el sitio web	48
Paso 3. Seleccionar una muestra representativa de páginas	49
Paso 4. Auditar la muestra seleccionada.....	50
Paso 5. Registrar los resultados y elaborar un informe.....	51
Declarar la conformidad tras seguir la WCAG-EM.....	52
Guías y enlaces de interés	52
 Principio 1. Perceptible.....	53
Pauta 1.1 Alternativas textuales	54
Pauta 1.2 Medios tempodependientes	59
Pauta 1.3 Adaptable	72
Pauta 1.4 Distinguible.....	85
 Principio 2. Operable.....	104
Pauta 2.1 Accesible por teclado	105
Pauta 2.2 Tiempo suficiente	110
Pauta 2.3 Convulsiones	119
Pauta 2.4 Navegable.....	123
Pauta 2.5 [NUEVA] Formas de introducir información.....	135
 Principio 3. Comprensible.....	143
Pauta 3.1 Fácil de leer y de comprender.....	144
Pauta 3.2 Predecible	153
Pauta 3.3 Entrada de datos asistida	160
 Principio 4. Robusto	167
Pauta 4.1 Compatible.....	168

Documentos PDF accesibles.....	174
La accesibilidad de los documentos PDF	175
Las técnicas específicas de PDF	176
Implementar las técnicas PDF	178
Comprobar el documento de origen y el PDF.....	192
Exportar a PDF	194
Guías y enlaces de interés	196
ARIA, el aliado (casi) desconocido	197
Qué es ARIA.....	198
Roles	203
Estados y propiedades.....	205
Buenas prácticas	215
Cómo revisar ARIA	218
Las técnicas específicas de ARIA	220
Guías y enlaces de interés	221
Recursos y herramientas	222
WCAG-EM Report Tool - Generador de Informes	223
Audit Tool WCAG 2.1	224
Validación automática de accesibilidad	225
Validación automática de código.....	227
Validación de contraste de color	228
Validación del contraste de sonidos.....	229
Validación de destellos	230
Validación de la legibilidad de los textos.....	231
Validación del espaciado de texto	232
Barras de navegador.....	233
Lectores de pantalla	234
Reproductores de video accesibles.....	236
Simulación de discapacidades.....	237

Resúmenes y esquemas.....	238
Diagramas conceptuales de las WCAG 2.1	239
Principios, Pautas y Criterios de Conformidad	243
Criterios de conformidad por niveles	249
Criterios para diseñadores.....	252
Criterios para creadores de contenido	254
Criterios para desarrolladores.....	256
Glosario.....	258
Índice Global.....	261



Olga Revilla se considera una artesana de la interacción, ya que lleva 20 años diseñando la Arquitectura de Información y Experiencia de Usuario de todo tipo de proyectos y empresas.

Además, escribe en su blog *Itakora*, publica libros y artículos, y enseña Experiencia de Usuario en las principales universidades de diseño.

De educación multidisciplinar, es Licenciada en Periodismo, MBA y doctora en Ingeniería Industrial.



Olga Carreras cuenta con 20 años de experiencia en el desarrollo de proyectos web y está especializada en accesibilidad web y accesibilidad de documentos electrónicos, siendo en la actualidad consultora independiente.

Es la autora del blog *Usable y Accesible* y co-creadora del curso *Aprende Accesibilidad Web paso a paso*.

Imparte habitualmente formación en empresas, organismos públicos y universidades.

