



Accesibilidad Web

WCAG 2.2 de forma sencilla

OLGA REVILLA MUÑOZ
OLGA CARRERAS MONTOTO

Prólogo
Emmanuelle Gutiérrez y Restrepo

EDICIÓN 2024
REVISADA Y ACTUALIZADA

Accesibilidad Web

WCAG 2.2 de forma sencilla

Olga Revilla Muñoz
Olga Carreras Montoto

Prólogo de Emmanuelle Gutiérrez y Restrepo



Itákora Press

Accesibilidad Web. WCAG 2.2 de forma sencilla

Autoras: Olga Revilla Muñoz y Olga Carreras Montoto

Copyright © 2024 Itákora Press. Todos los derechos reservados.

Edición: Enero 2024

Este libro traduce, resume y cita contenidos de las *Web Content Accessibility Guidelines 2.2*. Copyright © [2023] *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

Accesibilidad del PDF: Olga Carreras Montoto

Imágenes:

- mosaico en portadas de Geralt en *Pixabay*;
- hombre en el metro de Andrea Piacquadio en *Pexels*;
- hombre conduciendo de Dan Gold en *Unsplash*.
- personas de *thispersondoesnotexist.com*

Iconos de “Idea” por Ralf Schmitzer; “Video” de Alfa Design; y “Stop” de Márcio Duarte; “Gestos de puntero” Jeff Portaro; “Sobre con flecha” de Andrey Vasiliev; todos ellos de *The Noun Project*. “Finger drag sliders” de *Freepik*

Logo de cafetera: Olga Revilla Muñoz

Revilla Muñoz, Olga; Carreras Montoto, Olga

Accesibilidad Web. WCAG 2.2 de forma sencilla. / Revilla Muñoz, Olga; Carreras Montoto, Olga —Madrid: Itákora Press, 2024 — 366 p.: 46 il.; 25.4 cm.

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 quien ayuda a hacer del mundo un lugar un poco mejor.

Olga Revilla

*Para Jesús, Silvia y Samuel,
al otro lado de "Usable y accesible"*

Olga Carreras

Gracias 2.2

Olga Revilla
Itákora

Los orígenes de este libro se sitúan en 2008, cuando se publicaron las pautas WCAG 2.0. En esa época los móviles empezaban a ganar cuota de mercado gracias en especial a los iPhone, lanzado sólo un año antes. Los móviles y las WCAG 2 supusieron un antes y un después para las personas con discapacidad, que ganaron dos grandes aliados en su acceso a la información y funcionalidades digitales.

Hoy el tema de conversación son las inteligencias artificiales: describen las imágenes, subtitulan los videos, resumen textos complicados en lenguaje fácil de comprender, traducen en tiempo real... De nuevo, una revolución para todas aquellas personas con limitaciones en el acceso a la información que pueden disfrutar de las ventajas de las tecnologías en su día a día. Las WCAG 2.2 ya empiezan a recoger esta revolución tecnológica al eliminar una de sus recomendaciones, ya que las tecnologías actuales sí saben cómo procesar el código, incluso el que no es válido. ¿Veremos desaparecer otras recomendaciones a medida que mejore la tecnología? Es bastante probable. Mientras tanto, el W3C nos irá pautando el camino a seguir para que las tecnologías sean cada vez más humanas.

En esta nueva edición hemos incorporado las novedades de las WCAG 2.2, revisado textos e imágenes para hacerlos más actuales y claros, reorganizado secciones, mejorado la legibilidad de los textos, actualizado el catálogo de herramientas y hemos incorporado tres temas de actualidad.

Por un lado, veremos cómo aplicar la accesibilidad dentro de los sistemas de diseño. Lo que antes era una creación artesanal se ha convertido en un proceso estandarizado con módulos validados. Esto aumenta la probabilidad de desarrollar productos digitales accesibles a menor coste y a escala.

Por otro lado, las aplicaciones que se ejecutan en una única URL representan un reto mayúsculo para las personas que dependen de los lectores de pantalla. Explicamos cómo hacer las *Single Page Applications* para que nadie se pierda en el camino.

Por último, un capítulo dedicado a cuidar las palabras que utilizamos. Que lleguen a la mayor parte de la población con respeto a sus diferencias está en nuestras manos si sabemos cómo.

Quedan temas en el tintero que nos hubiera gustado añadir, pero serán para próximas ediciones: aplicaciones móviles, realidad virtual y aumentada, videojuegos, *chatbots* e inteligencias artificiales varias, domótica, ciudades inteligentes... toda tecnología es susceptible de ser usada para reducir distancias entre las personas. O de aumentarlas si no se usan correctamente.

Muchas gracias a mis compañeras de viaje. A Olga Carreras por ser una generosa e inagotable vía de conocimiento; a Emmanuelle Gutiérrez y Restrepo por arroparnos en la Fundación Sidar; y a Olga C. Santos por aportar su visión y experiencia académica. También muchas gracias a todas las personas que nos contactáis por las redes sociales para expresarnos vuestro aliento. Este libro divulgativo es para que, desde el puesto que tengamos, más lejos o más cerca de la accesibilidad, aportemos nuestro granito a hacer de este mundo un lugar más accesible y mejor.

Prefacio

Olga Carreras
Usable y accesible

Me gustaría agradecer la cálida acogida que recibió el libro “Accesibilidad Web. WCAG 2.1 de forma sencilla” en 2018, este respaldo nos ha motivado en gran medida para embarcarnos en esta nueva edición. El libro que os presentamos ahora no sólo amplía su alcance para abordar los nuevos criterios de conformidad de las WCAG 2.2, recomendación desde el 5 de octubre de 2023, sino que también se presenta como una edición actualizada, revisada en profundidad y enriquecida con tres nuevos capítulos: “*Single Page Application (SPA)* accesibles”, “Sistemas de Diseño accesibles” y “Cuidar la escritura”.

Nuestro objetivo es continuar con el trabajo de difusión y concienciación que iniciamos juntas hace 5 años, con el compromiso de hacerlo de forma sencilla, accesible y gratuita para todas las personas.

Durante estos años hemos vivido la publicación de nuevos estándares, la aprobación de nuevas leyes y el nacimiento de nuevas iniciativas como el *AccessibleEU*. Vivimos la revolución tecnológica de la inteligencia artificial (IA), y ya es común habitual hablar de accesibilidad en contextos de realidad virtual, aumentada o mixta.

Cabría esperar que la accesibilidad en los sitios web o las aplicaciones móviles ya estuviera superada. Sin embargo, sigo detectando los mismos errores en las auditorías de accesibilidad actuales que en las que realizaba hace 5 años: incumplimiento de muchos criterios de conformidad debido al desconocimiento o la falta de comprensión de los mismos; documentos no accesibles; falta de aplicación del estándar WAI-ARIA o mala aplicación del mismo; así como dificultades para gestionar y mantener la accesibilidad en el tiempo.

En mi experiencia como formadora, nunca he encontrado a alguien que diseñe o implemente sitios web o aplicaciones no accesibles como una elección consciente, sino como resultado de la falta de información. La inmensa mayoría, cuando comprenden cómo acceden las personas con discapacidad a la web o a las aplicaciones móviles, las barreras que se encuentran, lo fácil que es a menudo evitarlas, y cómo beneficia a todas las personas, interiorizan con facilidad las buenas prácticas para aplicarlas en su trabajo diario. Las expresiones más habituales suelen ser “si hubiera sabido todo esto antes...”, “se me han abierto los ojos...”, “nunca me había planteado todo esto...”, “no conozco a nadie con discapacidad...”.

Creo que la divulgación, la concienciación y la formación constituyen un pilar básico para lograr que la tecnología sea de verdad para todos, y espero que este libro sea una piedra más en la construcción de ese pilar.

Ha sido un placer volver a compartir este viaje con el entusiasmo y la dedicación de Olga Revilla, así como un honor contar con el valioso respaldo de Emmanuelle Gutiérrez y Restrepo, una gran referente para todas las personas comprometidas con la inclusión digital.

Prólogo

Emmanuelle Gutiérrez y Restrepo
Patrono Fundador y directora de la Fundación Sidar

Seis años después de la exitosa versión centrada en las WCAG 2.1 las talentosas autoras Olga Revilla y Olga Carreras nos guían nuevamente por un viaje esencial para todos aquellos comprometidos con la creación de páginas y contenidos web accesibles. Al embarcarnos en la revisión de «Accesibilidad Web. WCAG 2.2 de forma sencilla», nos sumergimos en una nueva odisea por el vasto universo de la accesibilidad digital.

En un mundo digital en constante evolución, la accesibilidad web se erige como un pilar fundamental. Este libro no es simplemente una actualización de las pautas; es una brújula para aquellos que desean no solo comprender sino también implementar de manera efectiva las WCAG 2.2. La accesibilidad web se convierte, así, en un arte al alcance de todos.

En las siguientes páginas, los lectores encontrarán no solo la esencia de las Pautas de Accesibilidad para el Contenido Web, sino también un valioso compendio de prácticas cruciales para desarrolladores, diseñadores y responsables de contenido. A medida que las autoras exploran las novedades introducidas por las WCAG 2.2, nos invitan a descubrir cómo estas pautas continúan evolucionando para abrazar la diversidad y la inclusión.

Este libro no es solo un documento técnico; es un faro ético en el diseño de interacción. La inclusión de la ética refleja la responsabilidad que cada creador de contenido web tiene hacia sus usuarios, garantizando experiencias significativas y accesibles para todos.

La metodología WCAG-EM se erige como una herramienta vital, presentada de manera clara y aplicable en cada fase de evaluación. Desde definir el alcance hasta declarar la conformidad, los lectores son guiados a través de un proceso que transforma la teoría en acción.

No se trata solo de seguir pautas, sino de comprenderlas. Las autoras desglosan cada principio, cada pauta, en un formato claro y práctico. Desde la perceptibilidad hasta la robustez, cada concepto se ilustra con ejemplos y guías de implementación.

Además, este libro no se limita a las Pautas WCAG; explora herramientas esenciales como ARIA, aborda los retos de las *Single Page Applications*, y destaca la importancia de los sistemas de diseño accesibles. Las secciones sobre comunicación clara y lenguaje inclusivo subrayan la relevancia de un enfoque humano en el diseño.

En un mundo lleno de tecnologías emergentes, las WCAG 2.2 se presentan como un faro guía para garantizar que ningún usuario sea dejado atrás. Este libro, con su enfoque claro y directo, se convierte en una herramienta indispensable para todos los profesionales del diseño web comprometidos con la inclusión digital.

Es un honor presentar esta obra como una contribución destacada al campo de la accesibilidad web. Es un honor y ha sido un placer prologarla, en esta ocasión con la ayuda inestimable de nuestro querido y común amigo ChatGPT. Un honor y un orgullo seguir apoyando a miembros destacados de la comunidad del Seminario Sidar. Que sea un faro para aquellos que buscan no solo cumplir con estándares, sino también elevar el estándar de la experiencia en línea para todos.

Contenidos

Introducción a la Accesibilidad Web.....	9
Las Pautas WCAG	32
Los requisitos de conformidad.....	43
Evaluar la accesibilidad con WCAG-EM	53
Principio 1. Perceptible	63
Principio 2. Operable	111
Principio 3. Comprensible.....	156
Principio 4. Robusto.....	183
Documentos PDF accesibles.....	190
ARIA, el aliado del HTML accesible	213
Single Page Applications (SPA) accesibles.....	243
Sistemas de diseño accesibles.....	251
Cuidar la escritura.....	262
Herramientas de validación	281
Herramientas de trabajo.....	299
Resúmenes y esquemas.....	308
Glosario.....	329
Índice global.....	332



Introducción a la **Accesibilidad Web**

En este capítulo presentamos el concepto de accesibilidad web, su importancia y cómo hacerlo en cada paso de una interacción.

Por otro lado, relacionamos dos conceptos estrechamente unidos a la accesibilidad: la usabilidad, y la ética en el diseño de interacción.

También recomendamos los conocimientos mínimos que deben tener los diferentes profesionales que participan en la construcción de los sitios web.

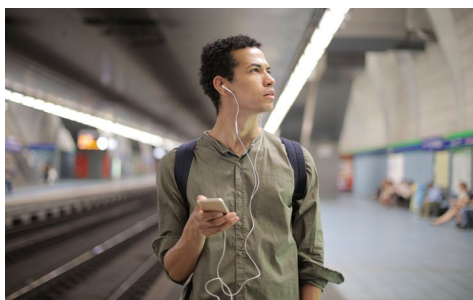
Por último, mostramos cómo implantar y gestionar la política de accesibilidad web en una organización.

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”, y muchos no sabrán por dónde empezar. Muchas personas piensan que hacer una web o una app accesible es para que las personas ciegas “puedan ver” el contenido de las pantallas. Sin embargo, el concepto de Accesibilidad Web va mucho más allá de las personas ciegas o de “poder ver” las pantallas. Y también es mucho más que un mero requisito funcional o una norma impuesta.

La Accesibilidad Web es una oportunidad para conseguir que el mayor número posible de personas puedan percibir, comprender y operar nuestras webs o apps en distintos tipos de dispositivos.

Fíjate en las siguientes dos personas usando su móvil para recibir las indicaciones necesarias para llegar a su destino. La primera es una persona ciega con auriculares que escucha las indicaciones manipulando la pantalla de su móvil; y la segunda es una persona que conduce un coche interactuando con el móvil a través de la voz. Las dos personas tienen limitaciones en su visión y manejo de la pantalla, una de ellas permanente y otra situacional y, sin embargo, ambas se están beneficiando de un diseño accesible que les permite acceder al contenido y las funcionalidades necesarias para conseguir su propósito.



Las limitaciones que afectan a las personas con discapacidad de forma permanente pueden afectar a otras personas de forma temporal, o bien pueden experimentar la limitación por la situación en la que se encuentran en un momento determinado. En la Tabla 1 tienes algunas limitaciones y cómo una misma solución puede ayudar a todas las personas, basado en el [Toolkit de Microsoft "Diseño inclusivo y situaciones discapacitantes"](#)¹

Tabla 1 La diversidad como fuente de inspiración

Limitación	Permanente	Temporal	Situacional	Solución común
Sin visión	Ceguera	Cataratas	No mirar a la pantalla, como al conducir un coche	Facilitar interacción por voz
Baja visión	Miopía magna	Conjuntivitis	Información legal en letra pequeña	Permitir hacer zoom en la pantalla
Dificultad para distinguir colores	Daltonismo	Deslumbramiento por luces fuertes	Pantalla de móvil oscurecida al tener poca batería	Contraste suficiente
Pérdida de audición	Sordera	No comprender bien el idioma	Entorno ruidoso; o uno silencioso, como una biblioteca	Subtitulado de un vídeo
Baja movilidad	Espasticidad, pérdida de un brazo, párkinson, artrosis	Esguince, tendinitis, síndrome del 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 la concentración	Autismo, trastorno por déficit de atención con hiperactividad	Estado de ansiedad	Realizar otra actividad, como cuidar niños mientras se mira el móvil	Simplicidad en la organización de los contenidos
Comprensión lectora	Dislexia, discapacidad intelectual, sordera de nacimiento	Bajo nivel educativo, bajo dominio del idioma	Entorno complejo, como una calle bulliciosa	Sencillez de los textos

Por ejemplo, los problemas que experimenta una persona sorda para entender un vídeo son los mismos problemas que experimenta un oyente con el mismo vídeo en un entorno muy ruidoso; o las dificultades que tiene una persona con párkinson para apuntar con un puntero en la pantalla son parecidas a las que puede tener una persona mayor al mover un ratón.

¹ <https://www.microsoft.com/design/inclusive/>

La discapacidad es una oportunidad de explorar cómo la tecnología crea situaciones limitantes y al mismo tiempo es capaz de resolverlas. 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.

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 rellenar 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 más que un mero requisito, es la oportunidad de generar un producto de calidad que facilite la vida de todas las personas que lo usan:** un problema de accesibilidad no sólo es bloqueante para una persona con discapacidad, sino que también es una molestia para muchas otras personas. Con la Accesibilidad Web conseguimos que todas las personas ejerzan su derecho a ser autónomas en internet, es decir, a ser capaces de acceder a los contenidos y funcionalidades independientemente de su diversidad funcional (sensorial, motriz, intelectual o mental) o del contexto de uso (por ejemplo, las condiciones tecnológicas o ambientales).



La **Accesibilidad Web** se define como el conjunto de características que debe incorporar un sitio web, una aplicación móvil o un documento digital para que el mayor número posible de personas en el mayor número posible de circunstancias pueda acceder a él, usarlo y comprenderlo.

La interacción accesible

La interacción es un proceso mediante el cual una persona y un sistema (es decir, cualquier dispositivo con una funcionalidad) se comunican entre sí a través de su interfaz.

Algunas formas de comunicación de **las personas con los sistemas** pueden ser:

- **Vocal:** por ejemplo, pidiéndole a Alexa, Siri o Google información del tiempo.
- Moviendo, señalando y seleccionando con un **puntero:** por ejemplo, con un ratón.
- Escribiendo con un **teclado físico:** por ejemplo, con un ordenador portátil.
- Escribiendo con un **teclado en pantalla:** por ejemplo, con un móvil.
- **Tocando** directamente una pantalla: por ejemplo, con una *tablet*.
- Introduciendo un objeto en una **ranura:** por ejemplo, la tarjeta de crédito en el cajero automático.
- Posicionando el **cuerpo** de una determinada manera: por ejemplo, en un videojuego de realidad virtual.

Algunas formas de comunicación de **los sistemas con las personas** pueden ser:

- **Sonora:** por ejemplo, al emitir un pitido indicando un éxito o un error.
- **Visual:** por ejemplo, mostrando en pantalla el efecto de mover el puntero o introducir un texto.
- **Háptica:** por ejemplo, en un juego de realidad virtual los mandos vibran cuando hay otro avatar cerca.
- Entregando un **papel impreso:** por ejemplo, generar un ticket de aparcamiento.
- **Térmica:** por ejemplo, cambiar la temperatura al apagar la calefacción.
- **Olfativa:** por ejemplo, un ambientador emitiendo un olor agradable al entrar la persona en la habitación.

Según las situaciones discapacitantes que se han recopilado en la Tabla 1, piensa en cómo las limitaciones permanentes, temporales o situacionales pueden afectar a esta comunicación. Por ejemplo, una persona con guantes de invierno o una persona con párkinson moderado tendrán dificultades para introducir la tarjeta en el cajero si la ranura exige mucha precisión. Del mismo modo, una persona sorda y una persona en una discoteca no oirán una llamada de teléfono.



En el capítulo [Herramientas de trabajo](#) encontrarás algunos simuladores para ponerte en los zapatos de personas que encuentran limitaciones en su interacción con los sistemas.

Para diseñar una interacción accesible debemos pensar en los **5 pasos que componen la interacción** de las personas con los sistemas, y realizarnos las siguientes preguntas:

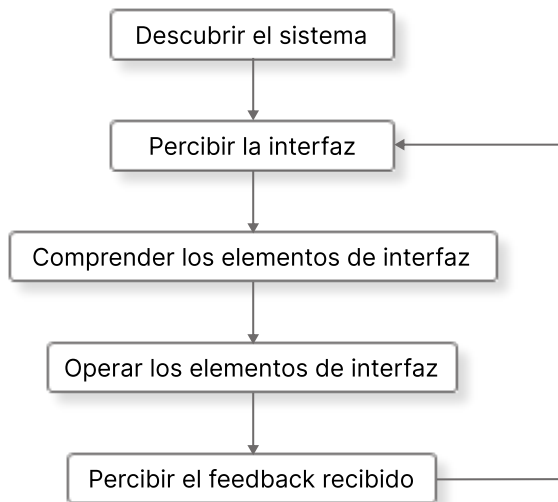


Ilustración 1 La secuencia de interacción

- **Paso 1. La persona conoce la existencia de un sistema dentro de su entorno.**
¿Cómo lo descubre? ¿Qué barreras y limitaciones existen que le impidan descubrirlo?
- **Paso 2. La persona percibe la interfaz, sus elementos, sus estados y sus propiedades.**
¿Cómo percibe la interfaz? ¿Qué sentidos están implicados? ¿Qué ocurre si una persona no puede usar uno o varios de los sentidos?
- **Paso 3. La persona comprende la interfaz, y las formas de comunicarse y manipular los controles (*affordances*).**
¿La información mostrada es suficiente? ¿Es excesiva? ¿Es fácil de comprender? ¿Se deja el suficiente tiempo para que se estudie?
- **Paso 4. La persona opera con los controles de la interfaz.**
¿Puede manipular los controles con facilidad? ¿Qué pasa si no puede utilizar el mecanismo de entrada de información? ¿Qué alternativa tiene? ¿Puede equivocarse?
- **Paso 5. La persona percibe y comprende el *feedback* de la operación,** y está dispuesta para volver a interactuar de nuevo con el sistema. De nuevo entran en juego las limitaciones de percepción y de comprensión. ¿Cómo percibe la retroalimentación? ¿Qué ocurre si no puede usar el sentido que está implicado? ¿Es fácil de comprender la información devuelta?



En el [video "Accesibilidad TIC. Secuencia de interacción"](#)², realizado por el profesor de la UNED Alejandro Rodríguez Ascaso, tienes un ejemplo práctico explicado.

A la hora de diseñar una interacción para un sistema, debemos prever las limitaciones (permanentes, temporales y situacionales) que se pueden dar en cada paso, y aportar soluciones para resolverlas en el caso de que se presenten. **Estas limitaciones pueden impedir, dificultar o reducir la capacidad de la persona para descubrir la interfaz, percibirla, comprenderla, u operarla.**

² <https://canal.uned.es/video/5a6f9224b1111f65148b459f>

La accesibilidad y la usabilidad

La accesibilidad, en el fondo, no deja de ser usabilidad de las interfaces extendida a un mayor rango de personas y situaciones. Cuando diseñamos interfaces accesibles las hacemos más usables. A continuación, se incluyen los [10 principios que Jakob Nielsen](#)³ utiliza para describir una interfaz usable y cómo se alinean con una interacción accesible, tal y como la hemos descrito en la Ilustración 1:

1. **Visibilizar el estado del sistema:** devolviendo una retroalimentación constante a la persona para que ésta siempre sepa qué está pasando. (Paso 5)
2. **Utilizar el mismo lenguaje que la persona que usa el sistema:** con mensajes comprensibles, con palabras y términos que le sean conocidos. (Paso 3)
3. **Dar control y libertad a la persona que usa el sistema:** el sistema debe adaptarse para que sea totalmente controlado por las personas. (Paso 2 y 4)
4. **Ser consistente y seguir estándares:** la persona debe ser capaz de entender el significado de las palabras, acciones o situaciones del sistema de acuerdo con los estándares de la plataforma. (Paso 2 y 3)
5. **Prevenir de errores:** el sistema debe ser capaz de prevenir y evitar los errores antes de que la persona se encuentre con algún mensaje indicando que algo ha fallado. (Paso 3 y 4)
6. **Minimizar la carga de memoria de la persona que usa el sistema:** el sistema debe ser capaz de minimizar esa carga a través de objetos o imágenes que faciliten a la persona su reconocimiento. (Paso 3)
7. **Ser flexible y eficiente en el uso:** el sistema debe reconocer al tipo de persona que lo usa y dejar que ésta personalice su experiencia de uso. (Paso 2 y 4)
8. **Ofrecer diálogos estéticos y un diseño minimalista:** el sistema debe ser capaz de aportar la mínima información relevante para la persona que lo usa, sin que ésta pierda de vista el contenido más importante del sitio web. (Paso 2)
9. **Ayudar a la persona que usa el sistema a reconocer, diagnosticar y recuperarse de los errores:** el sistema debe ser capaz de expresarlos en un lenguaje que la persona reconozca, aportando información sobre lo que ha ocurrido y proponiendo algún tipo de solución. (Paso 4 y 5)
10. **Proporcionar ayuda y documentación:** ubicadas en un lugar de fácil acceso para la persona que usa el sistema, escritas en un lenguaje que la persona conozca y, a ser posible, que no sean muy extensas. (Paso 3)

³ <https://www.nngroup.com/articles/ten-usability-heuristics/>

La ética en el diseño de la interacción

Diseñar una interacción implica mucho más que establecer una comunicación entre el humano y la máquina, hay que tener en cuenta los aspectos éticos de esta comunicación. En nuestra opinión, se deben contemplar los siguientes siete principios éticos en el diseño de interacción para que las personas controlen qué sucede, cuándo, dónde, cómo y por qué.

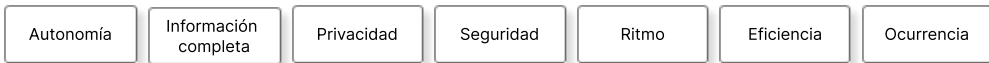


Ilustración 2 Principios éticos del control de la interacción

Autonomía	En primer lugar, la persona debe ser capaz de actuar por sí misma, sin necesidad de recurrir a otra persona. La autonomía de una persona es crucial para su autoestima e incluye la capacidad de hacer elecciones, tomar decisiones y asumir las consecuencias de éstas. Las personas necesitamos de interfaces para relacionarnos con las máquinas, en unos casos más habituales (como una pantalla) y en otros menos habituales (como un lector de pantalla). Diseñar y programar de forma robusta permitirá a las personas utilizar el mecanismo adecuado a sus necesidades.
Información completa	El sistema debe proporcionar la información suficiente y necesaria para que la persona sea consciente de las acciones que debe realizar y de sus consecuencias. No es necesario incluir absolutamente toda la información, de tal manera que sobrecargue la capacidad cognitiva y provoque tedio. Se trata de dar la información que se considere completa para ayudar a tomar una decisión informada. Esto es especialmente necesario en sitios transaccionales donde los actos pueden tener graves consecuencias, como una transferencia bancaria, una compra en una tienda o el envío de un currículo a una oferta de trabajo.
Privacidad	Además, el sistema debe garantizar la privacidad de la persona. No sólo garantizar los derechos establecidos por la ley de protección de datos de su país ⁴ , sino también que los contenidos de la conversación entre la persona y el sistema no se propaguen en el entorno sin su permiso. Por ejemplo, si se pide el número de la tarjeta de crédito mediante un teclado numérico, que el sistema no lo repita en voz alta para confirmar si es correcto o no. O si se está introduciendo una contraseña, tener un mecanismo para enmascararla y evitar que alguien la vea.
Seguridad	El principio de privacidad está muy ligado al siguiente principio, el de seguridad . La persona debe sentir que su conversación es segura, que realmente está sucediendo, que sus actos están teniendo consecuencias, que si hay un riesgo se le avisa, y de que si comete un error tiene la capacidad de recuperarse de ese error.

⁴ En la Unión Europea es el “Reglamento General de Protección de Datos (RGPD)”.

Ritmo

Por otro lado, la interacción puede ser puntual, como en mecanismos muy sencillos (por ejemplo, el timbre de la puerta), pero habitualmente es secuencial y tantas veces como la persona decida, por lo que también debemos tener en cuenta la variable “tiempo”. La persona debe manejar el **ritmo de la conversación** (en la medida que la técnica nos lo permita). ¿Tiene la persona tiempo suficiente para llevar a cabo las acciones o comprenderlas? ¿El sistema tarda en reaccionar y puede dar la impresión de que algo va mal?

Eficiencia

Al mismo tiempo, la conversación debe ser **eficiente**. Debe haber los clics y diálogos mínimos pero suficientes para que la persona no sufra en cada paso. Si la persona tiene que pasar mucho tiempo o dedicar mucho esfuerzo para completar una tarea, probablemente se frustrará y abandonará el proceso.

Ocurrencia

Del mismo modo, debe ser la persona quien decida **cuándo ocurre** esa conversación, cuándo empieza y cuándo termina la interacción, abandonando cuando lo desee. Todos tenemos ejemplos de actualizaciones del sistema operativo que bloquean el uso de nuestro dispositivo en el momento más inoportuno.



En general, si potenciamos el **control interno** de la persona, es decir, si percibe que los eventos ocurren principalmente como efecto de sus propias acciones, será más feliz y reaccionará mejor ante nuestro sistema.

Si, por el contrario, la persona percibe que no tiene ese control, sino que es **externo** a sus acciones, desarrollará ansiedad y rechazará nuestro sistema.

Qué profesionales deben saber de Accesibilidad Web

Tradicionalmente se ha vinculado la Accesibilidad Web a los desarrolladores⁵ de *front-end* y auditores de accesibilidad. Pensar que sólo estos perfiles están implicados en hacer un sitio accesible es tan erróneo como pensar que un edificio accesible es sólo cosa de los albañiles: si aguas arriba el resto de los profesionales no están implicados y preparados, los desarrolladores sólo podrán cubrir errores de la mejor manera que puedan.

Implementar una web accesible es el día a día de muchos profesionales digitales, en especial **creadores de contenidos, diseñadores UX/UI y desarrolladores de front**, que son quienes están en primera línea de batalla con el contenido, la interfaz y el código: cualquier cambio introducido, por pequeño que sea, puede suponer conseguir o perder la accesibilidad de un proceso o de un conjunto de páginas. Para ellos hemos dedicado un capítulo específico.

Sin embargo, **todos los perfiles implicados** en un sitio web influyen en el resultado final de la accesibilidad 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 cliente que lo ha contratado:

- 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 todas las personas;
- los **consultores de analítica** pueden detectar problemas de accesibilidad (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 subtítulo, los cambios intensos de luminosidad o el contraste del audio, entre otras cosas;
- los **revisores de calidad (QA)**, no sólo porque evalúan internamente el cumplimiento de las pautas con herramientas automatizadas, sino también porque deben incorporar en sus comprobaciones las herramientas que usan las personas con discapacidad y asegurar su correcto funcionamiento;
- 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:

- los **responsables de compras** que deciden la contratación de herramientas informáticas de gestión de contenidos deben conocer las implicaciones de accesibilidad de ese software, así como qué barreras o facilidades supondrán para el equipo que lo va a usar;

⁵ Por simplicidad del lenguaje, se utilizará el masculino genérico para referirse a las profesiones.

- 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 la normativa, así como de posibles litigios y multas por el incumplimiento de la legislación;
- el **director de marketing, comunicación y relaciones públicas** probablemente quieran 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 sólo en la web, sino en todos los aspectos de ésta.

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 UX/UI y desarrolladores es un error común que debemos desterrar y empezar a responsabilizarnos de la parte de accesibilidad que nos toca.



La organización internacional de estándares de internet **W3C** (*World Wide Web Consortium*) ha establecido una serie de responsabilidades por cada perfil profesional: [Accessibility Responsibility Breakdown](https://www.w3.org/community/wai-engage/wiki/Accessibility_Responsibility_Breakdown)⁶

⁶ https://www.w3.org/community/wai-engage/wiki/Accessibility_Responsibility_Breakdown

Qué debes saber según tu profesión

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 UX/UI 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 difusa línea que separa las responsabilidades de cada perfil: en algunas ocasiones, el diseñador UX/UI entra en el contenido, el creador de contenidos toca el código, o el propio desarrollador crea contenido audiovisual mediante código.

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 UX/UI**, 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 front** son los que tienen la responsabilidad de la maquetación HTML y CSS, con la implementación complementaria de ARIA, más el comportamiento en JavaScript (con 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 las personas que accedan al sitio web perciban y comprendan la información que se les quiere transmitir.

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

Para facilitar la **legibilidad** y comprensión de los textos deben estructurarlos 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 únicamente 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 ésta; 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. Por su parte, el título de la página debe ser corto, conciso y único en el sitio web.

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.

Ciertos destellos y movimientos pueden causar **ataques de epilepsia y mareos**, por lo que los creadores de contenidos 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 que la imagen, salvo que ésta sea decorativa, y evitar las imágenes de texto.

El uso del **color** es importante, debe tenerse en cuenta el contraste de los colores utilizados y no transmitir información únicamente 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 editor. El gestor permitirá en muchos casos marcar también las citas, las abreviaturas, el idioma de los contenidos, las celdas de encabezado o incluir resúmenes en las tablas.

Por último, como responsables de los **documentos** que se adjuntan al sitio web (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 UX/UI

Los diseñadores UX/UI deciden el aspecto gráfico, la organización de los elementos y su interacción dentro del sitio web.

En primer lugar, probablemente lo más conocido por todos los diseñadores UX/UI 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.

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 únicamente 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 UX/UI es la **consistencia**. Deben crear una estructura y un sistema de navegación consistente a lo largo de todo el sitio. También debemos informar a la persona 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.). Una de las novedades de las WCAG 2.2 precisamente es la homogeneidad de la ubicación de la ayuda o las formas de contacto, para que las personas puedan encontrarlas rápidamente.

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, la persona no puede tener ninguna duda sobre a dónde le llevará un enlace, 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 consistentes a lo largo de las páginas, el diseñador UX/UI 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 las personas que luego van a acceder a él podrán 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.

En relación con la disposición de elementos en pantalla, los diseñadores UX/UI deben tener en cuenta que los elementos que pueden recibir el **foco** no deben quedar tapados por otros elementos del diseño. Es habitual que las cabeceras y los pies fijos, u otras capas adhesivas, tapen el foco de teclado, además de que generen problemas al visualizar las páginas con zoom.

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 la persona se equivoque. En el caso de que se equivoque, identificar claramente los errores y ofrecerle sugerencias para que se recupere del error. Además, hay que facilitar la introducción de datos, sobre todo cuando ya se hayan introducido anteriormente. Hay que prestar especial atención al proceso de autenticación, para evitar que suponga un esfuerzo cognitivo que impida a algunas personas llevarlo a cabo.

La variedad de **tamaños** de pantallas es enorme, cosa que los diseñadores UX/UI 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 o, al menos, de 24 por 24 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 las personas controlar su reproducción, por lo que hay que diseñar los mecanismos necesarios para ello.

Los diseñadores UX/UI deben saber, al igual que los creadores de contenidos, que ciertos tipos de destellos y de movimientos pueden causar **ataques de 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 la persona 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 navega de una página web a otra.

A nivel de interacción, los diseñadores UX/UI deben asegurar que todas las personas puedan hacer un **manejo sencillo** del sitio web, en especial aquellas personas que tienen dificultades para manejar el ratón o el móvil. Aunque los desarrolladores aseguran la interacción por teclado, si se diseñan interacciones táctiles o de ratón complejas (gestos de arrastrar, gestos de deslizar, gestos con varios dedos, etc.), habrá que prever que se deben ofrecer alternativas de teclado y con gestos sencillos. Las WCAG 2.2. piden en especial que las interacciones de arrastre, como en los controles deslizantes o en los paneles o listas ordenables y movibles mediante *drag & drop*, tengan alternativa con un puntero.

Por último, otra forma de facilitar la vida a las personas que tienen menor capacidad para responder de forma rápida o son menos hábiles con la tecnología 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 de *front*

En este apartado incluimos los perfiles responsables del código de las páginas, como maquetar o definir el comportamiento.

En el pasado, los productos de apoyo, como los lectores de pantalla, interpretaban una página web directamente a partir de su código. En la actualidad, la mayoría de ellos lo hacen a través de la API de accesibilidad. Por eso, las WCAG 2.2 han decidido que ya no es un requisito obligatorio validar el código HTML o CSS, puesto que, si hay un error en el mismo que afecte a la accesibilidad, ya saldrá a la luz en otros criterios. Sin embargo, sigue siendo una buena práctica validar el código para verificar que sigue correctamente el estándar, ya que facilita no sólo la interpretación del código por las diferentes tecnologías, sino también por otros desarrolladores que tengan que analizar posteriormente ese código.

La principal responsabilidad de los desarrolladores es que el código sea **semánticamente** adecuado a la información, relaciones, secuencias y controles que genera. Para ello, se deberán 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 las personas 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 las personas usuarias de productos de apoyo.

Los desarrolladores deben ayudar a rellenar los **formularios** mediante una identificación clara y consistente de los campos, las ayudas e instrucciones, y los errores, de tal manera que sean perceptibles, operables y comprensibles para todas las personas que los vayan a cumplimentar. Deberán tener especial cuidado con los procesos de autenticación para no obligar a las personas a recordar o transcribir información. Permite que puedan copiar sus datos de acceso en los campos de texto y no incluyas en el proceso pruebas complicadas, como puzles o cálculos matemáticos.

Aunque es muy habitual ya el uso de archivos HTML, CSS y JavaScript para la estructura, la presentación y la funcionalidad respectivamente, los diseñadores UX/UI 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 y definiendo las imágenes decorativas en las CSS.

Es una responsabilidad compartida con el creador de contenidos que muchas de las imágenes del sitio web tengan una **alternativa** adecuada.

Por otra 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 el DOM.

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

Por otro lado, debe garantizar que las personas que navegan por el sitio web, incluidas aquellas que acceden con un lector de pantalla, obtengan el **feedback** necesario de las consecuencias de sus acciones, y tengan 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 el texto pueda ampliarse, que sea posible hacer zoom sin problemas, de tal manera que los elementos se reajusten sin necesidad de hacer *scroll* horizontal.

Otra forma de facilitar la vida a las personas 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.

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 pronunciarlos correctamente.

También pueden ser responsables del **título** de las páginas o el patrón a partir del cual se generan, por lo que deberán entonces tener en cuenta que tiene que ser corto, conciso y único en el sitio web e identificar claramente no sólo la página sino también el propio sitio web.

Implantar y gestionar la política de Accesibilidad Web de una organización

Integrar la Accesibilidad Web dentro de los procesos diarios de una organización 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 en los relativos a sitios web como a aplicaciones 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 y que hemos recopilado en la siguiente ilustración.

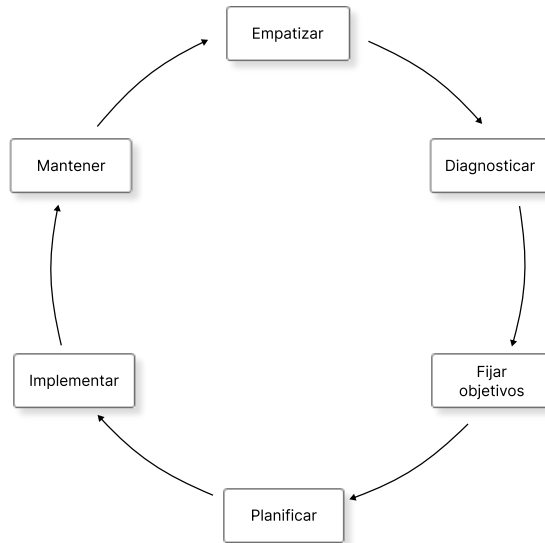


Ilustración 3 Ciclo de implementación y gestión de la Accesibilidad Web en una organización

En primer lugar, la organización debe **empatizar**, tomando 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 Herramientas de trabajo](#) puedes consultar algunos simuladores de discapacidades para que puedas intuir cómo perciben, comprenden y operan tu web muchas personas. Empatizar con las necesidades de las personas con discapacidad puede ayudarnos a integrar la Accesibilidad Web en nuestra organización, lo que proporciona beneficios adicionales, como los siguientes:

- Aumentar la facilidad de uso para todos, mejorando los **índices** de satisfacción y de experiencia de uso.
- Al mejorar la facilidad de uso, el **target** se expande, con la oportunidad de alcanzar un **mayor público**, y aumentar las 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.

Una buena práctica es utilizar la técnica de Personas para crear perfiles con limitaciones, bien permanentes, temporales o situacionales. En el epígrafe siguiente puedes ver varias creadas de ejemplo para una universidad online.

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 2.2 (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** 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 web.
- **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 web.
- **Incluir el *feedback* de las personas que acceden a nuestro sitio web** mediante un mecanismo de contacto específico que permita recoger sus quejas y sugerencias. El análisis y seguimiento de este *feedback* ayuda a definir acciones correctoras, a adaptar los sitios y aplicaciones web a las necesidades reales de las personas, y a la mejora continua.

Una vez que cada perfil de la organización 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 y la implicación de toda la organización. En el [capítulo *Evaluar 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. Es necesario seguir trabajando no sólo para **mantener** el objetivo perseguido, sino para alcanzar nuevos logros. Para ello, sigue siendo necesario formarse, estar al tanto de la nueva legislación, reevaluar el sitio web, implementar mejoras y revisar 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: [Planning and Managing Web Accessibility](http://www.w3.org/WAI/planning-and-managing)⁷

⁷ <http://www.w3.org/WAI/planning-and-managing>

Ejemplos de “Personas” para una universidad

PEDRO GARCÍA

Hombre, 32 años

Profesor de Estadística



DATOS RELEVANTES	Doctor con Matrícula de Honor. Tiene TDAH (Trastorno por Déficit de Atención e Hiperactividad)
PERFIL TECNOLÓGICO	Usa el ordenador y el móvil perfectamente y con autonomía. Tiene un ordenador viejo y lento.
OBJETIVOS	Enseñar las principales teorías a los alumnos de primer año.
RETOS	Tiene que preparar el material porque es su primer año.
OBSTÁCULOS	Le cuesta centrarse en un único tema.

EMILIA GUERRERO

Mujer, 45 años

Estudia Psicopedagogía



DATOS RELEVANTES	Trabaja a media jornada, el resto del tiempo lo ocupa en la crianza de sus dos hijas de 7 y 5 años.
PERFIL TECNOLÓGICO	Usa el móvil perfectamente y con autonomía, pero no tiene ordenador en casa.
OBJETIVOS	Aprender a manejar a sus hijas.
RETOS	Sacar tiempo para estudiar y hacer las tareas.
OBSTÁCULOS	Recibe muchos mensajes de los foros que no le interesan, necesita que le digan claramente lo que tiene que hacer.

SAID AHMAN

Hombre, 23 años

Estudia Ingeniería de Caminos



DATOS RELEVANTES	Estudiante de Erasmus, está haciendo un curso para aprender el idioma local.
PERFIL TECNOLÓGICO	Usa el ordenador y el móvil perfectamente y con autonomía. El teclado de su ordenador no está adaptado al español.
OBJETIVOS	Conocer la cultura local.
RETOS	Aprobar alguna asignatura.
OBSTÁCULOS	Apenas entiende el idioma y no conoce a nadie. Los vídeos de las clases no tienen los subtítulos y le cuesta seguir las explicaciones.

SANDRA LÓPEZ

Mujer, 23 años

Estudia Administración y Dirección de Empresas



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 apps que transcriben texto.

OBJETIVOS Terminar la universidad y hacer prácticas en una empresa.

RETOS Comprender las teorías económicas y estadística.

OBSTÁCULOS Los libros tienen un lenguaje muy complejo.
Los vídeos de las clases no tienen los subtítulos y le cuesta seguir las explicaciones.

ERIC DEL VALLE

Hombre, 43 años

Ayudante administrativo



DATOS RELEVANTES Acaba de ser padre, duerme muy poco.

PERFIL TECNOLÓGICO Usa el ordenador durante todo el día.

OBJETIVOS Procesar y verificar las matrículas y pagos de los estudiantes.

RETOS Aprender a utilizar la nueva intranet.

OBSTÁCULOS Cuando se implantó la herramienta él estaba de baja de paternidad y ahora tiene que aprender por su cuenta a través del manual.

LUISA SÁNCHEZ

Mujer, 19 años

Estudia Magisterio



DATOS RELEVANTES Es su primer año en la universidad, está un poco descolocada.

PERFIL TECNOLÓGICO Usa el móvil bien, pero apenas sabe usar el ordenador, sólo lo que le enseñaron en el instituto. No tiene uno en casa.

OBJETIVOS Sacarse la carrera y luego opositar a maestra.

RETOS Tiene que hacer muchos trabajos prácticos.

OBSTÁCULOS Recibe muchas notificaciones, emails y apuntes, no le da tiempo a mantenerse al día. Va muy lenta con el teclado de ordenador.



Las Pautas WCAG

En este capítulo presentamos las conocidas recomendaciones del consorcio W3C, incluyendo su evolución histórica, su estructura y la documentación que la compone.

Finalizamos el capítulo explicando su utilidad e importancia.

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 regionales](#)⁸.

Las [WCAG nacieron en su versión 1.0](#) en 1999 con 14 pautas y 65 puntos de comprobación⁹. En 2008 el W3C publicó las WCAG 2.0, una evolución de las anteriores que habían quedado obsoletas debido a los avances tecnológicos. Esta nueva versión contaba con 12 pautas que agrupaban 61 criterios de conformidad con más de 500 técnicas asociadas.

En 2018, el W3C aprobó las WCAG 2.1, una evolución de las pautas que abordaba una mayor diversidad de tecnologías, como las pantallas táctiles de los móviles, y daba mayor soporte a colectivos con baja visión, con discapacidad cognitiva o con dificultades para el aprendizaje. Las WCAG 2.1 incorporaron una nueva pauta (*Pauta 2.5 Formas de introducir la información*) y 17 nuevos criterios de conformidad (5 de nivel A, 7 de nivel AA y 5 de nivel AAA).

En 2023, el W3C aprobó las WCAG 2.2, una evolución que continua el trabajo de las WCAG 2.1 y responde a sus mismas necesidades. En esta nueva versión, se añaden 9 criterios de conformidad (2 de nivel A, 4 de nivel AA y 3 de nivel AAA) y se elimina un criterio que se considera obsoleto.

Tabla 2. Resumen comparativo de las diferentes versiones de las WCAG

Año	Versión	Pautas	Organizadas en
1999	WCAG 1.0	14 pautas	65 puntos de comprobación
2008	WCAG 2.0	12 pautas	61 criterios de conformidad
2018	WCAG 2.1	13 pautas	78 criterios de conformidad (17 criterios nuevos)
2023	WCAG 2.2	13 pautas	86 criterios de conformidad (9 criterios nuevos, 1 criterio eliminado)

⁸ <https://www.w3.org/WAI/policies/>

⁹ <https://www.w3.org/TR/WCAG10/>

De forma paralela, en enero de 2021 se publicó el primer borrador de las WCAG 3.0, que ha sido actualizado varias veces desde entonces. Hay que tener en cuenta que la elaboración y publicación de las WCAG 3 es un proceso que llevará años completar y que es paralelo a la evolución de las WCAG 2.

Las WCAG 3.0 han sido renombradas como [“W3C Accessibility Guidelines \(WCAG\) 3.0”](#)¹⁰, un nombre que permite mantener el acrónimo pero que, a la vez, refleja su nueva orientación hacia todo tipo de contenido, no sólo de contenido web.

El nuevo estándar traerá cambios muy importantes en diversos aspectos: filosofía, alcance, estructura, contenido o modelo de conformidad. Pero habrá que esperar a su publicación final para poder aplicarlo, ya que se está trabajando en asegurar que las recomendaciones que se definan cumplan su objetivo. Mientras, debemos aplicar las WCAG 2.2, que son las que están actualmente validadas y aprobadas.

Novedades introducidas por las Pautas WCAG 2.2

Las pautas WCAG 2.2 de 2023 eliminan el criterio de conformidad 4.1.1 “Procesamiento” e incorporan 9 criterios de conformidad nuevos.

En concreto, los nuevos criterios y sus niveles son:

- 2.4.11 Foco no oculto (Mínimo) (AA)
- 2.4.12 Foco no oculto (Mejorado) (AAA)
- 2.4.13 Apariencia del foco (AAA)
- 2.5.7 Movimientos de arrastre (AA)
- 2.5.8 Tamaño del área de interacción (Mínimo) (AA)
- 3.2.6 Ayuda consistente (A)
- 3.3.7 Entrada redundante (A)
- 3.3.8 Autenticación accesible (AA)
- 3.3.9 Autenticación accesible (Mejorada) (AAA)

¹⁰ <https://www.w3.org/TR/wcag-3.0/>

Documentos que componen las Pautas WCAG 2

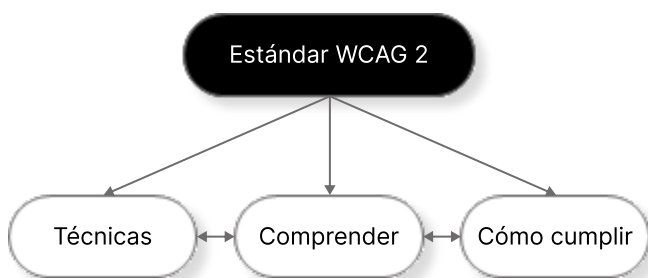


Ilustración 4 Composición de las WCAG

Las WCAG 2 se componen de **cuatro documentos interconectados**.

Por una parte, el propio “Estándar”, que es normativo y no sufre cambios; y, por otra parte, tres documentos informativos que se actualizan cada cierto tiempo:

- **Cómo cumplir con las WCAG 2:** es una guía de referencia rápida y la herramienta principal que incluye los requisitos (criterios de conformidad y técnicas).
- **Técnicas para las WCAG 2:** proporcionan información específica, como fragmentos de código HTML accesible. Son informativas, por lo que no es obligatorio usarlas. La conformidad con WCAG 2 se basa en los criterios de conformidad, no en las técnicas.
- **Comprender las WCAG 2:** es una referencia adicional para aprender a implementar las WCAG 2 y está dirigida a personas que quieran comprender las pautas y criterios de conformidad en profundidad.

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

Cómo se organizan las Pautas WCAG 2.2

Las WCAG 2 están organizadas en cuatro niveles:

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

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

86 criterios de conformidad

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).

Alrededor de 600 técnicas y errores a evitar

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.

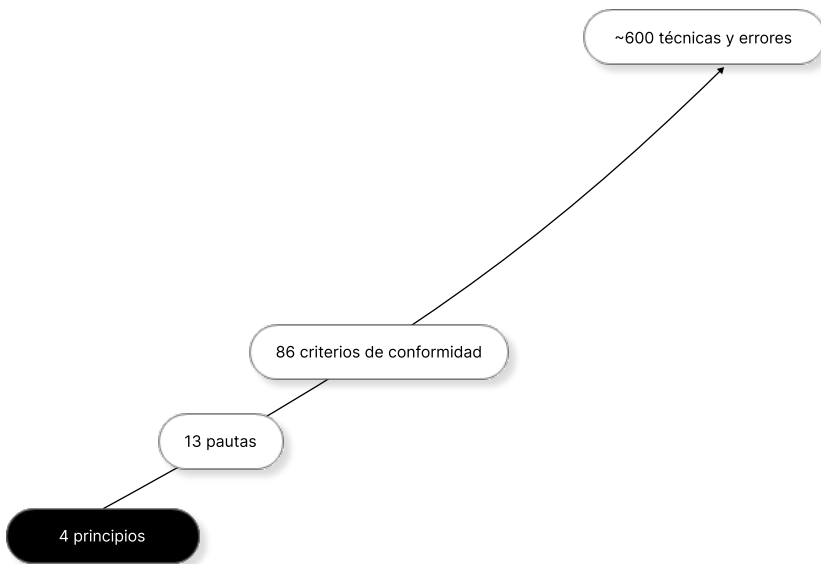


Ilustración 5 Organización de las WCAG 2.2

Los cuatro principios y sus pautas

Como decíamos antes, las WCAG 2.2 se organizan en principios, y estos en pautas. El objetivo es superar los cuatro principios, para lo cual debemos cumplir todas sus pautas.

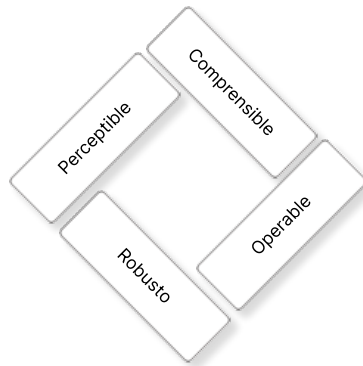


Ilustración 6 Los cuatro principios

El sitio web debe ser “Perceptible”

Nuestro sitio web puede ser visitado por personas con diferentes tipos de preferencias y capacidades, también por robots (buscadores, traductores...). Nuestra información e interfaz deben tener en cuenta esta necesidad y, por ello, debemos ofrecer alternativas a las personas 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** Distinguido: el contenido es fácil de ver y escuchar

El sitio web debe ser “Operable”

Los diseñadores UX/UI 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** Las personas tienen tiempo suficiente para leer y utilizar el contenido
- Pauta 2.3** El contenido no causa convulsiones
- Pauta 2.4** Navegable: las personas pueden navegar, encontrar contenido y saber dónde están en todo momento
- Pauta 2.5** Facilita formas de introducir información

El sitio web debe ser “Comprensible”

Si la persona que accede a nuestro sitio web 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 transmitido e interpretado por los diferentes agentes de usuario.

Los **agentes de usuario** son todos aquellos programas que muestran contenidos de internet, como los navegadores (Chrome, Firefox, Edge, Safari, Internet Explorer ...), reproductores multimedia (QuickTime, Real Player, Windows Media Player...), *plugins* (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 las personas con discapacidad, más allá de la que proporcionan las aplicaciones que se usan habitualmente. Son productos de apoyo, por ejemplo, un magnificador de pantalla, un teclado en pantalla, un software de reconocimiento de voz, un lector de pantalla...

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 requisitos técnicos concretos que se deben cumplir cuando se está creando un sitio web. Estos requisitos están enunciados de tal manera que se pueden comprobar si se cumplen o no.

Los criterios se clasifican en 3 niveles atendiendo a una serie de variables.

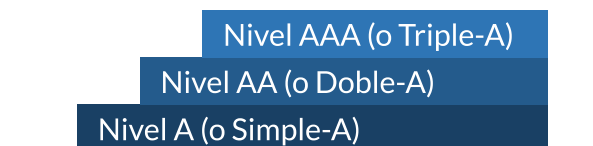


Ilustración 7 Niveles de los criterios de conformidad

- **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 debe tener cada criterio, el W3C valora las siguientes variables:

- **Esencialidad:** ¿Si no se cumple este criterio, los agentes de usuario (incluyendo los productos de apoyo) pueden o no reproducir la página web?
- **Universalidad:** ¿Este criterio de conformidad es aplicable para todos los sitios web, independientemente de su tema, del tipo de contenido, de la tecnología usada, etc.?
- **Aprendizaje:** ¿Los creadores de contenidos pueden aprender en menos de una semana a cumplir este criterio de conformidad?
- **Libertad:** ¿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?
- **Alternativas:** ¿Hay 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.



Al final del libro encontrarás todos los [criterios de conformidad organizados por niveles](#).

Las técnicas y los errores

Cada criterio de conformidad recomienda una serie de **técnicas** conocidas, documentadas y de uso extendido. Cada técnica dispone de un procedimiento de prueba que nos va a permitir saber si la estamos aplicando correctamente.

Observa que hemos utilizado el verbo “**recomendar**”, porque ninguna técnica es obligatoria, ya que las WCAG 2.2 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 los objetivos y la razón de ser del 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, ARIA, PDF...).

Los **errores** documentados por las WCAG 2.2 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.



Cuando explicamos cada criterio en este libro, identificamos las técnicas suficientes como “Técnicas para cumplir el criterio”; las técnicas aconsejables como “Técnicas para mejorar la usabilidad”; y los errores a evitar como “¡No lo hagas!”.

¿A qué tecnologías se aplican las WCAG?

Las WCAG 2.2 son neutrales con respecto de la tecnología, es decir, para construir un sitio web se puede utilizar cualquier tecnología. Existen dos conceptos relacionados: "tecnología compatible con la accesibilidad" y "dependencia tecnológica".

Una tecnología es **compatible con la accesibilidad** (en inglés, *accessibility supported technology*), como lo es una página HTML o un documento PDF, cuando las personas que usan 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 a la persona, y ésta, 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 *plugin*, como era el caso de Flash.

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 SPA (*Single Page Application*) es una página web cuyo contenido se modifica dinámicamente mediante JavaScript de manera asíncrona, usando a menudo *frameworks* como Angular o librerías con React. Estas aplicaciones pueden ser accesibles gracias en gran medida al estándar WAI-ARIA, por eso JavaScript se considera una tecnología compatible con la accesibilidad, pero genera una dependencia tecnológica.

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, HTML, 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, como el elemento de HTML *canvas*, o usamos tecnologías compatibles con la accesibilidad, como PDF, HTML o SVG, 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, y actualizarse a las 2.2

1. Porque son la recomendación oficial del **W3C**, la organización de desarrollo de estándares web más reconocida del mundo.
2. Porque son el fruto de muchos **años de trabajo** y del consenso de una comunidad internacional de especialistas en accesibilidad.
3. Porque las WCAG 2 han sido **reconocidas** 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. La EN 301 549 incluye actualmente las WCAG 2.1 y se espera que se actualice a las WCAG 2.2 en su próxima revisión ya en curso.
4. Porque las WCAG 2.1 son **legalmente obligatorias** en muchos países, como en los estados miembros de la Unión Europea o en EEUU, y se espera que adopten las WCAG 2.2, por tanto, sería buena idea que fueras teniéndolas ya en cuenta. Por ejemplo, en Europa, al estar integradas en la norma EN 301 549, las WCAG 2.1 son obligación legal para los sitios web y aplicaciones móviles del sector público desde la aprobación de la Directiva 2016/2102 (traspuesta a la legislación española mediante el Real Decreto 1112/2018); y para el sector privado desde la aprobación de la Directiva 2019/882, conocida como la *European Accessibility Act* (traspuesta a la legislación española mediante la Ley 11/2023). Puedes leer más en [legislación sobre accesibilidad digital en el blog de Olga Carreras](#) ¹¹
5. Porque son **sumatorios**: cumpliendo con las WCAG 2.2 también se cumple con las WCAG 2.1 y con las WCAG 2.0, dado que es una actualización que suma, no cambia los criterios anteriores.
6. Porque se puede comprobar su cumplimiento. Incluso cuentan con su propia **metodología de evaluación oficial**, perfectamente aplicable también con las WCAG 2.2. Se explica esta metodología en el [capítulo "Evaluar la accesibilidad con WCAG-EM"](#).
7. Porque existen **validadores** automáticos y herramientas que facilitan la evaluación. Nos servirán igualmente los validadores de las WCAG 2.0 o las WCAG 2.1, pero pronto veremos que comienzan a actualizarse a las WCAG 2.2.
8. Porque sirven para **todas las tecnologías web**, y están pensadas para no quedarse obsoletas.
9. Porque disponen de **normas claras** para evaluar la conformidad de una página.
10. Porque **demuestran** quién está realmente interesado en la accesibilidad.
11. Porque las WCAG 2.2 **mejoran** la accesibilidad para las personas que acceden mediante dispositivos móviles, y tienen más en cuenta las necesidades de las personas con baja visión y con discapacidad cognitiva.

¹¹ <https://olgacarreras.blogspot.com/2005/01/referencia-sobre-legislacin-espaola.html>



Los **requisitos** de **conformidad**

Implementar las WCAG es la base de la accesibilidad, pero no es su techo. Siempre podemos hacer más de lo que recogen las pautas.

En este capítulo presentamos los requerimientos mínimos que una página web debe cumplir para que se considere conforme, aunque pueden darse pasos más allá para conseguir un mayor grado de accesibilidad.

Además, explicamos cómo comunicar el nivel alcanzado mediante las declaraciones de conformidad.

Los 5 requisitos

Para que el contenido de un sitio se considere “conforme” con las WCAG 2.2 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 2.2 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 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 (A).
- 2.1.2 Sin trampas para el foco del teclado (A).
- 2.2.2 Poner en pausa, parar y ocultar (A).
- 2.3.1 Umbral de tres destellos o menos (A).

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 31 criterios de conformidad de nivel A.
- **Nivel AA:** la página (o su alternativa) cumple 55 criterios de conformidad: los 31 de nivel A más los 24 de nivel AA.
- **Nivel AAA:** la página (o su alternativa) cumple los 86 criterios de conformidad: los 31 de nivel A, los 24 de nivel AA y los 31 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 Europa, en EEUU, y en muchos otros países, el nivel mínimo exigido es el 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:** crear y/o mejorar un sitio web conlleva una serie de recursos humanos, técnicos y económicos que no todas las organizaciones pueden permitirse. La formación del personal, las herramientas a nuestro alcance o el tiempo necesario son elementos clave que determinan 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, como se ha visto en los apartados anteriores, 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 para hacerlos accesibles que crearlos accesibles desde cero con una buena planificación**. 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 sólo al final de este.

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.

Existen varias maneras de declarar la conformidad.

- La primera manera es a través del modelo que proporcionan **las WCAG 2.2**. A continuación, encontrarás una sección con ejemplos y una plantilla que te ayudará a crear una declaración de conformidad según la información requerida por las WCAG 2.2.
- En los estados miembro de la **Unión Europea** y, por tanto, también en España, los sitios web y las aplicaciones móviles del sector público (o que reciben financiación pública) tienen la obligación de incluir una declaración de conformidad con una estructura y unos contenidos muy concretos, especificados por la Comisión Europea. Hemos incluido un apartado explicando cómo deben ser estas declaraciones de conformidad.
- Para países como **EEUU** hablamos de las declaraciones de conformidad ACR (*Accessibility Conformance Report*) y de VPAT® (*Voluntary Product Accessibility Template*). Aclaremos esta terminología en el último apartado de este capítulo.

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

Declaración de conformidad de las WCAG 2.2

Las declaraciones de conformidad de las WCAG no son obligatorias, pero, si se utilizan, deben incluir determinada información necesaria, y otra opcional. Esta información se debe incluir en formato de texto natural, y opcionalmente con [metadatos](#)¹² legibles por máquinas.

Tabla 3 Información a incluir en la declaración de conformidad de las WCAG 2.2

Información obligatoria	Información opcional
<p>Fecha de la declaración de conformidad.</p> <p>Las pautas que se han seguido: el título, la versión y la URI.</p> <p>El nivel de conformidad alcanzado: A, AA o AAA.</p> <p>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.</p> <p>Las tecnologías de contenido web usadas de las que se depende, listadas en la propia declaración, o en una página aparte.</p>	<p>Criterios de conformidad satisfechos superiores al nivel declarado: <i>“También se cumplen los siguientes criterios de conformidad ... de nivel ...”</i></p> <p>Las tecnologías de contenido web usadas, pero de las que no se depende, listadas de la misma manera que las que sí generan dependencia.</p> <p>Agentes de usuario y productos de apoyo con los que se han probado las páginas: <i>“Los agentes de usuario y productos de apoyo utilizados en las pruebas son: [navegador+versión] en [sistema operativo+versión] con [plug-in+versión] y con [producto de apoyo+versión]”</i></p> <p>Pasos dados más allá de lo indicado por las WCAG 2.2 por ejemplo, pruebas de usabilidad con personas con discapacidad.</p> <p>Lista de características de accesibilidad específicas del contenido.</p>

Tabla 4 Plantillas para incluir la información obligatoria en la declaración de conformidad de las WCAG 2.2

Si aplica al sitio completo	Si aplica a una única página	Si aplica a varias páginas
<p>“A [fecha], todas las páginas web en [url del dominio] cumplen con las Pautas de Accesibilidad para el Contenido Web 2.2 (https://www.w3.org/TR/WCA G22) Nivel de conformidad [A, AA, AAA]. Las tecnologías utilizadas de las que se depende son...”</p>	<p>“A [fecha], la página web [título de la página] en [URI de la página] cumple con las Pautas de Accesibilidad para el Contenido Web 2.2 (https://www.w3.org/TR/WCA G22/) Nivel de conformidad [A, AA, AAA]. Las tecnologías utilizadas de las que se depende son...”</p>	<p>“A [fecha], las páginas web listadas a continuación: - [título de la página 1] en [URL de la página 1] - ... - [título de la página n] en [URL de la página n] cumplen con las Pautas de Accesibilidad para el Contenido Web 2.2 (https://www.w3.org/TR/WCA G22/) Nivel de conformidad [A, AA, AAA]. Las tecnologías utilizadas de las que se depende son...”¹³</p>

¹² <https://www.w3.org/WAI/WCAG22/Understanding/understanding-metadata>

¹³ Si solamente es aplicable a varias páginas del sitio, puedes listar las páginas en forma de listado por simplicidad como recomendamos, pero también se pueden usar expresiones regulares o lógica booleana.

En el caso de que la **página web no sea estática** y cambie sin el control de la organización (por ejemplo, los comentarios en un blog, la publicidad insertada dinámicamente o un widget de Twitter-X), es prácticamente imposible revisar continuamente si lo que se añade es accesible. Para solucionar este problema, las WCAG 2.2 ofrecen dos posibles soluciones:

- 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. 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 2.2, pero podría cumplirlo si se quitaran los siguientes contenidos procedentes de fuentes no controladas: ...”

Otra figura que contemplan las WCAG 2.2 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 las personas que accedan a la versión en chino 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 2.2, pero podría cumplirlo si se hubiera posibilitado el uso accesible para el idioma chino.”

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



Ilustración 8 Logotipos de conformidad según niveles



Puedes descargar los logotipos de conformidad en el sitio oficial del W3C: [Adding WCAG Conformance Logos](https://www.w3.org/WAI/standards-guidelines/wcag/conformance-logos/)¹⁴

¹⁴ <https://www.w3.org/WAI/standards-guidelines/wcag/conformance-logos/>

Plantilla de declaración de conformidad de acuerdo con las WCAG 2.2

Conformidad (Obligatorio)	
Fecha de la conformidad	[dd/mm/aaaa]
Pautas	WCAG - <i>Web Content Accessibility Guidelines</i>
Versión	2.2
URI	https://www.w3.org/TR/WCAG22/
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]
Java	...	[Sí No]
...	...	[Sí No]

Agentes de usuario utilizados en las pruebas (Opcional)				
Navegador	Versión	Sistema Operativo	Plug-ins	Productos de apoyo
Chrome
Edge
...

Declaración de conformidad en la Unión Europea (y España)

La Comisión Europea, a través de la [Decisión de Ejecución \(UE\) 2018/1523](#)¹⁵, publicó el modelo de declaración de accesibilidad de conformidad con la Directiva 2016/2102 (transpuesta a la legislación española mediante el [Real Decreto 1112/2018](#)¹⁶) para los sitios web y aplicaciones móviles del sector público o que reciben financiación pública.



Todos los sitios web y aplicaciones móviles del sector público (o que reciben financiación pública) en Europa y, por tanto, en España, deben incluir una declaración de conformidad que siga este modelo.

La declaración de conformidad debe cumplir con los siguientes requisitos:

- Debe actualizarse **mínimo una vez al año**, o cada vez que se haga una revisión.
- Debe proporcionarse en formato **accesible**.
- Las afirmaciones sobre la conformidad **deberán ser exactas y estar basadas en una evaluación** llevada a cabo por el propio organismo público, o por un tercero, indicando el método empleado.
- En los **sitios web**, la declaración de conformidad estará disponible en un apartado del sitio enlazado desde todas las páginas y denominado "Accesibilidad", o su equivalente en el idioma en el que se encuentre la página.
- En las **apps nativas**, estará en el sitio web de la entidad junto con el enlace para su descarga, o bien se facilitará al descargarla de las plataformas de distribución de aplicaciones.

Por otra parte, la declaración de conformidad debe estar estructurada en los siguientes apartados:

- **Situación de cumplimiento:** se indica si el sitio web o aplicación móvil es plenamente conforme, parcialmente conforme o no conforme con la normativa y la legislación.
- **Contenido no accesible:** se indica todo aspecto no conforme haciendo referencia a los requisitos concretos de la norma de accesibilidad. También se especifica el contenido que no es conforme porque la entidad se ha acogido a la excepción de carga desproporcionada, o bien porque es contenido que no entra dentro del ámbito de la legislación, como el contenido de terceros.
- **Preparación de la declaración:** se indica la fecha de la declaración y de la última revisión, así como el método empleado.
- **Observaciones y datos de contacto:** incluye los mecanismos de contacto para informar al organismo de cualquier incumplimiento de los requisitos de accesibilidad y solicitar información y contenido en versión accesible, así como los datos de contacto de los responsables de la accesibilidad del sitio y de la tramitación de las solicitudes.
- **Procedimiento de aplicación:** incluye la información del procedimiento para recurrir una notificación o solicitud realizada que se considera insatisfactoria.

¹⁵<https://eur-lex.europa.eu/legal-content/ES/TXT/?qid=1539938081477&uri=CELEX:32018D1523>

¹⁶https://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-12699

- **Contenido opcional:** otra información que se considere apropiada, como medidas adicionales para alcanzar un grado de accesibilidad mayor al legalmente requerido; un teléfono de asistencia adicional para las personas con discapacidad o usuarias de productos de apoyo, etc.



Puedes consultar modelos y ejemplos de declaraciones de accesibilidad en la web del Observatorio de Accesibilidad Web (OAW): [Declaraciones de accesibilidad](#)¹⁷

17

https://administracionelectronica.gob.es/pae_Home/pae_Estrategias/pae_Accesibilidad/implantacion-rd-1112-2018/declaracion_accesibilidad.html

Declaraciones de conformidad ACR Y VPAT

Un ACR (*Accessibility Conformance Report*) es un informe de conformidad de accesibilidad, de un producto o servicio, que los proveedores entregan en EEUU en las contrataciones con las agencias gubernamentales, que se encuentran obligadas por la *Section 508*.

Sin embargo, cada vez es más habitual que los departamentos de adquisiciones soliciten este informe en el sector privado, especialmente en transacciones entre empresas (B2B).

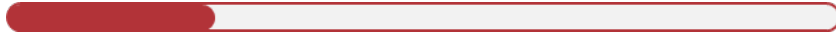
Estos informes suelen realizarse con la plantilla VPAT® (*Voluntary Product Accessibility Template*), creada por el *Information Technology Industry Council* (ITI), que está disponible en cuatro ediciones:

- **Edición WCAG:** para informar del cumplimiento de las WCAG 2.
- **Edición 508:** para informar del cumplimiento de la *Section 508*.
- **Edición UE:** para informar del cumplimiento de la EN 301 549.
- **Edición INT:** la edición internacional utilizada para informar del cumplimiento de los tres estándares anteriores.



Puedes consultar y descargar los [modelos VPAT® en la web oficial del ITI](#)¹⁸

¹⁸ <https://www.itic.org/policy/accessibility/vpat>



Evaluar la accesibilidad con **WCAG-EM**

En este capítulo presentamos la metodología de validación de sitios web completos, paso a paso.

También mostramos cómo se declara su conformidad y por último explicamos cómo usar una herramienta para aplicarla.

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.

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**.

Guías de interés de WCAG-EM

- [*Documentación oficial de la metodología WCAG-EM.*](#)¹⁹
- [*Traducción resumida de la metodología WCAG-EM de Olga Carreras.*](#)²⁰

¹⁹ <http://www.w3.org/TR/WCAG-EM/>

²⁰ <https://olgacarreras.blogspot.com.es/2012/04/metodologia-de-evaluacion-de.html>

Proceso de revisión

La metodología WCAG-EM consta de **5 pasos**, algunos de los cuales pueden solaparse o llevarse 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.

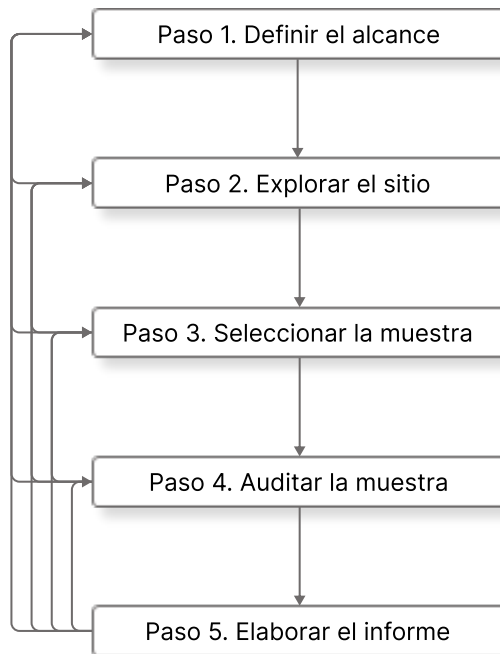


Ilustración 9 Pasos de la metodología WCAG-EM

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 un 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 estados²¹ 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 estas, 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, etc.

²¹ Las páginas web generadas dinámicamente pueden mostrar contenido, funcionalidad y apariencia diferentes según el usuario, la interacción, el dispositivo y otros parámetros. Un ejemplo sería la página de inicio de un sitio web con la sesión iniciada o sin iniciar: hay un cambio en una parte de la página, pero comparten la misma URI.

Paso 3. Seleccionar una muestra representativa de páginas

Lo ideal sería poder evaluar el sitio web completo, pero en muchos casos el volumen de páginas del sitio 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 esta, 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 únicamente 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 de forma estructurada**, en la que debemos incluir:

- Las páginas relevantes para todas las personas.
- 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 2.2

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, 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 vídeos en la página y el criterio evalúa si los vídeos están subtitrados), el criterio se considera “satisfecho” 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 que seleccionamos de forma estructurada.



En el [capítulo Herramientas de validación](#) puedes encontrar aplicaciones y documentos 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 (como capturas de pantalla o el código fuente analizado), ya que puede ser muy útil en caso de debate, así como para poder mejorar la accesibilidad y comprobar posteriormente dichas mejoras: 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 2.2 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 web 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 2.2.

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 2023, todas las páginas de <http://www.ejemplo.com> son conformes con las Web Content Accessibility Guidelines 2.2 (WCAG 2.2) 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 2023, 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 2.2 (WCAG 2.2).

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>”

Herramienta de ayuda WCAG-EM Report Tool

Para el proceso de auditoría, el W3C pone a disposición la herramienta [WCAG-EM Report Tool](https://www.w3.org/WAI/eval/report-tool/#/)²² que permite registrar todos los datos de la evaluación siguiendo la metodología WCAG-EM, y generar un informe automáticamente.

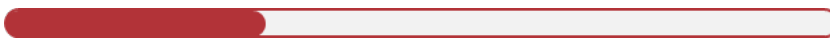
Aunque funciona online, se pueden guardar los datos en local y realizar la auditoría en diferentes momentos. Además, el informe generado se puede descargar en formato PDF y JSON.

Está disponible en inglés, holandés y francés para todas las versiones de las WCAG, incluidas las WCAG 2.2.

The screenshot displays the 'WCAG-EM Report Tool' interface. At the top, there is a navigation bar with the title 'WCAG-EM Report Tool' and a progress indicator showing five steps: 1. Define Scope (active), 2. Explore Website, 3. Select Sample, 4. Audit Sample, 5. Report Findings, and a 'View Report' link. The main content area is titled 'Step 1: Define the Evaluation Scope' and includes a brief 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.' Below this, there are several input fields: 'Website name' (text input), 'Scope of the website' (text area), 'WCAG Version' (dropdown menu set to 'WCAG 2.2'), 'Conformance target' (dropdown menu set to 'Level AA'), 'Accessibility support baseline' (text area), and 'Additional evaluation requirements' (text area). On the right side, a 'Your report' summary box shows 'Reported on 0 of 55 WCAG 2.2 AA Success Criteria' and a table of criteria: Perceivable (0 of 20), Operable (0 of 20), Understandable (0 of 13), and Robust (0 of 2). A 'View Report' button is located below this table. At the bottom of the form, there are navigation buttons for 'Previous step' (labeled 'Start') and 'Next step' (labeled 'Explore Website'). A footer section titled 'Help improve this page' provides contact information and links for 'Email', 'Fork & Edit on GitHub', and 'New GitHub Issue'.

Captura de pantalla 1 WCAG-EM Report Tool

²² <https://www.w3.org/WAI/eval/report-tool/#/>



Principio 1.

Perceptible

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

Nuestra información y los componentes de la interfaz de usuario deben tener en cuenta esta circunstancia.

Por ello, debemos proporcionar alternativas si la persona no puede usar alguno 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 de las personas. 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 a las personas tiene una alternativa textual que cumple el mismo propósito.

Excepciones:

- **Controles** o mecanismos para que la persona 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 muestra en pantalla: 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.

Alternativas de descripción corta

- Utiliza el atributo *alt* en imágenes.
- 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*.
- En un grupo de imágenes relacionadas entre sí, utiliza la descripción de una de ellas para describir el grupo.

Alternativas de descripción larga

- Utiliza el atributo *aria-describedby* en las imágenes.
- Utiliza el cuerpo en el elemento *object*.
- Incluye una descripción larga en otra ubicación, con un enlace inmediatamente adyacente al contenido no textual.
- Incluye una descripción larga cerca del contenido no textual, indicando en la descripción corta la ubicación de la descripción larga.

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 para incluir una descripción que tenga el mismo propósito y la misma información que el contenido no textual.

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 para proporcionar una breve descripción del contenido no textual, más una alternativa de descripción larga.

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

- Utiliza el atributo *aria-label* para etiquetar objetos, como un botón, una región o una función MathML.
- Utiliza el atributo *aria-labelledby* para concatenar una etiqueta desde varios nodos de texto, por ejemplo, en los campos de texto incluidos en una tabla.
- Utiliza el atributo *alt* en las *area* de mapas de imagen y en las imágenes usadas como botones.
- Utiliza el atributo *label* para asociar las etiquetas con los campos de formulario; y si no puedes usarlo, utiliza el atributo *title*.
- En los enlaces, proporciona un texto de enlace que describa su propósito, por ejemplo, si el enlace incluye una imagen con atributo *alt* y un texto, el texto del enlace será la combinación de ambos.

Si el contenido no textual es un audio o un video, grabado o en directo; es un examen o ejercicio que no sería válido si se presentara en modo texto; o su intención primaria es crear una experiencia sensorial:

- Proporciona una etiqueta descriptiva usando una de las técnicas de descripción corta.
- Para los audios en directo y los vídeos solo en directo, proporciona una alternativa de descripción corta que describa su propósito.
- Proporciona una alternativa de descripción corta que sea el nombre aceptado o un nombre descriptivo del contenido no textual.

Si el contenido no textual es un CAPTCHA:

- Proporciona un texto alternativo que describa su objetivo y ofrece otro CAPTCHA que sirva para el mismo propósito usando una modalidad diferente.

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

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



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

- [Guía para incluir textos alternativos adecuados a las imágenes](#)²³
- [Mapa de decisión para proporcionar textos alternativos adecuados a las imágenes](#)²⁴

Técnicas para mejorar la usabilidad

- Usa los atributos CSS *margin* y *padding* para formatear los espacios entre los elementos en vez de imágenes espaciadoras.

¡No lo hagas!

- No utilices las CSS para incluir imágenes que contengan información importante.
- No ofrezcas una alternativa textual que no incluya la información comunicada por las diferencias de color en la imagen.
- No olvides actualizar las alternativas textuales cuando cambie el contenido no textual.
- No utilices una alternativa textual que no sea realmente descriptiva. Por ejemplo, `alt="icon.jpg"` o `alt="imagen2"`.
- No marques las imágenes decorativas de una manera que permita a los productos de apoyo anunciarlas. Por ejemplo, no les añadas `role="img"`, o una descripción como `alt="spacer"`.
- No omitas el atributo `alt` en los elementos `img`, `area` e `input` de tipo `image`, aunque éstos se usen sólo para decorar. Deja el atributo vacío (`alt=""`).
- No proporciones descripciones largas que no transmitan el mismo propósito o presenten la misma información que el contenido no textual.
- No utilices caracteres que se parezcan a otros sin dar un texto alternativo. Por ejemplo, el carácter U+0063 y U+03F2 parecen la letra "c", pero el primero es del alfabeto occidental y el segundo del alfabeto griego, y las herramientas de conversión de texto a voz no los procesan de la misma manera.
- No utilices arte ASCII sin ofrecer una alternativa textual.

²³ https://www.usableyaccessible.com/textosalternativosaccesibles/texto_alternativo_wizard.php

²⁴

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

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 también, por ejemplo, un juego interactivo, que presenta contenido visual y/o auditivo y cuyos elementos de interacción usamos en determinados momentos.

Otro concepto que aparece en esta pauta es el de **documento alternativo al medio tempodependiente**, lo que entendemos habitualmente por transcripción. Una transcripción es un contenido que incluye una secuencia correcta de descripciones textuales de la información visual y/o auditiva y que permite a las personas que no pueden ver u oír conseguir los mismos resultados en cualquier interacción basada en el tiempo. Por ejemplo, el guion empleado para crear un contenido multimedia sincronizado puede representar con precisión el contenido resultante 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 vídeos sin audio en directo. En esos casos, utiliza el sentido común y los medios a tu alcance para ofrecer la alternativa más adecuada.

Tabla 5 Resumen de medios tempodependientes y alternativas

Medio	Tipo	Nivel	Alternativa	Criterio
Audio solo	Grabado	A	Transcripción textual	1.2.1
	En directo	AAA	Transcripción textual	1.2.9
Vídeo solo	Grabado	A	Transcripción textual o Pista de audio con la misma información	1.2.1
		AAA	Transcripción textual	1.2.8
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	AA	Subtítulos	1.2.4



Además, **todos los medios tempodependientes, ya sean en directo o grabados,** deben contar con un título y/o una descripción que permita a las personas 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.

Excepción:

- 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 audio que describa el contenido más importante del vídeo.

Técnicas para mejorar la usabilidad

- Usa el elemento *track* de HTML 5 para incluir una descripción textual en el elemento *video*.

¡No lo hagas!

- No utilices una alternativa textual que no sea descriptiva (por ejemplo, el nombre del fichero).
- 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:

- 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 ("murmullos 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, el elemento *track* del elemento *video* de HTML 5 u otro formato que soporte el reproductor.

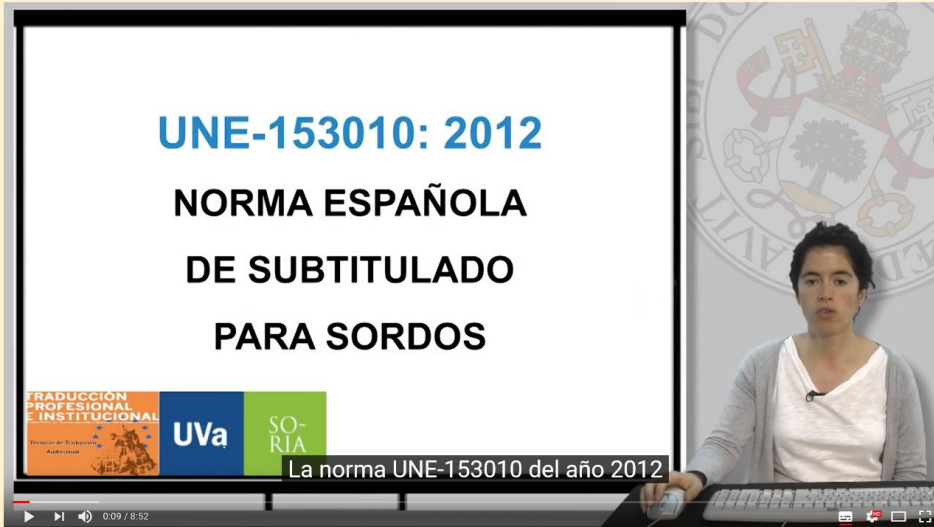
¡No lo hagas!

- 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 añada información adicional a la página.
- No te olvides de etiquetar la alternativa textual para un medio sincronizado como la alternativa de dicho medio.



La norma [UNE 153010:2012](#)²⁵ 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.).

En el [vídeo “UNE-153010:2012 Norma española de subtulado para sordos”](#)²⁶ de la Universidad de Valladolid la explican detalladamente.



²⁵ <https://www.aenor.com/normas-y-libros/buscador-de-normas/UNE/N0049426>

²⁶ <https://www.youtube.com/watch?v=UuUfnctQKYQ>

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:

- 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 diálogos.

Existe también la variante de **audiodescripción ampliada**, 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](#)²⁷ tienes un ejemplo de audiodescripción



²⁷ https://www.youtube.com/watch?v=S1MPQbcwS_Q

Técnicas para cumplir el criterio

- Ofrece una alternativa para medios tempodependientes, bien con un enlace a la alternativa situado junto al contenido no textual; bien usando el cuerpo del elemento *object*.
- Añade una segunda pista de sonido, seleccionable por la persona que está accediendo al contenido, que incluya la audiodescripción.
- Ofrece una versión del vídeo con audiodescripción -o con audiodescripción ampliada- usando SMIL o usando cualquier reproductor que soporte audio y vídeo.
- Ofrece un texto alternativo estático para describir un vídeo donde únicamente se muestra una persona hablando en un fondo invariable, como en una conferencia de prensa.

Técnicas para mejorar la usabilidad

- Añade audiodescripciones con el elemento *track* del elemento *video* de HTML 5.

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 la persona que accede al contenido, 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.
- Ofrece un texto alternativo estático para describir un vídeo donde únicamente se muestra una persona hablando en un fondo invariable, como en una conferencia de prensa.

Técnicas para mejorar la usabilidad

- Añade audiodescripciones con el elemento *track* del elemento *video* de HTML 5.

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, 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](#)²⁸ puede integrar la interpretación en lenguaje de signos tanto dentro del flujo (primera imagen) como en una pantalla diferente (segunda imagen). Esta innovación fue introducida a partir de [la tesis doctoral "Accesibilidad No Intrusiva en la Comunicación Audiovisual en la Web"](#)²⁹ de Emmanuelle Gutiérrez y Restrepo.

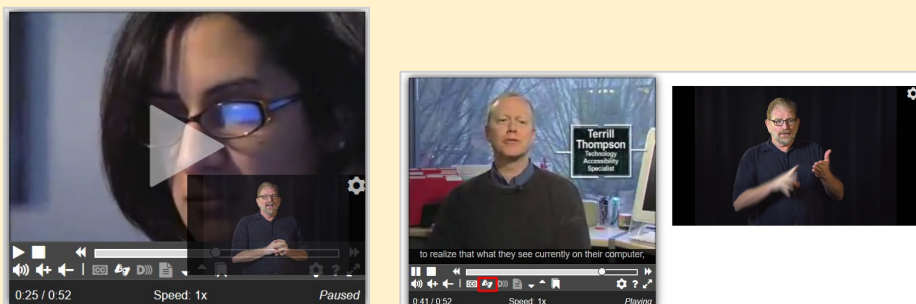


Ilustración 10 Integración de la lengua de signos en el reproductor Able Player

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 superpuesto en la imagen usando SMIL.

²⁸ <https://ableplayer.github.io/ableplayer/>

²⁹ <https://inclusiondigital.net/a11dnointrusiva/>

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 con el elemento *track* del elemento *video* de HTML 5.

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):

- Ofrece una alternativa para medios tempodependientes, bien con un enlace a la alternativa situado junto al contenido no textual; bien usando el cuerpo del elemento *object*.

Si el contenido es vídeo solo (grabado):

- Ofrece una alternativa que presente la misma información.

¡No lo hagas!

- No te olvides de etiquetar la alternativa textual para un medio sincronizado como la alternativa de dicho medio.

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 el contenido audio solo cuando se emite en directo.
- Incorpora un servicio de subtulado para sordos en directo en la página web.

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.

Las personas que acceden a tu sitio web tienen necesidades y preferencias muy diferentes, por lo que tienes que comprobar que toda la información esté disponible de tal forma que pueda ser percibida por todas las personas de diferentes maneras (al ser leída en voz alta, al ser presentada con un diseño visual más simple, etc.), con diferentes agentes de usuario y en diferentes dispositivos con distintas orientaciones de pantalla.

Si hay información incorporada en la presentación de la página que los productos de apoyo no pueden diferenciar o interpretar, no se podrá presentar en otros formatos según los necesite el usuario. Por ejemplo, la forma en la que se organizan las zonas de una página o los contenidos dentro de la misma, o la representación del contenido de una forma determinada. Para conseguirlo, hay que hacer que toda la información sea identificable y procesable mediante software.

Las WCAG 2.2 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 usuarios³⁰”.

³⁰ Aunque en esta edición hablamos de personas que usan el sitio web en vez de usuarios, mantenemos aquí el término al estar usado por el W3C en la definición.

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):

- Identifica las zonas y regiones de la página (cabecera, menú, zona principal y sus secciones, pie, etc.) mediante elementos HTML semánticos o mediante roles de ARIA.
- Utiliza el atributo *aria-labelledby* para nombrar las zonas y regiones de una página y los controles de la interfaz de usuario.
- Utiliza elementos semánticos para marcar la estructura y utiliza marcado semántico para marcar el texto especial o enfatizado. Por ejemplo, mediante etiquetas como *em*, *strong*, *blockquote*, *cite*, *sup*, etc.
- 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 a las personas disponer de diferentes presentaciones, por ejemplo, utilizando sus propias CSS.
- Identifica semánticamente una fuente de iconos con *role="img"*.
- Agrupa los enlaces relacionados usando el elemento *<nav>*.
- 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"*;
 - 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);
 - 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 asociar el título de la tabla con la tabla;
- el atributo *scope*, o los atributos *id* y *header*, para asociar las celdas de encabezado con las celdas de datos;

Si tienes un formulario, utiliza:

- 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 elemento *label*;
- los elementos *fieldset* y *legend* para describir grupos de controles de formulario, o roles ARIA de agrupación, como *role="group"* y *role="radiogroup"*;
- el atributo *optgroup* para agrupar elementos *option* dentro de un elemento *select*.
- el atributo *aria-labelledby* para nombrar los controles de la interfaz de usuario, como pueden ser los componentes de un formulario.

Si la tecnología utilizada 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, utilizando 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.
- Ubica las etiquetas para maximizar la predictibilidad de las relaciones entre éstas y los controles a los que representan.
- Utiliza el atributo *aria-describedby* para proporcionar una etiqueta más descriptiva a los controles de formulario, por ejemplo, asociando al campo una instrucción.
- Identifica los campos obligatorios de formulario con el atributo *aria-required*.
- Organiza la página usando encabezados.

¡No lo hagas!

- No utilices los cambios de presentación del texto para transmitir información sin utilizar el marcado o el texto adecuado. Por ejemplo, no simules un encabezado mediante un párrafo con un estilo determinado.
- No utilices caracteres de espacio en blanco (espacios, tabulaciones, saltos de línea o retorno de carro) para formatear texto en varias columnas, o para formatear contenido 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. Por ejemplo, no utilices una etiqueta de encabezado únicamente para crear un efecto visual.
- No utilices los elementos *th* o *caption*, ni los atributos *summary*, *headers* o *scope*, en las tablas de maquetación utilizadas para diseñar la página.
- No utilices el elemento *pre* para marcar información tabular.
- No insertes contenido informativo utilizando los selectores *:before* y *:after*, o la propiedad *content* en CSS.
- No te olvides de marcar las celdas de encabezado en las tablas de contenido; y de revisar que la asociación de las celdas de encabezado con las celdas de contenido mediante los atributos *headers* e *id* se ha realizado correctamente.
- No uses *role= "presentation"* (o su equivalente *role="none"*) para marcar contenidos con información semántica.

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.

Una secuencia correcta de lectura es cualquier secuencia donde las palabras y párrafos se presentan en un orden que no cambia el significado del contenido.

Técnicas para cumplir el criterio

- Ordena todo el contenido de la página web mediante una secuencia significativa.
- Marca las secuencias del contenido como significativas utilizando una de las siguientes técnicas, y ordena todo el contenido de la página web mediante una secuencia significativa.
 - utiliza marcas de Unicode para mezclar texto “de derecha a izquierda” y “de izquierda a derecha”.
 - utiliza el atributo *dir* en elementos en línea para resolver problemas con textos que necesitan diferentes direcciones de lectura y que se presentan anidados.
 - posiciona el contenido en base al marcado de estructura.
 - controla el espacio entre letras mediante el atributo CSS *letter-spacing*.
- Ajusta el orden del DOM al orden visual.

¡No lo hagas!

- No utilices caracteres de espacio en blanco (espacios, tabulaciones, saltos de línea o retorno de carro):
 - para formatear contenido como si fuera una tabla;
 - para formatear texto en varias columnas; o
 - para controlar el espaciado dentro de una palabra.
- No utilices una tabla para maquetar contenido que no tenga sentido cuando la tabla se muestre de forma lineal.
- No cambies el significado del contenido por la forma de ubicar la información mediante las CSS.

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.

Las personas usuarias de un 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 únicamente 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.

¡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

El contenido no obliga a ser visto y manejado en una única orientación (vertical u horizontal) de la pantalla.

Excepción:

- Una orientación específica es esencial para transmitir la información o para que el contenido pueda ser operado.

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

Este criterio responde a las necesidades de las personas que tienen la pantalla en una posición fija, por ejemplo, anclada a la silla de ruedas. El objetivo es que el contenido se pueda disfrutar en cualquier orientación, aunque para ello haya que reajustar el contenido a la orientación de la pantalla.

Técnicas para cumplir el criterio

- Dispón un control que permita acceder a los contenidos en diferentes orientaciones de pantalla, si la reorientación automática está restringida.

¡No lo hagas!

- No bloques la orientación de la pantalla.
- No muestres un mensaje solicitando a la persona que reoriente el dispositivo (salvo cuando una orientación específica es esencial).

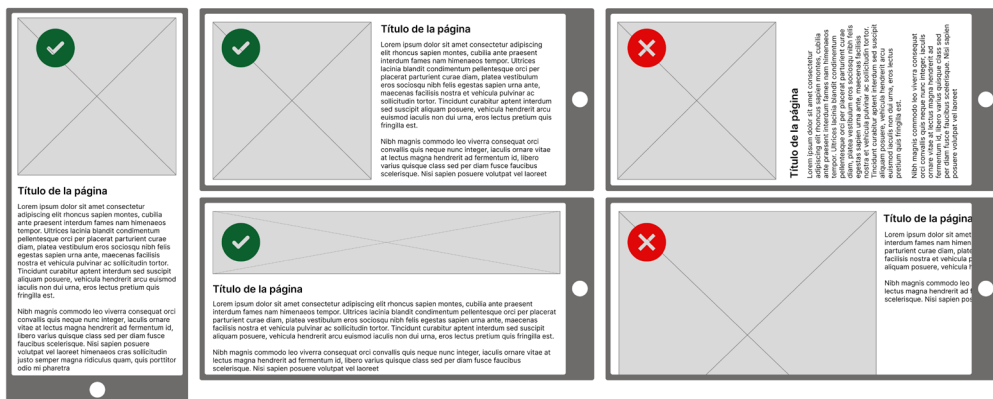


Ilustración 11 Formas correctas e incorrectas de disponer contenidos en formato apaisado dado el contenido vertical que se muestra en primer lugar

Identificación del propósito del campo

El propósito de cada campo que recoge información sobre la persona 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 los campos de un formulario, permitiendo además su procesamiento automático.

Por ejemplo, si el campo de formulario que nos pide el teléfono está identificado como tal en el código (mediante el atributo *autocomplete*), y ya hemos introducido anteriormente nuestro teléfono en el formulario de otro sitio web, el navegador puede autocompletar el campo, reduciendo el esfuerzo de completar el formulario y la posibilidad de cometer un error. Además, el navegador podría permitirnos personalizar la presentación de estos campos, por ejemplo, mediante pictogramas para identificar más claramente los campos del mismo tipo en diferentes formularios o sitios web.

El propósito identificado y concreto del campo debe ser uno de los listados por las WCAG 2.2. Se trata de un listado de campos habituales en la identificación de una persona, como nombre, teléfono o tarjeta de crédito. Más adelante mostramos el listado de valores que puede tener el atributo *autocomplete* y cómo se tiene que aplicar.

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 de uso de todas las personas.

Técnicas para cumplir el criterio

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

¡No lo hagas!

- No utilices valores incorrectos en el atributo *autocomplete*.

El atributo *autocomplete*

Los agentes de usuario a menudo tienen características que ayudan a las personas a rellenar formularios basándose en formularios rellenados anteriormente, y para ello se formuló el [atributo *autocomplete* dentro de HTML5.2](#)³¹.

Existen dos formas de implementar el atributo *autocomplete*:

- Como “ancla”, describiendo el significado del valor.
- Como “expectativa”, describiendo qué se espera de las personas.

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 la persona o porque nunca lo va a tener que reutilizar 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:
 - Opcionalmente, un *token* que empiece por “section-” para indicar que se agrupa con otros campos.
 - 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.
 - A continuación, una de estas dos opciones:
 - Un *token* con uno de los posibles valores listados en las páginas siguientes.
 - Dos *tokens* concretos, en el siguiente orden:
 - Una indicación de si el contacto es de casa (*home*), del trabajo (*work*), su móvil (*mobile*), un fax (*fax*), o un busca (*pager*).
 - El detalle del email (*email*), del teléfono, (*tel*, entre otras opciones), o de la mensajería instantánea (*impp*).

En los siguientes ejemplos, observa cómo se comporta el atributo *autocomplete* en el último paso del formulario de un proceso de compra.

Para el importe, los dos primeros campos son de tipo *hidden*, y funcionan como anclas, arrastrando la cantidad y la moneda de la transacción de pasos anteriores.

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

³¹ <http://www.w3.org/TR/html52/sec-forms.html#sec-autofill>

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.

```
<!-- 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 a la persona que accede a la página que introduzca la dirección de envío y la dirección de facturación. Observa cómo se concatenan los atributos *autocomplete*:

```
<!-- dirección de envío -->
<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">
```

Tabla 6 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 1ª región administrativa más alta (por ejemplo, la comunidad autónoma)	<i>text</i>
address-level2	En una dirección, la 2ª región administrativa más alta (por ejemplo, la provincia)	<i>text</i>
address-level3	En una dirección, la 3ª región administrativa más alta (por ejemplo, la localidad)	<i>text</i>
address-level4	En una dirección, la 4ª 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>
cc-type	Tipo de medio de pago	<i>text</i>
transaction-currency	Moneda de la transacción	<i>text</i>
transaction-amount	Cantidad de la transacción	<i>number</i>
language	Idioma favorito	<i>text</i>
bday	Fecha de nacimiento	<i>date</i>
bday-day	Día de la fecha de nacimiento	<i>number</i>
bday-month	Mes de la fecha de nacimiento	<i>number</i>
bday-year	Año de la fecha de nacimiento	<i>number</i>
sex	Sexo	<i>text</i>
url	Página web	<i>url</i>
photo	Fotografía	<i>url</i>
tel	Teléfono completo	<i>tel</i>

Autocomplete	Descripción	Tipo
tel-country-code	Código telefónico del país	<i>text</i>
tel-national	Teléfono sin el código telefónico del país	<i>text</i>
tel-area-code	Código telefónico de la región	<i>text</i>
tel-local	Teléfono sin el código del país ni de la región	<i>text</i>
tel-local-prefix	Prefijo local	<i>text</i>
tel-local-suffix	Sufijo local	<i>text</i>
tel-extension	Extensión telefónica	<i>text</i>
email	Correo electrónico	<i>email</i>
impp	Enlace directo mediante mensajería instantánea	<i>url</i>

Identificación del propósito

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

El objetivo de este criterio es mejorar la personalización y el ajuste a las preferencias de las personas. Por ejemplo, la identificación de las regiones de la página a nivel de código permite a las personas eliminar o resaltar regiones con su agente de usuario.

Muchas personas con vocabularios limitados dependen de términos o símbolos familiares para utilizar la web. Sin embargo, lo que resulta familiar para una persona puede no serlo para otra. Cuando los autores indican el propósito a nivel de código, las personas pueden aprovechar la personalización y las preferencias de usuario para cargar un conjunto de símbolos o un vocabulario que les resulte familiar.

Es similar a agregar información de rol (como lo requiere el criterio 4.1.2), pero en lugar de proporcionar información sobre qué es el componente de la interfaz de usuario (como una imagen), proporciona información sobre lo que representa el componente (como un enlace a la página de inicio).

Técnicas para cumplir el criterio

- Indica mediante programación el propósito de los iconos, las regiones y los componentes de la interfaz de usuario.
- Utiliza *landmark roles* de ARIA para identificar regiones de la página.
- Utiliza [microformatos](#)³² para marcar componentes de la interfaz de usuario. Los microformatos son pequeños patrones de HTML usados para representar semánticamente contactos de personas, eventos de calendario, entradas de blogs, lugares, etc.

Técnicas para mejorar la usabilidad

- Permite que los agentes de usuario encuentren la versión del contenido que mejor se adapte a sus necesidades.
- Utiliza semántica para identificar características importantes (por ejemplo, *coga-simplification="simplest"*)³³.
- Utiliza los atributos *aria-invalid* y *aria-required*.

³² <http://microformats.org>

³³ <https://rawgit.com/w3c/coga/master/techniques/index.html>

Pauta 1.4

Distinguible, separa el fondo del primer plano

Facilita a las personas 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.

Uso del color

El color no se utiliza como el único medio visual para 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 de determinadas palabras, fondos u otros contenidos:

- Comprueba que la información representada por los colores está disponible en texto.
- Incluye una indicación de texto para las etiquetas de los controles de formulario coloreadas, por ejemplo, para indicar que son campos obligatorios o con error.
- Comprueba que hay señales visuales adicionales cuando las diferencias de color en el texto transmiten una información.
- Cuando los enlaces o controles se diferencian solo por el color, como las palabras que son enlace dentro de un párrafo, comprueba que su color tiene una ratio mínima de contraste de 3:1 con el texto adyacente, y ofrece señales visuales adicionales cuando se pasa el cursor por encima de ellos, por ejemplo, que quede subrayado, en negrita o con un tamaño de fuente más grande.

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

- Diferencia los datos no sólo mediante los colores sino también con patrones o con 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

- Usa las CSS para cambiar la presentación de los componentes de la interfaz de usuario cuando reciben el foco.

¡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 de las personas para utilizar toda la página, **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 cerca del comienzo de la página que permita a las personas apagar los sonidos que suenan automáticamente.
- Reproduce los sonidos sólo a petición de las personas.

¡No lo hagas!

- No reproduzcas ningún sonido durante más de tres segundos sin un mecanismo para que las personas puedan apagarlo.
- No te olvides de incluir una forma para pausar o detener un elemento multimedia de HTML 5 que se reproduce automáticamente.

Contraste mínimo

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

Los textos e imágenes de texto con un tamaño de letra grande tienen una relación de contraste mínima de 3:1.

Excepciones:

- forman parte de un componente inactivo de la interfaz de usuario,
- son decorativos,
- no se presentan a las personas,
- 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 los colores del texto (incluido el texto dentro de las imágenes de texto) y el fondo debe ser:

Texto normal (redonda)		Texto en negrita	
Menor de 18 puntos	Igual o mayor de 18 puntos	Menor de 14 puntos	Igual o mayor de 14 puntos
4,5:1	3:1	4,5:1	3:1

La relación entre tamaños en puntos y píxeles CSS es 1pt = 1.333px, por tanto, 14 puntos y 18 puntos equivalen aproximadamente a 18.5 píxeles y 24 píxeles.

En ambos casos, como alternativa a cumplir con el contraste de color:

- Puedes no especificar el color de fondo ni el color de texto, y no cambiar los predefinidos.
- Puedes ofrecer un mecanismo con suficiente contraste para permitir a las personas cambiar a una presentación con suficiente contraste.

Técnicas para mejorar la usabilidad

- Elige una tecnología que permita cambiar el color de primer plano y de fondo de los bloques de texto. Estos controles son comunes en los agentes de usuario, por tanto, diseña páginas que funcionen con los navegadores más populares, de modo que el desarrollador no anule estos controles que permiten cambiar los colores.

¡No lo hagas!

- No especifiques colores de primer plano sin especificar colores de fondo o viceversa.
- No utilices imágenes de fondo que ofrezcan un contraste insuficiente con el texto (o con las imágenes de texto) de primer plano.

Cambio de tamaño del texto

Todo el texto puede ser ajustado sin productos de apoyo hasta un 200% sin pérdida de contenido o de funcionalidad.

Excepción:

- Este criterio no aplica ni a los subtítulos ni a las imágenes de texto.

Técnicas para cumplir el criterio

- Elige una tecnología que soporte zoom, común en los agentes de usuario habituales, como los navegadores o los lectores de archivos PDF.
- Garantiza que los contenedores de texto cambian de tamaño cuando el texto redimensiona, y usa medidas que sean relativas a otras, utilizando una o más de las siguientes técnicas:
 - utiliza unidades *em* para especificar el tamaño de los contenedores de texto;
 - utiliza medidas relativas para definir el tamaño de fuente, esto es, porcentajes, nombres de tamaños de fuente o unidades *em*.
 - para cambiar el tamaño de los contenedores de texto utiliza diseños líquidos, o calcula el tamaño y la posición de tal forma que escale con el tamaño del texto.
- Ofrece un mecanismo en la página web que permita a las personas incrementar el tamaño de todos los textos de la página hasta un 200%.
- Comprueba que no hay pérdida de contenido o funcionalidad cuando el texto redimensiona pero los contenedores de texto no cambian su tamaño.

Técnicas para mejorar la usabilidad

- Escala los elementos de formulario que contienen texto.
- Especifica medidas relativas para los anchos de columna de tal modo que las líneas puedan tener una media de 80 caracteres o menos cuando se redimensiona la ventana del navegador.
- Controla la presentación visual del texto mediante CSS.

¡No lo hagas!

- Evita que el texto, las imágenes o los controles queden cortados, superpuestos u ocultos cuando el texto se visualice a un 200%.
- No te olvides de redimensionar también los controles de formulario basados en texto cuando el texto se visualice a un 200%.
- No utilices de manera incorrecta las unidades *VW* (*Viewport Width*) en la definición del tamaño del texto, porque puede provocar que el texto no crezca al hacer zoom o al cambiar su tamaño.

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.

Excepciones:

- la imagen de texto es visualmente **configurable** según los requisitos de las personas;
- la forma particular de presentar el texto resulta **esencial** para la información que se transmite (por ejemplo, un logotipo).

Técnicas para cumplir el criterio

- Controla la presentación visual del texto mediante CSS.
- Utiliza CSS para reemplazar texto con imágenes de texto y ofrece un mecanismo a las personas para que reviertan el cambio.
- Separa la información y la estructura de la forma de presentación para permitir a las personas disponer de diferentes presentaciones, por ejemplo, utilizando sus propios estilos CSS.

Técnicas para mejorar la usabilidad

- Indica los tamaños de las fuentes en porcentajes, en unidades *em* o con nombres de tamaños de fuente.
- Controlar el espaciado de letras dentro de las palabras mediante el atributo CSS *letter-spacing*.
- Posiciona el contenido en base al marcado de estructura.

Contraste mejorado

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

Los textos e imágenes de texto con un tamaño de letra grande tienen una relación de contraste mínima de 4,5:1.

Excepciones:

- forman parte de un componente inactivo de la interfaz de usuario,
- son decorativos,
- no se presentan a las personas,
- 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:

Texto normal (redonda)		Texto en negrita	
Menor de 18 puntos	Igual o mayor de 18 puntos	Menor de 14 puntos	Igual o mayor de 14 puntos
7:1	4,5:1	7:1	4,5:1

En ambos casos, como alternativa a cumplir con el contraste de color:

- Puedes no especificar el color de fondo ni el color de texto, y no cambiar los predefinidos.
- Puedes ofrecer un mecanismo con suficiente contraste para permitir a las personas cambiar a una presentación con suficiente contraste.



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 grabado de audio solo 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.



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

En el capítulo [Herramientas de validación](#) 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 las personas.
- 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 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 las personas puedan 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...) en las CSS si no has especificado los colores de fondo y de texto del contenido principal.
- Especifica los bordes y el *layout* en las CSS para delimitar las áreas de la página si no has especificado los colores de fondo y de texto.
- Elige una tecnología que permita cambiar el color de primer plano y de fondo de los bloques de texto. Estos controles son comunes en los agentes de usuario, por tanto, diseña páginas que funcionen con los navegadores más populares, de modo que el desarrollador no anule estos controles que permiten cambiar los colores.
- No especifiques el color de fondo ni el color de texto, y no cambies los predefinidos.
- Ofrece en la página un mecanismo de selección de colores que permita a las personas 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 de tal modo que las líneas puedan tener una media de 80 caracteres o menos cuando se redimensiona la ventana del 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 en las CSS.
- 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:

- Ofrece un botón en la página que permita a las personas incrementar el espaciado de línea y de párrafo.
- Especifica el espaciado de línea en las CSS.

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 mediante una o más de estas técnicas:
 - utiliza medidas relativas para definir el tamaño de fuente, esto es, porcentajes, nombres de tamaños de fuente o unidades *em*.
 - utiliza porcentajes para especificar el tamaño de los contenedores de texto;
 - calcula el tamaño y la posición de los contenedores de texto de tal forma que escalen con el tamaño del texto.
- Ofrece opciones dentro del contenido para cambiar el diseño sin que requieran hacer *scroll* horizontal para leer el contenido.

¡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

- Controla la presentación visual del texto mediante CSS.
- Utiliza CSS para reemplazar texto con imágenes de texto y ofrece un mecanismo a las personas para que reviertan el cambio.
- Separa la información y la estructura de la forma de presentación para permitir a las personas disponer de diferentes presentaciones, por ejemplo, utilizando sus propias CSS.

Técnicas para mejorar la usabilidad

- Indica los tamaños de las fuentes en porcentajes, en unidades *em* o con nombres de tamaños de fuente.
- Controlar el espaciado de letras dentro de las palabras mediante el atributo CSS *letter-spacing*.
- Posiciona el contenido en base al marcado de estructura.

Reajuste de elementos

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.
- 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 a que puedan aumentar el tamaño del contenido hasta un 400% correctamente, sin que ello les obligue a desplazarse en dos direcciones.

320 píxeles CSS equivalen a un ancho de pantalla inicial (*viewport width*) de 1280 píxeles con 400% de zoom; mientras que 256 píxeles CSS equivalen a 1024 píxeles con 400% de zoom.

Ejemplos de contenidos que requieren de desplazamiento en dos dimensiones, y que entran dentro de las excepciones, son: las imágenes grandes como mapas o diagramas, videos, juegos, presentaciones, tablas de datos (no celdas individuales) o barras de herramientas que es necesario tener a la vista mientras se manipula el contenido.

Técnicas para cumplir el criterio

- Utiliza *media queries* y *grid* CSS para reajustar las columnas.
- Usa CSS Flexbox para reajustar el contenido.
- Permite el reajuste de cadenas de texto y URL largas.
- Utiliza los atributos *width*, *max-width* y Flexbox para ajustar las etiquetas y los campos de formulario.
- Calcula por programación tamaños y posiciones de los elementos de forma que escalen con el tamaño del texto.
- Da opciones dentro del contenido para cambiar a un diseño que no requiera *scroll* para leer una línea de texto.

Técnicas para mejorar la usabilidad

- Utiliza *media queries* para eliminar las cabecera y pies fijos.
- Utiliza los atributos *max-width* y *height* para ajustar las imágenes al tamaño de pantalla.
- Con CSS, reajusta las tablas de datos simples y haz que las celdas de datos quepan en el tamaño de la pantalla (*viewport*).
- Ofrece un mecanismo para cambiar a la vista móvil en cualquier momento.

¡No lo hagas!

- Evita que el contenido desaparezca o no esté disponible cuando el contenido se reajuste.

Contraste no textual

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

- **Componentes de la interfaz de usuario:** información visual necesaria para identificar los componentes y los estados, excepto en los componentes inactivos o cuando la apariencia la determina el agente de usuario y no la modifica el autor;
- **Objetos gráficos:** partes de los gráficos necesarias para comprender el contenido, excepto cuando una presentación concreta es esencial para la información que se transmite.

Los botones, campos de formulario, iconos no decorativos, gráficas (y sus diferentes partes) u otros componentes activos de la interfaz necesitan poder ser percibidos y diferenciados claramente por todas las personas en todos sus estados. Por ejemplo, un campo de formulario activo con un color gris claro, que no contrasta al menos 3:1 con el color del fondo, será difícil de percibir.



Ilustración 12 Formulario con insuficiente contraste (en primer lugar) y con suficiente contraste (en segundo lugar)

No tienen la obligación de pasar esta ratio de contraste: los campos deshabilitados, los logotipos, las banderas, los gradientes de colores que representan una medición (como los mapas de color), los dibujos naturalistas o las fotografías.

Observa las siguientes cuatro gráficas de tarta que representan la misma información. Los sectores sur y este tienen tonos muy parecidos y difíciles de diferenciar. Sin embargo, en las gráficas de la derecha, los sectores no están adyacentes sino separados por líneas, lo cual permite diferenciar claramente el límite de cada sector.

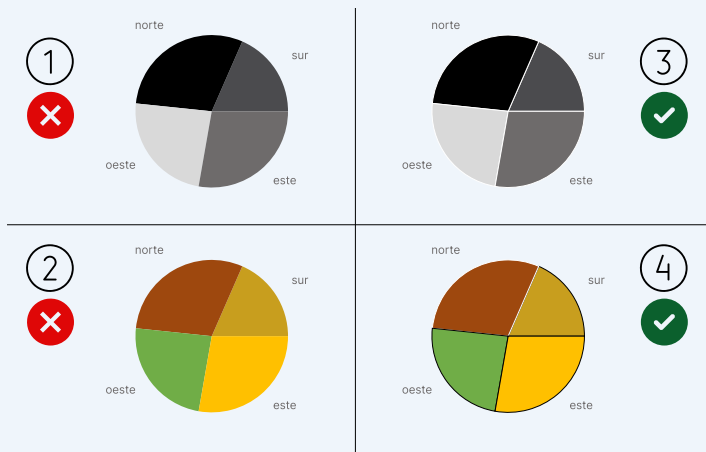


Ilustración 13 Gráficas sin suficiente contraste (izquierda) y con suficiente contraste (derecha).
Visión sin daltonismo

Observa cómo se verían las mismas gráficas con distintos tipos de daltonismo simuladas con el plug-in Stark para Figma.

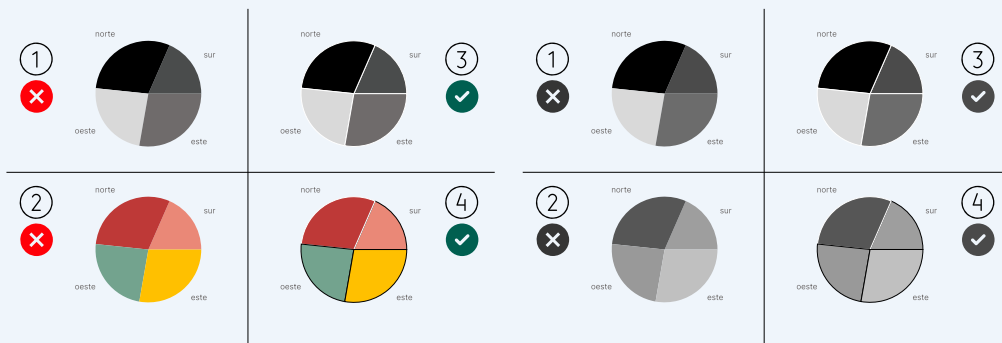


Ilustración 14 Daltonismo azul-amarillo - Tritanopia

Ilustración 15 Daltonismo completo - monocromacia o acromatopsia

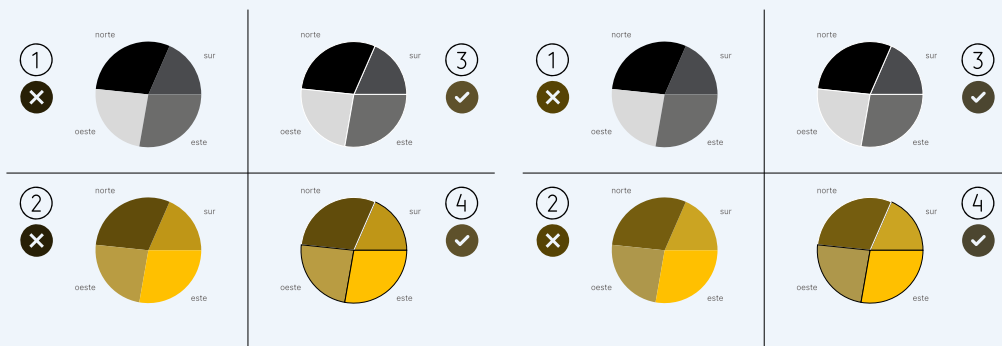


Ilustración 16 Daltonismo rojo-verde - protanopia

Ilustración 17 Daltonismo rojo-verde - deuteranopia

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. Observa cómo se percibe la misma gráfica según el tipo de visión.

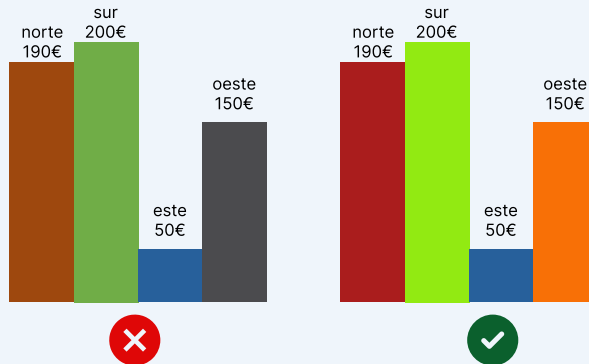


Ilustración 18 Gráfica de barras con barras sin suficiente contraste entre sí (en primer lugar) y con suficiente contraste entre sí (en segundo lugar). Visión sin daltonismo

Observa cómo se verían las mismas gráficas con distintos tipos de daltonismo.

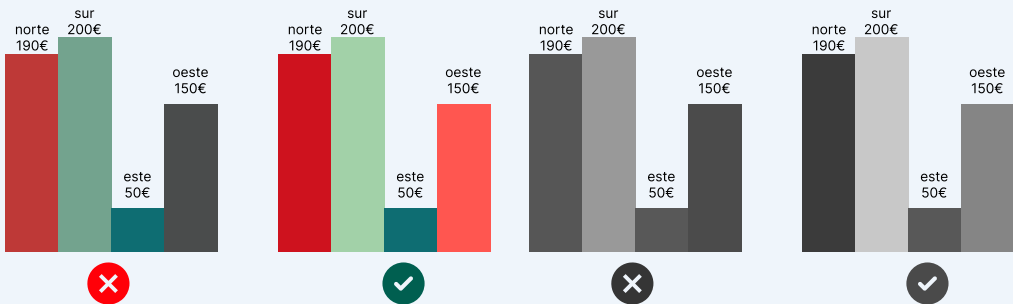


Ilustración 19 Daltonismo azul-amarillo - Tritanopia

Ilustración 20 Daltonismo completo - monocromacia o acromatopsia

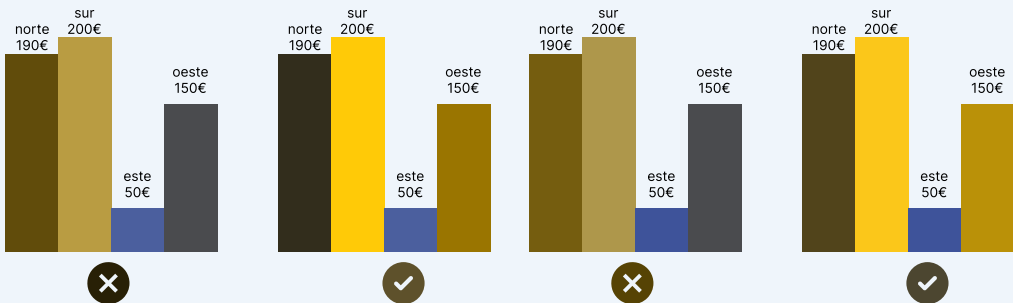


Ilustración 21 Daltonismo rojo-verde - protanopia

Ilustración 22 Daltonismo rojo-verde - deuteranopia

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

Técnicas para cumplir el criterio

El color se utiliza para identificar los componentes de la interfaz de usuario o para identificar los estados de los componentes de la interfaz de usuario:

- Utiliza un indicador del foco visible diseñado por el diseñador UX/UI.
- Ofrece un control con suficiente contraste que permita a las personas cambiar a una presentación que utilice suficiente contraste.

Se requiere el color para comprender el contenido gráfico:

- Asegúrate de que los iconos tienen una ratio de contraste de al menos 3:1.
- Proporciona suficiente contraste en los límites entre colores contiguos.

¡No lo hagas!

- No diseñes los contornos y los bordes de los elementos de una manera que elimine o haga invisible el indicador visual del foco.

Espaciado del texto

Si el lenguaje de marcado utilizado admite las propiedades de estilo que se listan a continuación, el texto se puede formatear de la siguiente manera sin perder contenido ni funcionalidad y 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.

Excepción:

- Los idiomas y sistemas de escritura humanos que no utilizan alguna de estas propiedades pueden alcanzar la conformidad usando sólo las propiedades que existan para esa combinación de idioma y sistema de escritura.

El objetivo de este criterio es garantizar que las personas que necesitan modificar la presentación del texto, como las personas con dislexia o con baja visión, podrán adaptar el texto a sus necesidades de forma que el contenido siga siendo legible y operable.

Este criterio no nos obliga a utilizar un espaciado concreto (como sí sucede en el criterio 1.4.8) ni a incluir una herramienta para personalizarlo, solo indica que, si la persona configura el espaciado de la manera que se especifica, hay que revisar que no se pierde contenido ni funcionalidad. Por otra parte, el desarrollador no debe impedir o interferir en la capacidad de personalización de estos estilos.

Hay idiomas o sistemas de escritura en los que algunas de estas métricas no son aplicables, por ejemplo, el japonés no utiliza espacios entre párrafos.

Este criterio solo aplica si el texto está implementado mediante lenguaje de marcado, por ello, aplica a las páginas HTML pero no a los documentos PDF. Tampoco aplica a las imágenes de texto ni a los subtítulos incrustados en el vídeo (en vez de incluidos en un documento separado).

En el capítulo [Herramientas de validación](#) encontrarás una herramienta que te ayudará a revisar este criterio.

Técnicas para cumplir el criterio

- Permite que el espaciado de tu texto se pueda anular o ajustar.

Técnicas para mejorar la usabilidad

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

¡No lo hagas!

- No permitas que el contenido quede recortado o superpuesto cuando se ajuste el espaciado del texto.

Contenido al pasar el cursor (*hover*) o al recibir el foco (*focus*)

Cuando un componente de la interfaz recibe y luego pierde el foco (de puntero o teclado), y esto genera que un contenido adicional se haga visible y luego se oculte, se cumplen estas tres condiciones:

- **Descartable:** hay un mecanismo para descartar el contenido adicional sin mover el puntero o el foco de teclado (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.
- **Desplazable:** si el puntero del cursor puede activar el contenido adicional, entonces el puntero se puede desplazar sobre él sin que desaparezca.
- **Persistente:** el nuevo contenido permanece visible hasta que se retira el puntero o el foco, la persona lo descarta o su información ya no es válida.

Excepción:

- 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 habituales de contenidos adicionales que se muestran cuando un componente recibe el puntero o el foco, son los submenús desplegables o los *tooltips* personalizados.

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. Cumpliendo este criterio nos aseguramos de que todas las personas pueden percibir el nuevo contenido y que lo pueden descartar sin problemas.

Técnicas para cumplir el criterio

- Haz que el contenido que se muestra al pasar el cursor o recibir el foco sea descartable, desplazable y persistente.
- Utiliza el *role=tooltip* de ARIA.
- Utiliza las pseudo-clases CSS *hover* y *focus*.

¡No lo hagas!

- No evites que el puntero del cursor se pueda desplazar sobre el contenido adicional sin que desaparezca.
- No evites que el contenido adicional se pueda descartar sin mover el puntero o el foco de teclado.
- No evites que el contenido adicional permanezca visible hasta que se descarte o no sea válido.



Principio 2.

Operable

Debemos ser conscientes de los diferentes periféricos y productos de apoyo que las personas utilizan para acceder a nuestros sitios web desde muy diversos dispositivos: ordenadores, tablets, teléfonos móviles, consolas, televisores, relojes, electrodomésticos, automóviles, kioscos interactivos, entornos de realidad aumentada y virtual ...

Al no tener el control del dispositivo que elegirá nuestra audiencia, debemos hacer los componentes de la interfaz de usuario y los elementos de navegación de tal forma que todas las personas puedan manejarlos.

Pauta 2.1

Manejable por teclado, no olvides a las personas 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 otros sistemas de entrada, como la voz o un ratón, y con una amplia variedad de productos de apoyo. Ninguna otra forma de entrada tiene esa flexibilidad o es universalmente compatible y operable por personas con distintas discapacidades.

Por el contrario, si sólo diseñas tu página web para ser usada con el ratón, habrá muchas personas que no podrán manejar tu página, como una persona ciega que usa un lector de pantalla, o una persona con limitaciones motoras que accede con un pulsador.

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

Teclado

Toda funcionalidad del contenido se puede manejar a través de una interfaz de teclado sin necesidad de alcanzar una determinada velocidad para cada pulsación de tecla.

Excepciones:

- Cuando la función requiera que la introducción de datos dependa del trayecto de los movimientos de la persona, y no sólo 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 no aplica sólo a los teclados físicos, sino también a los teclados que se muestran en pantalla, o a los teclados virtuales proyectados sobre superficies, o a cualquier otro que se invente en el futuro.

Técnicas para cumplir el criterio

- Garantiza que todas las funciones se pueden manejar sólo con el teclado.
- Garantiza el control del teclado mediante el uso de controles de formulario y enlaces HTML.
- Ofrece manejadores de eventos de teclado utilizando una de las siguientes opciones:
 - utiliza funciones de teclado y otras funciones específicas del dispositivo, como el uso de *mousedown* junto con *keydown*.
 - utiliza el evento *onclick* en enlaces y botones, ya que el evento *onclick* de estos elementos es independiente del dispositivo.
 - haz redundantes los eventos de ratón y teclado, por ejemplo, utiliza *mouseover* junto con *onfocus*.

Técnicas para mejorar la usabilidad

- Emplea los atributos de rol, estado y valor 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 los elementos HTML estáticos.

¡No lo hagas!

- No utilices sólo manejadores de eventos específicos de punteros como *onmousedown*.
- No quites el foco de un elemento por programación justo 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 utilizando el teclado, entonces el foco se puede quitar de ese componente utilizando sólo el teclado y, si se requiere algo más que las teclas de dirección o de tabulación, se informa a la persona del método apropiado para mover el foco.



Dado que cualquier contenido que no satisfaga este criterio puede interferir con la capacidad de las personas para utilizar toda la página, **todos los contenidos de la página web deben satisfacer este criterio.**

Técnicas para cumplir el criterio

- Comprueba que las personas usuarias de teclado no quedan atrapadas en el contenido.

¡No lo hagas!

- No combines varios formatos de contenido de tal forma que las personas puedan quedar atrapadas dentro de un tipo de formato, sin poder volver al resto del contenido únicamente con el teclado.

Teclado (sin excepciones)

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

Técnicas para cumplir el criterio

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 de la persona y no sólo de los puntos inicial y final, como un dibujo a mano alzada, entonces no será posible cumplir este criterio.

Atajos de teclado

Si se implementan atajos de teclado utilizando 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.

El objetivo de este criterio es evitar que las personas que acceden con un programa de reconocimiento de voz activen accidentalmente los atajos de teclado, lo cual es muy habitual cuando están asociados a un solo carácter. Además, las personas que acceden con el teclado y tienen 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 a componentes como listas o menús desplegables, ya que sus atajos únicamente están activos cuando los componentes tienen el foco.

Técnicas para cumplir el criterio

- Existe un mecanismo que permite a las personas desactivar o reasignar los atajos de teclado de un único carácter.

¡No lo hagas!

- No implementes atajos de teclado de un único 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 las personas el tiempo suficiente para leer y utilizar 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 las personas mayores y las 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:

- La persona puede eliminar el límite de tiempo antes de alcanzarlo.
- La persona 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 a la persona 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 las personas 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 debe considerarse junto con el [criterio de conformidad 3.2.1 A](#), que pone límites a los cambios inesperados en el contenido, o en el contexto, como resultado de una acción de la persona.

Técnicas para cumplir el criterio

Si hay un límite de tiempo de sesión:

- Ofrece una casilla de verificación en la primera página de un formulario que se distribuye en varias páginas que permita a las personas solicitar un tiempo de sesión mayor o que no haya límite de tiempo.
- Ofrece un mecanismo que permita a las personas anular el límite de tiempo

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

- Ofrece un mecanismo que permita a las personas anular el límite de tiempo.
- Ofrece un medio para aumentar el límite de tiempo 10 veces respecto al límite predefinido.
- Avisa a las personas cuando el límite de tiempo esté próximo a expirar; y permite que las personas puedan aumentar el límite predefinido.

Si hay un límite de tiempo para leer los contenidos:

- Permite a las personas pausar el contenido y volver a reproducirlo desde donde fue parado.
- Ofrece una manera para que las personas desactiven el límite de tiempo.
- Cuando el contenido se desplaza de forma automática mediante programación, ofrece un mecanismo para pausarlo.
- Ofrece un mecanismo que permita a las personas mostrar en una ventana separada, o en un área más grande de la página, y de forma estática, aquellos textos que se mueven, se desplazan o se actualizan solos.

¡No lo hagas!

- No refresques o redirecciones la página utilizando la etiqueta *meta http-equiv="refresh"* con 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, parpadea³⁴ o se desplaza, si comienza automáticamente, dura más de 5 segundos y se presenta en paralelo con otro contenido, existe un mecanismo para que la persona la pueda poner en pausa, detener u ocultar.

Excepciones:

- Cuando 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 la persona la pueda poner en pausa, detener u ocultar; o controlar la frecuencia de actualización

Excepciones:

- Cuando la actualización automática es una parte esencial de una actividad.



Dado que cualquier contenido que no satisfaga este criterio puede interferir con la capacidad de las personas para utilizar toda la página, **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 la persona pausa en un momento concreto. Cuando lo vuelva 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 la persona 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 a las personas pausar el contenido y volver a reproducirlo desde donde fue parado.
- Cuando el contenido se desplaza de forma automática mediante programación, se ofrece un mecanismo para pausarlo.
- Crea contenido que parpadee menos de 5 segundos.
- Utiliza tecnologías que permitan que las personas puedan 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.

³⁴ 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. El control puede estar en la parte superior de la página, o junto al contenido en movimiento.
- Ofrece un mecanismo para recargar la página sin contenidos parpadeantes. Eliminar el contenido que parpadea sólo es válido si la información del contenido parpadeante también se encuentra en la página recargada sin contenido parpadeante.

¡No lo hagas!

- No incluyas contenido que se desplaza sin incluir también un mecanismo para pausar y reiniciar el contenido, salvo que sea esencial para la actividad.
- 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 las personas puedan 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 las personas que puedan completar una actividad sin ningún límite de tiempo.

Interrupciones

Las personas pueden postergar o suprimir las interrupciones.

Excepción:

- las interrupciones implican una emergencia.

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

Técnicas para cumplir el criterio

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

¡No lo hagas!

- No refresques o redirecciones la página usando la etiqueta *meta http-equiv="refresh"* con un límite de tiempo.

Volver a autenticar

Cuando expira una sesión autenticada, las personas pueden 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 utilizando una de las siguientes técnicas:

- guarda los datos para que la persona pueda retomarlos tras volver a autenticarse; o
- codifica los datos introducidos por la persona como datos ocultos o cifrados en la página para volver a autenticarse.

¡No lo hagas!

No limites el tiempo de las sesiones sin ofrecer un mecanismo que guarde las entradas realizadas por la persona y que restablezca esa información cuando vuelva a autenticarse.

Límites de tiempo

Se avisa a las personas 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 la persona 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, la persona debería empezar de nuevo el proceso, con la consiguiente carga mental añadida.

Es importante avisar de forma correcta de los límites de tiempo, y de ser posible, añadir mecanismos para ajustar, extender o cancelar estos límites, como se ha explicado en el criterio de conformidad 2.2.1.

Las regulaciones de privacidad pueden requerir el consentimiento expreso de las personas 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 de la persona 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 pueda provocar mareos, vértigos, ataques, espasmos o convulsiones.

Las personas que tienen epilepsia fotosensitiva reaccionan ante cambios bruscos de luminosidad. La velocidad más alta permitida para mostrar contenido *destellante* 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 algunas personas, y, como vimos en el criterio 2.2.2, sólo 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 la persona pudiera evitarlos, o incluso puede haber personas que no saben que son sensibles a los destellos.

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 de las personas para utilizar toda la página, **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.

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, esto es, el área que destella debe ser menor del 25% de los 10 grados del campo visual (que representa el área central de visión en el ojo). Por ejemplo, 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.
- Utiliza una herramienta para asegurar que el contenido no viola el umbral del destello general o el umbral de destello rojo.



¿Cómo saber si el cambio en la luminosidad del contenido es más rápido de 3 veces por segundo o está por debajo del umbral indicado? En el capítulo [Herramientas de validación](#) encontrarás la referencia a la herramienta PEAT, que te ayudará a comprobar si tu contenido viola estos umbrales.

Tres destellos

Las páginas web no contienen nada que destelle más de tres veces por segundo.

Técnicas para cumplir el criterio

Comprueba que ningún componente del contenido destella más de tres veces por segundo.

Animaciones desde interacciones

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 la persona sube o baja la página. Otro ejemplo son las transiciones de fantasía entre páginas, o un zoom rápido en los elementos que cogen el foco.

El objetivo de este criterio es ayudar a las personas que tienen 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 utilizan con criterio, pueden ser una buena manera de comunicar la relación entre diferentes zonas, atraer la atención de las personas o simplemente añadir algo de diversión a la interacción.

Técnicas para cumplir el criterio

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

Pauta 2.4

Arquitectura de información fácilmente navegable

Proporciona medios para ayudar a las personas 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 las que acceden con lectores de pantalla o para las personas con discapacidad cognitiva.

Evitar bloques

Existe un mecanismo que permite a las personas evitar los bloques de contenido que se repiten en múltiples páginas web.

Técnicas para cumplir el criterio

Crea enlaces para saltar los bloques de contenidos repetidos con una de estas técnicas:

- Añade un enlace en el inicio de la página para ir directamente al contenido principal.
- Añade un enlace en el inicio de cada bloque repetido para ir al final de ese bloque.
- Añade enlaces en el inicio de la página para ir a cada área de contenido.

Agrupar los bloques de contenidos repetidos de tal manera que se puedan saltar con una de estas técnicas:

- Identifica las regiones de la página mediante *landmarks roles* de ARIA.
- Ofrece elementos de encabezado al inicio de cada sección de contenido.
- Utiliza elementos *iframe* y *frame* identificados con el atributo *title*.
- Utiliza menús expansibles y contraíbles.

Técnicas para mejorar la usabilidad

- Posiciona el contenido en base al marcado de estructura.
- Agrupa los enlaces relacionados utilizando el elemento `<nav>`.

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 mediante el elemento *title*.

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.

¡No lo hagas!

- Evita títulos de página que no identifiquen la página o sus contenidos, por ejemplo, con textos como "Documento sin título", "pagina.html" o un texto vacío.

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.

“Navegar secuencialmente” significa navegar avanzando con el foco de un elemento al siguiente en el orden definido, usando para ello una interfaz de teclado, por ejemplo, utilizando la tecla *Tab* (el tabulador) del teclado.

Técnicas para cumplir el criterio

- Coloca los elementos interactivos en un orden que siga las secuencias y las relaciones dentro del contenido.
- Haz que el orden del DOM coincida con el orden visual para que el foco recorra los elementos en un orden que siga las secuencias y las relaciones dentro del contenido.
- Si quieres que una página cambie dinámicamente, elige una de estas opciones:
 - Inserta contenido dinámico en el DOM inmediatamente después del elemento disparador.
 - Crea cuadros de diálogo personalizados que se puedan activar independientemente del dispositivo de entrada y colocados en el DOM después del elemento que los activó.
 - Reordena las secciones de la página con el DOM.

¡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.
- Evita utilizar cuadros de diálogo o menús desplegados que no estén adyacentes al botón que los activa cuando se navega secuencialmente.

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,

Excepción:

- Cuando el propósito del enlace resulte ambiguo para las personas en general.

Técnicas para cumplir el criterio

- Proporciona un texto de enlace que describa su propósito (si incluye una imagen, su texto alternativo forma parte del texto de enlace).
- Si utilizas un mapa de imagen, ofrece alternativas textuales para los elementos *area*.
- Permite a las personas elegir entre enlaces cortos o largos con un control al inicio de la página o por programación.
- Identifica el propósito del enlace utilizando el texto del enlace combinado con el texto de la oración que lo engloba.
- Proporciona una información complementaria sobre el propósito del enlace:
 - con el atributo *title* del enlace.
 - en el propio texto de enlace, ocultando parte del texto con CSS. 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 visualmente y/o para el lector de pantalla \(tabla resumen\)"](#).³⁵
- Identifica el propósito de un enlace utilizando el texto de enlace combinado con su contexto determinable por software, utiliza para ello:
 - el atributo *aria-labelledby* (ejemplo 1 en la tabla de la página siguiente); o
 - el atributo *aria-label*; o
 - el párrafo que contiene el enlace (ejemplo 2); o
 - el elemento de la lista que contiene el enlace (ejemplo 3); o
 - en una lista anidada, el elemento padre que contiene la lista en la que se encuentra el enlace (ejemplo 4); o
 - la celda de tabla que contiene el enlace más sus encabezados asociados.

En la Tabla 7 mostramos cuatro ejemplos de un texto de enlace combinado con su contexto de forma que pueda ser determinable por software:

³⁵ <https://olgacarreras.blogspot.com/2020/07/ocultar-contenido-de-una-pagina-web.html>

Tabla 7 Texto de enlace combinado con contexto determinable por software

Ejemplo	✓ Forma correcta	✗ Forma incorrecta
1	<p>Informe anual Descarga en PDF</p> <pre><h3 id="t1">Informe anual</h3> <p> Descarga en PDF</p></pre>	<p>Informe anual Descarga en PDF</p> <pre><h3>Informe anual</h3> <p>Descarga en PDF </p></pre>
2	<p>Informe anual en PDF</p> <pre><p>Informe anual en PDF</p></pre>	<p>Informe anual en PDF</p> <pre><p>Informe anual</p> <p>en PDF</p></pre>
3	<p>1. Informe anual (PDF) 2. ...</p> <pre> Informe anual (PDF) ... </pre>	<p>1. Informe anual 2. (PDF)</p> <pre> Informe anual (PDF) </pre>
4	<p>- Informe anual - (HTML) - (PDF) - ...</p> <pre> Informe anual (HTML) (PDF) ... </pre>	<p>Informe anual - (HTML) - (PDF)</p> <pre><p>Informe anual</p> (HTML) (PDF) </pre>

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.
- Identifica el propósito del enlace utilizando su propio texto y su encabezado anterior.

¡No lo hagas!

- No ofrezcas enlaces cuando el contexto necesario para comprenderlos no está relacionado con el enlace a nivel de código (con *aria-label* o *aria-labelledby*; o por estar en la misma oración, párrafo, elemento de lista o celda de tabla).
- No utilices una imagen sin un nombre accesible 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.

Excepción:

- Cuando la página es un paso intermedio o el resultado de un proceso.

Este criterio permite a las personas que encuentren el contenido de la manera que mejor se ajuste a sus necesidades.

Por ejemplo, hay personas que pueden preferir utilizar un buscador antes que un menú de navegación. Hay que tener en cuenta que solo será necesario que el buscador encuentre contenido dentro del **conjunto de páginas** en la que se encuentra. Un portal de comercio electrónico puede tener un buscador en las páginas de producto, que puede buscar sólo dentro de este conjunto de páginas, y un buscador dentro del centro de ayuda, que puede buscar sólo dentro del centro de ayuda.

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 (es eficaz en sitios web pequeños),
- una lista de enlaces a todas las páginas del sitio en todas las páginas del sitio (es eficaz en sitios web pequeños).

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 utilizado por las herramientas de navegación que disponen algunos navegadores.

Encabezados y etiquetas

Los encabezados y las etiquetas describen el tema o propósito.

Este criterio de conformidad no requiere que haya encabezados o etiquetas, tampoco que el contenido que actúa como encabezado o etiqueta esté correctamente marcado o identificado, puesto que estos aspectos se tratan en otros criterios, como los criterios de conformidad 1.3.1, 2.4.11, 3.3.2 o 4.1.2.

Este criterio de conformidad sólo indica que, en el caso de que se proporcionen encabezados y etiquetas, estos deben ser descriptivos. Ten en cuenta que “descriptivos” no significa que deban ser largos.

Técnicas para cumplir el criterio

- Ofrece encabezados y etiquetas que sean descriptivos.

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

- Utiliza los componentes de la interfaz que son resaltados por el agente de usuario cuando reciben el foco.
- Si quieres cambiar el diseño de los elementos cuando reciban el foco, hazlo con CSS o con programación.
- Utiliza el indicador del foco nativo del navegador porque así se podrá usar la configuración de alta visibilidad del foco de teclado que permiten las plataformas o ciertos productos de apoyo.
- Diseña un indicador de foco de sistema muy visible.
- Crea un indicador del foco de dos colores para garantizar un contraste suficiente en todos los componentes.

¡No lo hagas!

- No quites el foco de un elemento por programación justo cuando lo reciba.
- No ocultes el indicador visual del foco debido al estilo del borde o de la línea exterior del elemento. Por ejemplo, no lo ocultes con el estilo *outline: none* o *outline:0*.

Ubicación

Se informa de la ubicación de la persona 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 utilizado por las herramientas de navegación que disponen algunos navegadores.

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.

Excepción:

- Cuando el propósito del enlace deba resultar ambiguo para las personas en general.



Técnicas para cumplir el criterio

- Utiliza el atributo *aria-label*.
- Proporciona un texto de enlace que describa su propósito (si incluye una imagen, su texto alternativo forma parte del texto de enlace).
- Si usas un mapa de imagen, ofrece alternativas textuales para los elementos *area*.
- Permite a las personas elegir entre enlaces cortos o largos con un control al inicio de la página o por programación.
- Proporciona una información complementaria sobre el objetivo del enlace en el propio texto de enlace, ocultando parte del texto con CSS. Es muy importante que lo ocultes visualmente pero no para el lector de pantalla, por ejemplo con *text-indent:-1000px* pero no con *display:none*. Más información en el [artículo "Ocultar contenido visualmente y/o para el lector de pantalla"](#) de Olga Carreras.³⁶

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 de la tabla siguiente)
- Identifica el propósito del enlace usando su *title*. (Ejemplo 3)

Tabla 8 Identificar el propósito de los enlaces

Ejemplo	Resultado	Forma correcta
1	Informe anual 	<code>Informe anual </code>
2	 Inicio	<code>Inicio</code>
3	OK	<code>OK</code>

¡No lo hagas!

- No utilices enlaces como "Pulsa aquí" o "Leer más" sin un mecanismo para cambiar el texto del enlace a un texto más específico.
- No utilices una imagen sin un nombre accesible cuando la imagen sea el único contenido de un enlace.

³⁶ <https://olgacarreras.blogspot.com/2020/07/ocultar-contenido-de-una-pagina-web.html>

Encabezados de sección

El contenido está organizado mediante encabezados de sección.

Este criterio alude a estructurar las secciones de texto, no los componentes de la interfaz de usuario.

Técnicas para cumplir el criterio

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

Consejos sobre encabezados

1. Los encabezados deben **dividir** trozos grandes de texto.
2. Los encabezados deben estar situados justo **antes de su contenido** asociado.
3. Los encabezados deben **resumir** su contenido asociado.
4. Los encabezados deben **ser cortos** para permitir el escaneo rápido de la página.
5. Los encabezados deben ser **consistente** en todo el sitio web.
6. Los encabezados deben **estructurar** los contenidos de la página y no sólo ser usados para mostrar efectos visuales.
7. Todos los encabezados visuales deben tener un **encabezado de estructura** (*h1-h6*).
8. Las páginas deben tener **un único encabezado h1**, el cual es el título principal de la página y, al menos en las páginas interiores, debería estar ubicado en el inicio del contenido principal de la página.
9. No se deben **saltar niveles** de encabezados. 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. Se deben usar los elementos **legend y label** en los formularios, no los encabezados *h1-h6*.

Foco no oculto (mínimo)

Cuando un componente de la interfaz de usuario recibe el foco de teclado, el componente no puede estar completamente oculto por un contenido creado por el autor de la página:

- Si la persona puede reposicionar el contenido (mover las barras de herramientas, mover los cuadros de diálogo no modales...), sólo se considera su posición inicial.
- El contenido abierto por las personas puede ocultar el componente que recibe el foco. En estos casos, si la persona puede revelar el componente enfocado sin hacer avanzar el foco con el teclado, el componente enfocado no se considera oculto. Por ejemplo, pulsando la tecla ESC para descartar el contenido abierto; o utilizando las teclas de flecha para desplazar el contenido.

Los ejemplos más habituales de contenidos que se superponen a un elemento con el foco son las cabeceras y pies fijos o las capas no modales, como la capa de aceptación de *cookies*. Pero hay que tener en cuenta que, **si la superposición es completa, pero semiopaca**, estaría cumpliendo el criterio, porque no ocultaría por completo el foco.

Los menús desplegables, los calendarios de selección de fecha, los campos desplegables o los *tooltips* son componentes que tapan el contenido, pero no son persistentes, por lo tanto, no incumplen este criterio. Sin embargo, **si la implementación del componente permite que persistan** sí que pueden estar incumpliendo el criterio. Por ejemplo, si abres un menú desplegable y, cuando lo abandonas con el foco del teclado no se cierra y el foco de teclado acaba en un contenido oculto por ese menú desplegable no cerrado, se estaría incumpliendo el criterio.

Técnicas para cumplir el criterio

- Usa la propiedad CSS `scroll-padding` para que las personas puedan acceder a los componentes de la interfaz de usuario (por ejemplo: enlaces, botones y campos de formulario) que inicialmente están completamente ocultos por un componente de posición fija (por ejemplo, un banner de consentimiento de *cookies* fijo en la parte inferior de la página).

¡No lo hagas!

- No utilices cabeceras o pies de página fijos o adhesivos (conocidos como “componentes *sticky*”) de tal manera que puedan ocultar completamente elementos enfocables (como botones, enlaces, campos de formulario, etc.)

Foco no oculto (mejorado)

Cuando un componente de la interfaz de usuario recibe el foco de teclado, ninguna parte del componente puede estar completamente oculta por un contenido creado por el autor de la página.

Este criterio es similar al criterio anterior, pero más estricto. El criterio de conformidad 2.4.11 permite que alguna parte del componente que recibe el foco pueda estar oculta por un contenido creado por el autor (ya sea el diseñador UX/UI, el desarrollador o el creador de contenido), mientras que **el 2.4.12 prohíbe que alguna parte del componente que recibe el foco pueda quedar oculta** por un contenido creado por el autor.

Técnicas para cumplir el criterio

Simplemente sigue las mismas técnicas y consejos del criterio de conformidad 2.4.11 para cumplir este criterio.

Apariencia del foco

Cuando el indicador del foco de teclado es visible, el área del indicador del foco cumple lo siguiente:

- es al menos tan grande como el área de un perímetro de 2 píxeles CSS de grosor del componente o subcomponente sin el foco; y
- tiene una ratio de contraste de al menos 3:1 entre los mismos píxeles en el estado con el foco y sin el foco.

Excepciones:

- El indicador del foco lo determina el agente de usuario y el autor no puede ajustarlo.
- El autor no ha modificado ni el indicador del foco ni el color del fondo del indicador.

Este criterio complementa:

- al criterio 2.4.7 (Foco visible) que exige que el foco sea visible, de modo que ahora el criterio 2.4.13 exige además un nivel mínimo de visibilidad basado en el tamaño y el contraste; y
- al criterio 1.4.11 (Contraste no textual) que exige un contraste adecuado del indicador del foco con el fondo, de modo que ahora el criterio 2.4.13 añade que haya un contraste suficiente entre los dos estados del componente, con y sin el foco.

El criterio hace referencia específica al borde, no a efectos como sombras o brillos.

En la explicación del criterio encontrarás múltiples casos prácticos. Por ejemplo, si queremos que el indicador del foco muestre un borde de dos píxeles negros (con suficiente contraste con el fondo) alrededor de un botón, podemos elegir cuatro técnicas, y no todas cumplirían el criterio:

- Los tres primeros botones con foco ("*outset*", "*outline*" y "*border*") sí cumplirían el criterio ya que son igual o más grandes que el elemento de interfaz;
- Sin embargo, el cuarto botón con foco ("*inset*") no cumpliría, y necesitaría que el indicador fuera de al menos 3px.

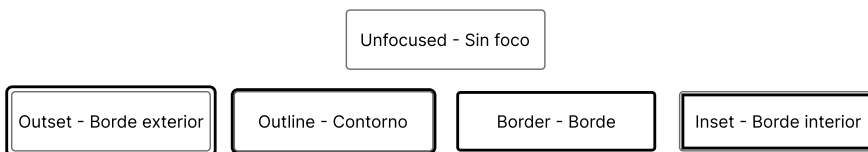


Ilustración 23. Ejemplos de botón sin y con foco

Técnicas para cumplir el criterio

- Diseña un indicador del foco de sistema muy visible.
- Crea un indicador del foco de dos colores para garantizar un contraste suficiente en todos los componentes.
- Crea un indicador del foco sólido dentro del componente.

¡No lo hagas!

- No utilices un borde CSS en los textos en línea que se puedan reajustar.

Pauta 2.5

Formas de introducir la información

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

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

Gestos del puntero

Todas las funcionalidades que utilicen 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.

Excepción:

- El uso de varios puntos o el recorrido es totalmente esencial.

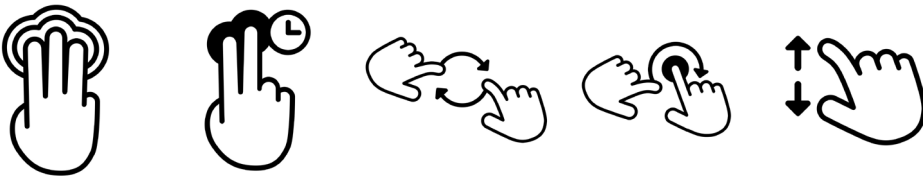


Ilustración 24 Diferentes gestos de puntero con 2 o 3 dedos

En estas imágenes 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 único punto son: toque (clic), doble toque (doble clic) y pulsaciones largas.

Hay que tener en cuenta que el criterio no aplica a las acciones que se requieren para operar el agente de usuario o el producto de apoyo, como deslizar para escrolar, separar dos dedos para hacer zoom o girar los dedos para activar el rotor de VoiceOver. Sin embargo, sí aplicaría a acciones de deslizar implementadas en la página, como el gesto de deslizar un carrusel.

Técnicas para cumplir el criterio

- Proporciona controles para lograr el mismo resultado que los gestos multipunto o basados en rutas. Por ejemplo, botones anterior y siguiente alternativos al gesto de deslizar.
- Permite que los controles deslizantes se puedan operar con un único toque.

¡No lo hagas!

- No proporciones una funcionalidad basada en un recorrido gestual sin una alternativa que se pueda operar con un único punto.

Cancelación del puntero

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 utiliza 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 táctil, hay al menos dos movimientos, uno de pulsar hacia “abajo” (*onmousedown*, *onkeydown*, *onkeypress*, *ontouchstart*) y otro de soltar hacia “arriba” (*onmouseup*, *onkeyup*, *ontouchend*). Por ejemplo, en un botón “Enviar” se espera que la acción se dispare en el evento “arriba”.

Las personas con dificultad para manejar el puntero, con baja visión o con limitaciones cognitivas necesitan la seguridad de que, cuando manejan el contenido, un toque “abajo” por error puede ser corregido. En la interacción por defecto de enlaces y botones con un simple clic, la cancelación del evento viene incorporada por defecto, si la persona descubre que ha seleccionado el elemento incorrecto, puede cancelar la acción alejando el puntero o el dedo del objetivo antes de soltarlo.

Un ejemplo de excepción sería un teclado mostrado en pantalla. Estamos habituados a que cuando se pulsa la tecla se muestre en pantalla el carácter elegido, es decir, que la funcionalidad se dispare en el evento “abajo” y, por tanto, sería una característica esencial de la funcionalidad. Ocurriría lo mismo en una aplicación de piano, sería esencial que el sonido sonara en el evento abajo.

Hay que tener en cuenta además que el criterio no aplica a las acciones que se requieren para operar el agente de usuario o el producto de apoyo.

Técnicas para cumplir el criterio

- Garantiza que las acciones de arrastrar y soltar (*drag&drop*) se pueden cancelar.
- Usa controles nativos para garantizar que la funcionalidad se activa en el evento “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 invierte el resultado.

Etiqueta en el nombre

Para los componentes de la 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 asociado. 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 las propiedades *aria-label* y *aria-labelledby* del estándar ARIA podemos modificar estos nombres accesibles. Puedes ampliar información en el [capítulo ARIA, el aliado del HTML accesible](#).

En el siguiente ejemplo, el nombre accesible del enlace “Leer más” pasaría a ser “Entra una nueva borrasca en la península”:

```
<h2 id="titNot1">Entra una nueva borrasca en la península</h2>
<p>Entradilla noticia 2</h2>
<p><a href="..." aria-labelledby="titNot1">Leer más...</a></p>
```

Los nombres accesibles permiten a las personas usuarias de productos de apoyo diferenciar los componentes, navegar por ellos y finalmente manejarlos. Puede parecer que el ejemplo anterior es una buena opción, porque el lector de pantalla me leerá un texto de enlace más claro, cumpliendo así con el criterio 2.4.4.

Sin embargo, **las personas que accedan por voz no podrán interactuar** con ese enlace. Estas personas dirán “Leer más” para activar el enlace (si hubiera varios enlaces con el mismo nombre quedarían resaltados con un número, para que la persona indicara cuál quiere activar). Pero el enlace ya no responde al nombre “Leer más”, sino al nombre “Entra una nueva borrasca en la península”, por tanto, no se activará.

Lo correcto es:

```
<h2 id="titNot1">Entra una nueva borrasca en la península</h2>
<p>Entradilla noticia 2</h2>
<a href="..." id="en1" aria-labelledby="en1 titNot1">Leer más...</a>
```

Con este código:

- el lector de pantalla leerá un enlace descriptivo: “Leer más... Entra una nueva borrasca en la península”, comprensible también fuera de contexto;
- las personas usuarias de un lector de pantalla, pero que ven la pantalla, tendrán una experiencia mejor si lo que anuncia el lector es igual que la etiqueta que ven, o al menos la contiene; y
- el enlace ya será accesible por voz mediante su etiqueta visible “Leer más”.



Puedes leer más sobre [cómo los navegadores calculan este nombre accesible](#)³⁷ por cada elemento y [cómo mapear los nombres de cada elemento con aria-label, aria-labelledby, y aria-describedby](#)³⁸.

Técnicas para cumplir el criterio

- Incluye el texto de la etiqueta visible como parte del nombre accesible.
- Haz que el nombre accesible coincida con la etiqueta visible.

Técnicas para mejorar la usabilidad

- Ubica las etiquetas de tal manera que ayude a predecir las relaciones.
- Si un icono no tiene un texto que lo acompañe, considera utilizar el texto flotante como nombre accesible.

¡No lo hagas!

Evita nombres accesibles que:

- no contengan el texto visible,
- contengan el texto visible pero las palabras estén en un orden diferente,
- contengan el texto visible pero una o más palabras se entremezclan con las palabras de la etiqueta visible.

³⁷ <https://www.w3.org/TR/accname-1.1/>

³⁸ <https://www.w3.org/TR/core-aam-1.1/>

Actuación por movimiento

La funcionalidad que se puede manejar con el movimiento del dispositivo o de la persona usuaria puede ser operada por componentes de la interfaz de usuario, y la respuesta al movimiento puede ser deshabilitada para prevenir una activación accidental.

Excepciones:

- El movimiento se utiliza 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 un movimiento de la persona (por ejemplo, mover la mano delante de la *webcam*), pueden ser activadas también 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 (como los 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

- Para toda entrada activada por movimiento, proporciona controles convencionales que realicen la misma función, y una configuración de preferencias que permita a las personas desactivar la actuación por movimiento.
- Asegura la compatibilidad con las características del sistema que permiten a las personas desactivar la actuación por movimiento.

¡No lo hagas!

- No implementes una actuación por movimiento sin la posibilidad de desactivarla.
- No deshabilites o entorpezcas de algún modo la posibilidad de las personas de desactivar la actuación por movimiento a nivel del sistema.

Tamaño del área de interacción (Mejorado)

El tamaño de las áreas interactivas con un puntero es al menos 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 desarrollador.
- 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 tienen temblores, baja visión, 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.

Los enlaces que están incluidos dentro de una frase o bloque de texto son una excepción, pero no lo son si la frase o el bloque de texto son en su conjunto un enlace.

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.
- Garantiza que los enlaces en línea tienen un área de interacción suficientemente grande.

¡No lo hagas!

- No diseñes áreas interactivas menores de 44 por 44 píxeles.

Mecanismos de entrada concurrentes

El contenido web no restringe el uso de las diferentes formas de introducir información.

Excepciones:

- Cuando la restricción es esencial.
- Cuando la restricción es necesaria para garantizar la seguridad del contenido.
- Cuando la restricción es necesaria para respetar la configuración de la persona usuaria.

No debemos asumir que las personas siempre utilizan 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 a las personas cambiar de una modalidad a otra, a su voluntad.

Técnicas para cumplir el criterio

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

¡No lo hagas!

- No limites las interacciones en los dispositivos con pantalla táctil, asumiendo que no se añadirán o utilizarán otros mecanismos de entrada, como un ratón o un teclado.

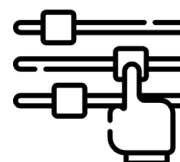
Movimientos de arrastre

Toda funcionalidad que opere mediante un movimiento de arrastre se tiene que poder realizar con un puntero sencillo sin arrastrar.

Excepción:

- El movimiento de arrastre es esencial.
- La funcionalidad está determinada por el agente de usuario y no es modificada por el desarrollador.

Hay que tener en cuenta que algunas personas no pueden realizar movimientos de arrastre de forma precisa. Otras utilizan un dispositivo de entrada, como un puntero de cabeza, control por voz o de seguimiento ocular, que hace que el arrastre sea complicado, propenso al error o totalmente imposible.



No es lo mismo "deslizar" que "arrastrar". Hablamos de "arrastrar" cuando, en el evento de bajada, un elemento (o una representación de su posición) sigue al puntero hasta el evento de subida. Que necesites gestos simples para "deslizar", por ejemplo, un carrusel en una pantalla táctil, ya está cubierto por el criterio "2.5.1 Gestos del puntero".

También hay que aclarar que un componente con *drag & drop* ya tiene que ser actualmente accesible por teclado gracias al criterio "2.1.1 Teclado", pero esto no implica que esa solución funcione mediante toques simples en el acceso táctil, eso lo cubre este nuevo criterio.

Técnicas para cumplir el criterio

- Garantiza que haya una alternativa disponible para los movimientos de arrastre que operan en el contenido que no requiera arrastrar.

¡No lo hagas!

- No olvides proporcionar un método de puntero único para que las personas operen una función que requiera un movimiento de arrastre.

Tamaño del área de interacción (Mínimo)

El tamaño de las áreas interactivas con un puntero es al menos 24 por 24 píxeles CSS.

Excepciones:

- **Espaciado:** las zonas de interacción que tienen un tamaño menor de 24 por 24 píxeles CSS se colocan de tal modo que, si se dibuja un círculo de 24 píxeles CSS de diámetro centrado en el cuadro delimitador de cada una, los círculos no intersecan con otro objeto o con el círculo de otra zona de interacción de un tamaño menor de 24 por 24 píxeles CSS.
- **Equivalente:** la función se puede lograr a través de un control diferente en la misma página que cumpla con este criterio.
- **En línea:** el área de interacción está dentro de una oración o su tamaño está restringido por la altura de línea del texto que no forma parte del área de interacción. La altura de la línea debe interpretarse como perpendicular al flujo de texto. Por ejemplo, en un idioma que se escribe verticalmente, la altura de la línea sería horizontal.
- **Control del agente de usuario:** el tamaño del área de interacción lo determina el agente de usuario y el autor no lo modifica. Por ejemplo, podrían ser los días del calendario en un `input type="date"`.
- **Esencial:** una presentación particular del área de interacción es esencial o es legalmente requerida para la información que se transmite. Por ejemplo, en los mapas, los pines que indican los lugares pueden presentarse muy pequeños a medida que se reduce el zoom, pero es esencial para mostrarlos en la ubicación correcta del mapa.

Este criterio es la variante menos estricta del criterio "2.5.5 Tamaño del área de interacción (mejorado)" de nivel AAA, donde se define el tamaño mínimo como 44 por 44 píxeles CSS, una dimensión que se debe procurar alcanzar siempre que sea factible.

El propósito de ambos criterios es garantizar que los elementos interactivos sean fáciles de activar sin riesgo de pulsar involuntariamente uno cercano. Cuando la región interactiva de un objeto es pequeña, las personas con temblores en las manos o dificultades en la destreza motora fina enfrentan dificultades para activarlos con precisión. Al proporcionar un tamaño adecuado, o un espacio suficiente entre las áreas de interacción, disminuye la probabilidad de activar accidentalmente el control incorrecto.

Observa el ejemplo de la Ilustración 25. Muestra tres botones de 16 píxeles de alto con un margen inferior y superior de 20 píxeles. El área de interacción (un círculo de 24 píxeles) puede actuar sobre los botones sin interferir en otros componentes.



Ilustración 25 Botones con un área de interacción de un tamaño suficiente

Sin embargo, como se ve en la Ilustración 26, cuando visualizas la misma página en una pantalla más pequeña, los botones se apilan uno encima de otro:

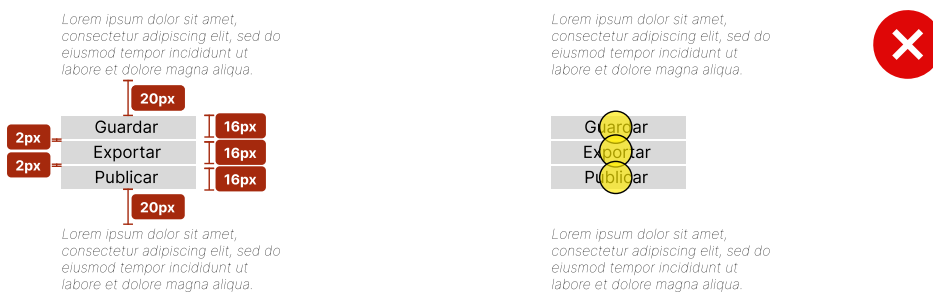


Ilustración 26 Botones con un área de interacción de tamaño insuficiente

En este caso, el área de interacción de los elementos se solapa y no hay suficiente espacio para pulsar en ellos sin cometer un error, por lo tanto, no sería suficiente para cumplir el criterio.

Es importante tener en cuenta que los enlaces marcados como lista en una estructura de navegación, como la que se muestra en la Ilustración 27, no cuentan como enlaces en línea y, por tanto, no son una excepción. Los diseñadores UX/UI pueden anticipar la posición relativa de estos enlaces y proporcionar suficiente espacio para la zona de interacción.

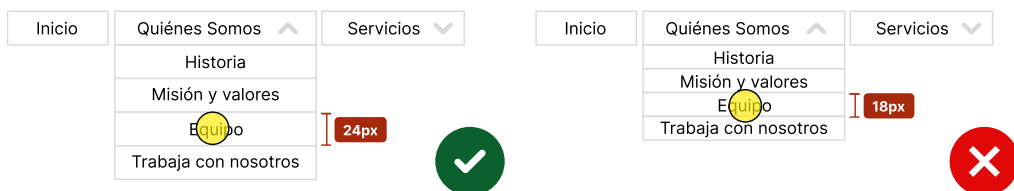


Ilustración 27 Menú desplegable con enlaces

Técnicas para cumplir el criterio

- Utiliza los atributos de altura mínima (`min-height`) y ancho mínimo (`min-width`) para garantizar un espacio suficiente entre las zonas interactivas.



Principio 3.

Comprensible

Si las personas que acceden a nuestro sitio web 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 preparen los textos de tal forma que cualquiera pueda comprenderlos.

Como complemento a esta pauta hemos dedicado un capítulo a "[Cuidar la escritura](#)".

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 utilizando los atributos de idioma.

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

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

En XHTML 1 debemos utilizar el atributo *lang* y 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.

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

Técnicas para cumplir el criterio

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



Puedes leer más sobre [internacionalización en la web del W3C](https://www.w3.org/International/questions/qa-html-language-declarations)³⁹.

³⁹ <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 utilizadas de modo inusual o restringido, incluidas las frases hechas 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 mediante una de estas técnicas:
 - enlaza a las definiciones; puedes utilizar listas de definición *dl*;
 - añade la definición en línea; en HTML puedes utilizar el elemento *dfn*.
- Defínela cada vez que aparezca mediante una de estas técnicas:
 - enlaza a las definiciones; puedes utilizar listas de definición *dl*;
 - proporciona un glosario;
 - proporciona una funcionalidad de búsqueda en un diccionario *online*.

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

- Enlaza a las definiciones; puedes utilizar listas de definición *dl*.
- Añade la definición en línea; en HTML puedes utilizar el elemento *dfn*.

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 de la página web:

- Proporciona la expansión o la explicación la primera vez que aparezca:
 - ofreciendo la forma expandida inmediatamente antes o después, por ejemplo, WAI (*Web Accessibility Initiative*); o
 - enlazando a su definición; o
 - utilizando el elemento *abbr*.
- Proporciona la expansión o la explicación cada vez que aparezca:
 - enlazando a su definición; o
 - utilizando el elemento *abbr*; u
 - ofreciendo un glosario; o
 - proporcionando una funcionalidad de búsqueda en un diccionario *online*.

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

- Proporciona la expansión o la explicación cada vez que aparezca:
 - enlazando a su definición, o
 - utilizando el elemento *abbr*.

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 complementario 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 UNESCO](#)⁴⁰).

Según la clasificación de la educación en España, se corresponde con un nivel de educación entre 1º y 3º de la [Educación Secundaria Obligatoria \(ESO\)](#)⁴¹.

En el capítulo [Herramientas de validación](#) encontrarás herramientas que te ayudarán 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 que 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 signos para la información, las ideas y los procesos que deben ser comprendidos para poder usar el contenido.



Puedes encontrar recomendaciones para elaborar documentos fáciles de leer en el capítulo [Cuidar la escritura](#) de este libro.

⁴⁰ <https://uis.unesco.org/en/topic/international-standard-classification-education-isc>

⁴¹ http://www.ine.es/daco/daco42/clasificaciones/cned14/CNED2014_capitulo0.pdf

Pronunciación

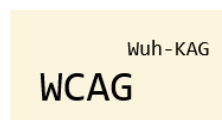
Cuando el significado de una palabra resulte ambiguo si no se conoce su pronunciación dentro del contexto, se proporciona un mecanismo para indicar su pronunciación.

En español, si no se cometen errores ortográficos, se cumple directamente este criterio.

Técnicas para cumplir el criterio

- Ofrece la pronunciación inmediatamente después de la palabra.
- Enlaza a las pronunciaciones.
- Ofrece un glosario que incluya la pronunciación de las palabras cuyo significado depende de la pronunciación.
- Ofrece la pronunciación utilizando [el elemento ruby de HTML](#)⁴², el cual añade pequeños textos de ayuda junto a un texto base y sirve para indicar la pronunciación o para hacer pequeños comentarios, como en el siguiente ejemplo. No confundas el elemento *ruby* de HTML con el lenguaje de programación del mismo nombre.

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



- Ofrece la pronunciación utilizando marcas diacríticas estándares, pero que puedan ser desactivadas. En algunos idiomas, la traslación escrita del lenguaje oral es un reto y se puede presentar de diversas maneras, como el 'pinyin' para el chino o los [marcados diacríticos 'okina' y 'kahako' para el hawaiano](#)⁴³. 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 a las personas de no mostrarlas si su ordenador no da soporte a la fuente utilizada.

⁴² <http://www.w3.org/TR/ruby>

⁴³ <https://www.hawaii.edu/site/info/diacritics.php>

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 la persona lo controle, puede confundirle, tanto si tiene como si no tiene alguna discapacidad.

Sé coherente y consistente al diseñar y dar nombre a los elementos de la interfaz, y permite acceder a la información de forma fácil.

Al recibir el foco

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

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

- **Cambiar de aplicación:** 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 rellenar campos de tipo cuenta bancaria o de tipo fecha);
- ir a otra página (incluyendo hacer creer a la persona que ha ido a otra página); o
- reorganizar el contenido de una página de forma significativa.

Técnicas para cumplir el criterio

- Utiliza 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

- Abre ventanas y pestañas nuevas desde un enlace sólo cuando sea necesario.
- Avisa a las personas de que se va a abrir una nueva ventana.

¡No lo hagas!

- No quites el foco de un elemento por programación justo cuando lo reciba.

Al recibir entradas

Cambiar la configuración de cualquier componente de la interfaz de usuario no provoca automáticamente un cambio de contexto a menos que la persona haya sido advertida de ese comportamiento antes de utilizar el componente.

Técnicas para cumplir el criterio

- Para iniciar un cambio de contexto, ofrece un botón que la persona pueda pulsar. Por ejemplo, un botón de envío, o un elemento de selección acompañado de un botón.
- Si un cambio en un control de formulario causa un cambio de contexto, describe antes qué sucederá.
- Utiliza el evento *onchange* en el elemento *select* sin provocar un cambio de contexto.

Técnicas para mejorar la usabilidad

- Avisa a las personas de 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 sin previo aviso una ventana nueva 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 la propia persona que usa el sitio.

El objetivo de este criterio es que se pueda predecir la ubicación de los mecanismos de navegación (menús, búsqueda, migas de pan, paginación, etc.), lo cual es muy útil, por ejemplo, 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, en algunas páginas podemos incluir un bloque de navegación secundaria, solo debemos comprobar que se mantiene el orden relativo del resto de los elementos.

Por otra parte, este criterio se refiere a los mecanismos que se repiten “dentro de un **conjunto de páginas**”, es decir, una colección de páginas que comparten un propósito común y son creadas por el mismo autor, grupo u organización.

Por ejemplo, las páginas del proceso de pago de un comercio electrónico son funcional y visualmente diferentes del resto de páginas del sitio y, por tanto, se las considera un conjunto de páginas distinto. Este conjunto de páginas puede tener mecanismos de navegación diferentes, como incluir un hilo de pasos o eliminar el menú principal para evitar el abandono del proceso de compra. Será necesario que, dentro de este conjunto de páginas, los mecanismos de navegación se presenten en el mismo orden relativo, pero este orden puede ser diferente al de otro conjunto de páginas del portal.

Técnicas para cumplir el criterio

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

¡No lo hagas!

No cambies el orden de los enlaces de navegación en las diferentes páginas.

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 utilizas un icono de flecha con el texto alternativo “Descargar” más el nombre del fichero que descarga, utiliza siempre la misma fórmula. O si un icono significa una cosa en una página, no debería significar algo diferente en otra página.

En el criterio se habla de los componentes “dentro de un **conjunto de páginas**”, es decir, dentro de una colección de páginas que comparten un propósito común y son creadas por la misma persona, grupo u organización.

Por ejemplo, dentro de un portal web puede haber un blog, un conjunto de páginas que son visualmente diferentes y publicadas por un equipo de personas distinto. La obligación de identificación consistente aplica dentro de este conjunto de páginas, pero no tiene por qué ser consistente con la del resto del portal.

Técnicas para cumplir el criterio

- Usa etiquetas, nombres y alternativas textuales de forma consistente (lo cual no significa que tengan textos idénticos) para contenidos que tengan la misma funcionalidad. Para incluir las etiquetas, nombres y alternativas textuales sigue las técnicas enumeradas en los criterios de conformidad “1.1.1 Contenido no textual” y “4.1.2 Nombre, función y valor”.

¡No lo hagas!

- No utilices dos etiquetas diferentes para la misma funcionalidad en diferentes páginas de un mismo conjunto de páginas del sitio.

Cambios a petición

Los cambios de contexto son iniciados únicamente a solicitud de las personas 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 a las personas que lo hagan en lugar de hacerlo automáticamente.
- Si la página web **puede redirigir automáticamente**, hazlo en el lado de servidor en lugar de en el lado de cliente; o hazlo en el lado de cliente con una etiqueta *meta-refresh* con tiempo nulo o "0".
- Utiliza 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 a la persona en el texto del enlace; o
 - utiliza un *script* respetando el principio de mejora progresiva, y avisa de que el contenido se abrirá en una ventana nueva.

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 este código:

```
//capturamos el evento
window.onload = addHandlers;

function addHandlers() {
// capturamos el enlace
  var objAnchor = document.getElementById('nuevaVentana');
  if (objAnchor){

// añadimos el aviso de apertura en nueva ventana
  objAnchor.firstChild.data = objAnchor.firstChild.data + ` (abre en
nueva ventana)`;

// lo lanzamos tanto en el 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, 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

- Abre ventanas y pestañas nuevas desde un enlace únicamente cuando sea necesario.

¡No lo hagas!

- No abras ventanas que no hayan sido solicitadas por las personas.
- No abras ventanas que se muestran tan pronto como el contenido se carga.
- No abras ventanas que se muestran cuando las personas introducen texto en un campo de entrada.
- No cambies el contenido principal mediante actualizaciones automáticas que las personas no puedan deshabilitar con un mecanismo dentro del contenido.
- No provoques un cambio de contexto cuando las personas retiran el foco de un elemento de formulario.
- No refresques o redirecciones la página utilizando etiquetas como *meta http-equiv="refresh"* con un límite de tiempo.

Ayuda consistente

Si una página web contiene cualquiera de los siguientes mecanismos de ayuda, y esos mecanismos se repiten en varias páginas web dentro de un conjunto de páginas web, estos mecanismos están en el mismo orden relativo respecto al contenido de otras páginas, a menos que la persona inicie un cambio:

- Datos de contacto humano (teléfono, email, horario de atención)
- Mecanismos de contacto humano (chat, formulario de contacto, redes sociales)
- Opción de autoayuda (página de preguntas frecuentes)
- Un mecanismo de contacto totalmente automatizado (un *chatbot*)

Los mecanismos de ayuda pueden proporcionarse directamente en la página o pueden proporcionarse a través de un enlace directo a una página diferente que contenga la información.

El objetivo del criterio no es exigir que haya opciones de ayuda, sino garantizar que, si las hay, las personas puedan encontrarlas en una ubicación consistente en todas las páginas.

Esta ubicación se refiere al **orden del contenido cuando se serializa el DOM** de la página, aunque se recomienda que la **posición visual** de este mecanismo de ayuda también sea consistente en todo el sitio web. Si el elemento de ayuda está visualmente en una ubicación diferente, pero en el mismo orden relativo, no incumplirá este criterio, pero será menos útil, menos usable, empeorando así la experiencia de uso.

La ubicación puede cambiar en la versión responsive. Este criterio se refiere al orden relativo de los elementos de contacto y ayuda entre las páginas mostradas en la misma variación de página (por ejemplo, el mismo nivel de zoom y la misma orientación).

Por otra parte, hay sitios web que constan de múltiples conjuntos diferentes de páginas con diferentes propósitos. Este criterio de conformidad permite que los **diferentes conjuntos de páginas web utilicen diferentes ubicaciones de mecanismos de ayuda**. Sin embargo, es mejor si los mecanismos de ayuda están ubicados de la manera más consistente posible incluso entre diferentes conjuntos de páginas web relacionadas.

Técnicas para cumplir el criterio

- Proporciona un enlace de “Contacto” en una ubicación consistente en todas las páginas.

¡No lo hagas!

- No incluyas la ayuda en una ubicación inconsistente.

Pauta 3.3

Ayuda a introducir datos. *Errar es humano*

Ayuda a las personas a evitar y corregir los errores.

Todos cometemos errores, y no iba a ser diferente en internet. Por ello, debemos prevenir que las personas 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 diseñes.

Identificación de errores

Si se detecta automáticamente un error en la entrada de datos:

- se identifica el elemento erróneo, y
- se describe el error a la persona mediante un texto.

Técnicas para cumplir el criterio

Si la persona no ha rellenado los campos obligatorios de un formulario:

- Proporciona una descripción textual que identifique los campos obligatorios que no se han completado.
- Utiliza el atributo *aria-invalid* para indicar los campos que han dado error.
- Valida el formulario en el lado de cliente y muestra un *alert()*.

Si la información proporcionada por la persona debe seguir un formato concreto o debe estar en un rango de valores determinado:

- Utiliza ARIA (*live regions*, *role="alertdialog"*, *role="alert"*) para notificar a los productos de apoyo, como un lector de pantalla, los errores en la entrada de datos.
- Utiliza el atributo *aria-invalid* para indicar los campos que han dado error.
- Cuando la persona incluye un valor que no está dentro de la lista de valores permitidos, avísale con un mensaje de texto explicando este hecho. Si es posible, debe incluir la lista de valores o sugerir el valor permitido que sea más similar al valor ingresado.
- Cuando la persona incluye datos que no cumplen con el formato o valores requeridos, avísale con un mensaje de texto explicando este hecho. Debes proporcionar:
 - ejemplos correctos, o
 - describir la entrada correcta, o
 - mostrar valores correctos similares a los introducidos por la persona, con instrucciones sobre cómo ingresar uno de estos valores correctos si la persona decide hacerlo.
- Valida el formulario en el lado del cliente y añade un mensaje de error en el DOM o con un *alert()*.

Técnicas para mejorar la usabilidad

- Ofrece un mecanismo para que las personas puedan saltar del mensaje de error al campo y viceversa.
- Avisa a la persona tanto si el formulario ha sido enviado con éxito como si no.

Etiquetas o instrucciones

Se proporcionan etiquetas o instrucciones cuando el contenido requiere la introducción de datos por parte de las personas.

Técnicas para cumplir el criterio

- Ofrece etiquetas descriptivas y una de las siguientes opciones:
 - utiliza *aria-describedby* para describir los controles de la interfaz;
 - utiliza *aria-labelledby* para definir la etiqueta concatenando textos de varios nodos;
 - utiliza roles de ARIA (*role="group"*, *role="radiogroup"*) para identificar controles de formulario relacionados;
 - indica el formato esperado de los datos y pon un ejemplo;
 - ofrece instrucciones textuales al inicio del formulario, o de un grupo de campos, que describan la entrada necesaria;
 - ubica las etiquetas de tal manera que ayude a predecir las relaciones;
 - añade una descripción textual para identificar los campos obligatorios no rellenados;
 - indica los campos obligatorios usando la etiqueta *label* o *legend*;
- Utiliza el elemento *label* para asociar las etiquetas con los controles;
- Describe los grupos de campos utilizando los elementos *fieldset* y *legend*;
- Utiliza un botón adyacente a un campo para explicar el objetivo de ese campo. Esta técnica se considera una técnica de último recurso, pues únicamente debe utilizarse cuando las otras técnicas no se pueden aplicar en la página.

Técnicas para mejorar la usabilidad

- Si un cambio en un control de formulario causa un cambio de contexto, describe antes qué sucederá.

¡No lo hagas!

- No formatees 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 a la persona, a menos que esto ponga en riesgo la seguridad o el propósito del contenido.

Técnicas para cumplir el criterio

Si se requiere que la información de un campo esté en un formato de datos específico:

- Utiliza *role="alertdialog"* de ARIA para notificar a los productos de apoyo, como un lector de pantalla, los errores en la entrada de datos.
- Cuando la persona incluye datos que no cumplen con el formato o valores requeridos, avísale con un mensaje de texto explicando este hecho. Debes proporcionar:
 - ejemplos correctos, o
 - describir la entrada correcta, o
 - mostrar valores correctos similares a los introducidos por la persona, con instrucciones sobre cómo ingresar uno de estos valores correctos si la persona decide hacerlo;
- Sugiere una forma de rellenar correctamente el campo.

Si se requiere que la información proporcionada por la persona esté en un rango de valores determinado:

- Utiliza *role="alertdialog"* de ARIA para notificar a los productos de apoyo, como un lector de pantalla, los errores en la entrada de datos.
- Cuando la persona incluye un valor que no está dentro de la lista de valores permitidos, avísale con un mensaje de texto explicando este hecho. Si es posible, debe incluir la lista de valores o sugerir el valor permitido que sea más similar al valor ingresado.
- Sugiere una forma de rellenar correctamente el campo.

Técnicas para mejorar la usabilidad

- Ofrece un mecanismo para que las personas puedan saltar del mensaje de error al campo y viceversa.
- Avisa a las personas tanto si el formulario ha sido enviado con éxito como si no.
- Valida el formulario en el lado del cliente y añade un mensaje de error en el DOM o con un *alert()*.

Prevención de errores en páginas legales, financieras y de datos

Para las páginas web que

- representan para la persona compromisos legales o transacciones financieras;
- modifican o eliminan datos controlables por la persona en sistemas de almacenamiento de datos; o
- envían las respuestas de la persona a una prueba,

se cumple al menos uno de los siguientes casos:

- La persona puede deshacer o cancelar el envío.
- El sistema detecta errores en la entrada de datos y permite a la persona corregirlos.
- Se proporciona un mecanismo para que la persona revise, confirme y corrija la información antes de enviar los datos.

Técnicas para cumplir el criterio

Si una acción de la persona provoca una transacción legal o comercial, como presentar la declaración de la renta o realizar una compra, elige una opción:

- **Antes del envío**, la persona 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**, da un tiempo a la persona para modificar o cancelar la acción.

Si una acción de la persona va a provocar un borrado de información, elige una opción:

- **Antes del envío**, la persona debe confirmar la acción para continuar; o debe seleccionar un *checkbox* para indicar que ha revisado los datos y está segura de querer borrar la información, lo cual se hará con un botón de envío.
- **Después del envío**, la persona puede recuperar la información borrada.

Si la página web incluye una prueba o examen:

- **Antes del envío**, la persona puede revisar y corregir sus respuestas; o se le pide confirmación para continuar con la acción seleccionada.

Técnicas para mejorar la usabilidad

- Avisa a la persona tanto si el formulario ha sido enviado con éxito como si no.
- Valida el formulario en el lado de cliente y avisa a la persona de los errores con un *alert()*.

Ayuda

Se proporciona ayuda dependiente del contexto.

Técnicas para cumplir el criterio

Si un formulario requiere entrada de texto:

- Ofrece un enlace de ayuda en cada página web.
- Ofrece ayuda mediante un asistente, es decir, mediante un avatar multimedia.
- Comprueba la ortografía del texto introducido en el campo y ofrece sugerencias de texto. Por ejemplo, mediante un mensaje “¿Quisiste decir...?” con un enlace a la palabra sugerida.
- Ofrece instrucciones textuales al inicio del formulario, o de un grupo de campos, que describan la entrada necesaria.

Si un formulario requiere entrada de texto en un formato de datos concreto:

- Indica el formato esperado de los datos y pon un ejemplo.
- Ofrece instrucciones textuales al inicio del formulario, o de un grupo de campos, que describan la entrada necesaria.

Técnicas para mejorar la usabilidad

- Utiliza el atributo *title* para presentar ayuda contextual.

Prevención de errores en todo tipo de páginas

Para las páginas web que requieren que la persona envíe información, se cumple al menos uno de los siguientes casos:

- La persona puede deshacer o cancelar el envío.
- El sistema detecta errores en la entrada de datos y permite a la persona corregirlos.
- Se proporciona un mecanismo para que la persona 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 la persona envíe información.

Entrada redundante

La información ingresada previamente por la persona, o proporcionada a la persona, y que debe ingresar nuevamente en el mismo proceso, puede autocompletarse o estar disponible para que la persona la seleccione.

Excepciones:

- Volver a ingresar la información es esencial.
- La información es necesaria para garantizar la seguridad del contenido (por ejemplo, incluir dos veces la nueva contraseña).
- La información ingresada anteriormente ya no es válida.

El objetivo de este criterio es garantizar que las personas puedan completar con éxito procesos de varios pasos, reduciendo el esfuerzo cognitivo y físico cuando se solicita información más de una vez durante un proceso. También reduce la necesidad de recordar información proporcionada en un paso anterior.

Por ejemplo, si en un proceso de compra tienes campos para introducir los datos de envío y de facturación, ofrece la posibilidad de completar los datos de facturación con los datos incluidos para el envío.

Hay que tener en cuenta que este criterio se refiere siempre a un mismo proceso, no agrega la obligatoriedad de almacenar información entre sesiones.

Técnicas para cumplir el criterio

- Proporciona los datos de un paso anterior en otros pasos del proceso.
- No pidas la misma información dos veces.

Autenticación accesible (Mínima)

No se requiere una prueba de función cognitiva (como recordar una contraseña o resolver un acertijo) para ningún paso de un proceso de autenticación.

Excepciones:

- Se proporciona un método **alternativo** de autenticación que no depende de una prueba de función cognitiva.
- Existe un **mecanismo** disponible para ayudar a la persona a completar la prueba de función cognitiva (como soporte de gestores de contraseñas o la posibilidad de copiar y pegar una contraseña).
- La prueba de función cognitiva consiste en **reconocer objetos** (pueden ser imágenes, vídeos o sonidos).
- La prueba de función cognitiva tiene como objetivo **identificar contenido no textual** que la persona proporcionó al sitio web (pueden ser imágenes, vídeos o sonidos).

El objetivo de este criterio es asegurar la disponibilidad de un método de inicio de sesión y acceso al contenido que sea accesible, fácil de usar y seguro, con un enfoque particular en mejorar la accesibilidad para las personas con discapacidad cognitiva. El proceso de autenticación incluye los posibles captchas que incluya, así como la opción de recuperar contraseña.

Se entiende por "prueba de función cognitiva" una tarea que requiere que la persona recuerde, manipule o transcriba información, por ejemplo:

- Memorización, como recordar un nombre de usuario, contraseña, conjunto de caracteres, imágenes o patrones.
- Transcripción de caracteres, como escribir un texto o un código sin poder copiarlo.
- Uso de ortografía correcta.
- Realización de cálculos.
- Resolución de rompecabezas.

Incluir tu nombre, correo electrónico o número de teléfono **no se considera una prueba de función cognitiva**, ya que son personales para cada persona y consistentes en todos los sitios web.

Un **error habitual** es obligar a ingresar una contraseña o un código en un formato diferente al original, por ejemplo, incluir solo los caracteres de ciertas posiciones:

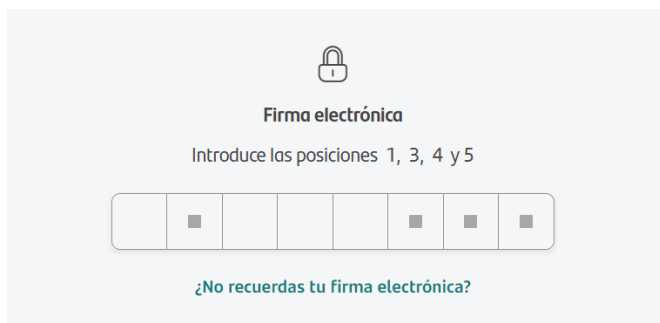


Ilustración de un formulario de autenticación. En la parte superior hay un icono de un candado. Debajo, el texto "Firma electrónica" y "Introduce las posiciones 1, 3, 4 y 5". A continuación, una barra de entrada con cinco cuadros; los cuadros 1, 3, 4 y 5 contienen un pequeño cuadrado gris, indicando que se deben introducir caracteres de esas posiciones. En la parte inferior, un enlace azul que dice "¿No recuerdas tu firma electrónica?"

Ilustración 28 Petición de unas posiciones concretas en una firma electrónica

Técnicas para cumplir el criterio

- Facilita la autenticación sin contraseña, a través de un enlace enviado mediante un correo electrónico.
- En HTML, marca correctamente los campos de correo electrónico y contraseña.
- Permite el uso del [WebAuthn](https://webauthn.io/)⁴⁴ o [oAuth](https://oauth.net/)⁴⁵ como alternativas al nombre de usuario y contraseña:
 - o *WebAuthn* es un estándar de autenticación web que permite a las personas iniciar sesión en sitios web y aplicaciones utilizando sus datos biométricos, como la huella dactilar o el reconocimiento facial.
 - o *oAuth* es un protocolo abierto para permitir la autorización segura con un método simple y estándar desde aplicaciones web, móviles y de escritorio.
- Utiliza la autenticación de 2 factores con dos técnicas diferentes.

¡No lo hagas!

- No impidas ingresar la contraseña o el código con el formato original con el que se creó.

⁴⁴ <https://webauthn.io/>

⁴⁵ <https://oauth.net/>

Autenticación accesible (Mejorada)

No se requiere una prueba de función cognitiva (como recordar una contraseña o resolver un acertijo) para ningún paso de un proceso de autenticación.

Excepciones:

- Se proporciona un método **alternativo** de autenticación que no depende de una prueba de función cognitiva.
- Existe un **mecanismo** disponible para ayudar a la persona a completar la prueba de función cognitiva (como soporte de gestores de contraseñas o la posibilidad de copiar y pegar una contraseña)

Este criterio es una versión más estricta del criterio anterior. En el nivel AAA no se admiten pruebas de función cognitiva que consistan en el reconocimiento de objetos o en la identificación de contenido personal no textual.

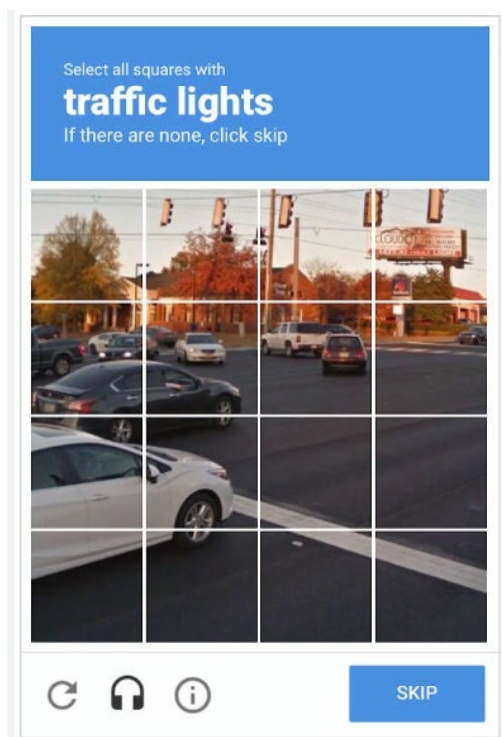


Ilustración 29 Los captchas basados en el reconocimiento de objetos están admitidos en el nivel AA (criterio 3.3.8) pero no el nivel AAA (criterio 3.3.9)

Técnicas para cumplir el criterio

Sigue las mismas técnicas, y evita los mismos errores, que en el criterio de conformidad 3.3.8.



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

Respetar 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

Este criterio se adoptó originalmente para abordar los problemas que los productos de apoyo tenían al analizar HTML directamente. Los productos de apoyo ya no necesitan analizar HTML directamente, ya que se comunican con el navegador a través de la API de accesibilidad y, en consecuencia, estos problemas ya no existen o se abordan mediante otros criterios. Por lo tanto, este criterio ya no tiene utilidad y por ello, ha sido eliminado oficialmente de las WCAG 2.2.

No obstante, recomendamos seguir sus consejos dado que facilitan la interpretación del código por parte de los navegadores.

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.

Excepción:

- Cuando las especificaciones permitan estas características.

Técnicas para cumplir el criterio

- Valida que cumples completamente la especificación. Puedes usar los validadores automáticos de código HTML y CSS referenciados en el capítulo [Herramientas de validación](#).
- Elige el estándar que desees y sigue su especificación, asegurando que la página web se puede procesar porque está bien formada o porque al menos:
 - las etiquetas de inicio y cierre se utilizan según la especificación.
 - cada atributo *id* tiene un valor diferente en la página web.
 - los elementos no tienen atributos repetidos.

¡No lo hagas!

- No cometes errores en las etiquetas de apertura o cierre ni en el marcado de los atributos.
- No utilices identificadores repetidos.

Nombre, función y valor

Para todos los componentes de la interfaz de usuario (como los elementos de formulario, los enlaces o los componentes generados por *scripts*, entre otros):

- el nombre y la función pueden ser determinados por software;
- los estados, propiedades y valores que pueden ajustar las personas se establecen por software utilizando métodos que son compatibles con los agentes de usuario, incluidos los productos de apoyo; y
- la notificación de los cambios en estos elementos está disponible para los agentes 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 utilizas componentes estándares de la interfaz de usuario en un lenguaje de marcado (por ejemplo, HTML):

- Utiliza el atributo *aria-label* para proporcionar una etiqueta invisible cuando no puedes utilizar una etiqueta visible.
- Utiliza el atributo *aria-labelledby* para proporcionar un nombre a los controles de la interfaz de usuario.
- Utiliza las características del marcado para exponer el nombre y la función; permite que las propiedades configurables por las personas se establezcan directamente; y avisa de los cambios; usando:
 - controles de formulario y enlaces estándar de HTML;
 - elementos *label* para asociar etiquetas de texto a controles de formulario; o el atributo *title* para identificar los controles si no puedes utilizar el elemento *label*;
 - el atributo *title* de los elementos *iframe* y *frame*;
 - HTML según la especificación.

Si utilizas programación para reutilizar un componente estándar de la interfaz de usuario en un lenguaje de marcado:

- Expón los nombres y roles, permite que las propiedades configurables por la persona se establezcan directamente, y avisa de los cambios. Puedes usar *aria-labelledby* para proporcionar un nombre a los controles de la interfaz de usuario.

Si utilizas un componente estándar de la interfaz de usuario con una tecnología de programación:

- Utiliza las características de la API de accesibilidad para exponer los nombres y las notificaciones de los cambios.

Si creas componentes de la interfaz de usuario con un lenguaje de programación:

- Hazlo utilizando una tecnología que soporte la notificación de accesibilidad de los cambios, utilizando:
 - los roles, estados y propiedades de ARIA para exponer la información sobre los componentes de la interfaz de usuario,
 - el atributo *aria-labelledby* para proporcionar un nombre a los controles de la interfaz de usuario.

¡No lo hagas!

- No programes de tal modo que los elementos *div* o *span* se comporten como controles de la interfaz de usuario sin proporcionarles un rol.
- No implementes controles personalizados que no usen la API de accesibilidad o lo hagan de manera incompleta. WAI-ARIA se puede utilizar para exponer la función, el nombre, el valor, los estados y las propiedades de un control personalizado a través de la API de accesibilidad.
- No te olvides de actualizar las alternativas textuales cuando cambie el contenido no textual.
- No te olvides de que los controles de la interfaz de usuario necesitan un nombre que pueda ser determinado 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 de tal forma 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

En el contenido implementado mediante lenguajes de marcado, los mensajes de estado pueden ser determinados por software a través de su rol o sus propiedades, de tal modo que puedan ser presentados a la persona usuaria de productos de apoyo sin recibir el foco.

Los mensajes de estado informan a las personas sobre el resultado de una acción, el estado de espera de una aplicación, el progreso de un proceso o la existencia de errores.

Para que se considere un mensaje de estado, **no puede entregarse a través de un cambio de contexto ni capturar el foco**. Este criterio no afecta a informaciones importantes que se presentan en diálogos modales, que deben ser reconocidos por la persona de forma inmediata y sí deben capturar el foco.

Por ejemplo, son mensajes de estado:

- 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 icono de ‘cargando’ en pantalla.
- Al enviar un formulario, se muestra un mensaje flotante que anuncia “Formulario enviado”.
- La lista de resultados de una búsqueda no se considera un mensaje de estado; sin embargo, el mensaje breve “Se encontraron 18 resultados” sí.

El objetivo de este criterio es garantizar que todas las personas reciban esta información y sean conscientes de los cambios importantes en el contenido sin interrumpir excesivamente su trabajo. Además, las personas que utilizan productos de apoyo puedan retrasar o suprimir esos mensajes de estado o, por el contrario, puedan resaltarlos cuando lo requieran.

Técnicas para cumplir el criterio

Si el mensaje de estado informa sobre el éxito o el resultado de una acción, o el estado de una aplicación:

- Indica que los datos se enviaron correctamente en combinación con el uso de *role*=“*status*”.

Si el mensaje de estado transmite una sugerencia o una advertencia sobre la existencia de un error:

- Utiliza el *role*=“*alert*” o *live regions* de ARIA para identificar los errores en combinación con alguna de estas técnicas:
 - proporciona descripciones de texto para identificar los campos obligatorios no completados;
 - proporciona descripciones de texto cuando los campos no se ajustan a los valores o formatos permitidos o necesarios;
 - ofrece sugerencias de corrección;
 - proporciona una revisión ortográfica y sugerencias para la entrada de datos.

Si el mensaje de estado transmite información sobre el progreso de un proceso:

- Utiliza el rol *log* para identificar la información secuencial actualizada.

- Utiliza el rol *progressbar*.
- Utiliza el rol *status* para presentar los mensajes de estado, y proporciona ayuda mediante un asistente en la página web.

Técnicas para mejorar la usabilidad

- Utiliza *aria-live* en chats o en el contenido mostrado con *hover* y *focus*.
- Utiliza los roles *marquee* o *timer*.
- Cuando corresponda, mueve el foco al nuevo contenido incluido mediante el uso de *alertdialog* para identificar errores.
- Ofrece opciones de personalización compatibles con que las alertas no esenciales sean opcionales.

¡No lo hagas!

- No proporciones mensajes de estado que no puedan ser determinados por programación mediante sus roles y propiedades.
- No utilices *role="alert"* o *aria-live="assertive"* en contenido que no es importante o sensible al tiempo.



Documentos PDF accesibles

Los archivos enlazados desde 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 explicamos cómo hacer un PDF accesible y cómo comprobarlo en una evaluación.

La accesibilidad en 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 2.2.

Como se explicó en la introducción de este 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 contenida en ellos si el documento PDF se ha elaborado de forma accesible. De esta manera, si el documento PDF es 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 2.2 salvo por los documentos PDF. Por lo tanto, si nuestros documentos PDF no son accesibles, nuestro sitio web tampoco lo será.

Los criterios de conformidad de las WCAG 2.2 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 únicamente 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. Estas técnicas también tienen beneficios adicionales, como mejorar la indexación y el **posicionamiento** del documento en los buscadores, así como mejorar su **usabilidad** para todas las personas.

Las técnicas específicas de PDF

Las WCAG 2.2 incluyen 23 técnicas específicas para la tecnología PDF.

1. **Añade textos alternativos a las imágenes** para que las personas sin visión puedan percibir las.
2. **Crea marcadores** para que todas las personas puedan desplazarse y encontrar la información 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. **Oculto las imágenes decorativas** para no molestar o dificultar la comprensión.
5. Indica los **controles de formulario obligatorios** para evitar errores a las personas.
6. **Utiliza los elementos de tabla (Table, TR, TH y TD) para etiquetar los 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**, mediante su expansión en el texto, o mediante un texto alternativo en su etiqueta, para que un lector de pantalla pueda leer su forma extendida.
9. **Incluye encabezados**, marcados como tales, para poder entender la organización de los contenidos y que la persona usuaria de un 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, etiquetados como tales, con un texto descriptivo** para que la persona 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 rellenan, especialmente cuando se accede con un lector de pantalla.
13. **Ofrece texto alternativo en los enlaces** cuando su texto no sea suficientemente explicativo.
14. Introduce en cada página **encabezados y pies** que ayuden a los usuarios a ubicarse.
15. **Añade un botón de “Enviar” a los formularios** para que las personas decidan cuándo se envían.
16. **Configura el idioma predeterminado** para que el lector de pantalla sepa cómo pronunciar, y las personas puedan comprenderlo.
17. **Numera visualmente las páginas**, por ejemplo en el pie del documento, **de forma coherente** con su numeración en los controles del visor PDF.
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.
20. **Repara las tablas mal etiquetadas** con el editor de tablas de Adobe Acrobat Profesional, hasta el momento, la mejor herramienta.
21. **Utiliza las etiquetas L y LI para etiquetar las listas** y permitir a las personas desplazarse rápidamente por los contenidos.
22. **Avisa a las personas cuando se equivoquen al rellenar un campo** porque debe seguir un formato determinado o introducir valores dentro de un rango; de este modo, facilitas a las personas cumplimentar el formulario y les evitas cometer errores al rellenarlo.
23. **Ofrece controles de formulario interactivos** asegurándote de que funcionan correctamente cuando se accede solo con el teclado.

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 ayuda 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; cómo exportar a PDF para no perder las mejoras de accesibilidad realizadas en el documento de origen; y cómo revisar el documento PDF para comprobar su accesibilidad.



Recomendamos el vídeo [Taller "PDF accesibles con Adobe Acrobat Profesional"](#)⁴⁶ impartido por Olga Carreras en el I Congreso Latinoamericano de Accesibilidad y Usabilidad.



Para modificar el documento de origen hemos elegido Microsoft Word (aunque en ocasiones también se mencionan Open Office, Google Docs o InDesign) y para modificar el propio PDF hemos elegido Adobe Acrobat Profesional. 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.

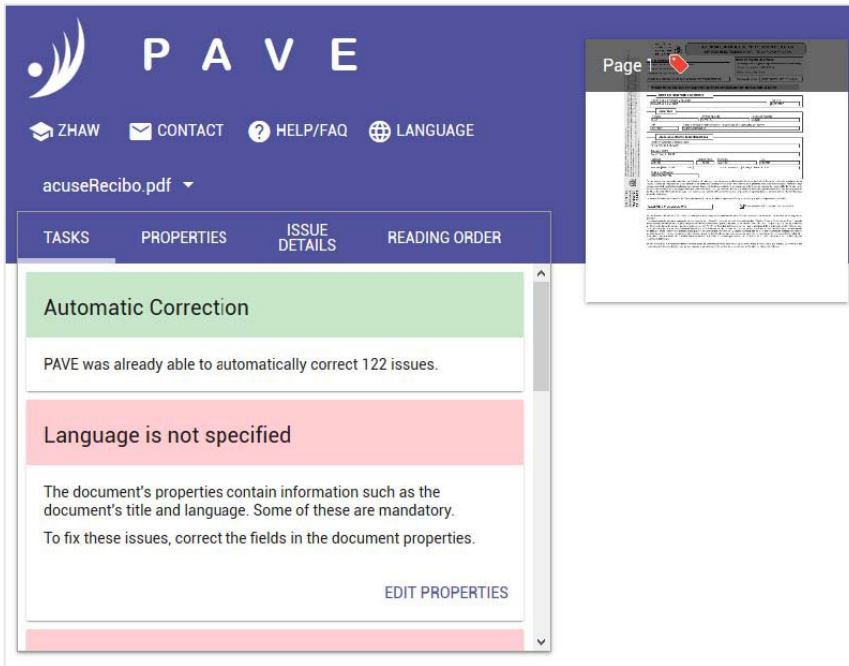
[Adobe Acrobat Profesional](#)⁴⁷ es la herramienta más recomendable actualmente para revisar y modificar la accesibilidad de un PDF, pero es un software propietario y de pago, diferente de Adobe Acrobat Reader. Aunque se puede trabajar la accesibilidad del PDF desde la versión Adobe Acrobat Pro 6, se recomienda utilizar la versión DC, por las mejoras significativas en sus herramientas específicas de accesibilidad.

⁴⁶ <https://olgacarreras.blogspot.com/2023/01/taller-pdf-accesibles-con-adobe-acrobat.html>

⁴⁷ <https://acrobat.adobe.com/es/es/acrobat/acrobat-pro.html>

Existe otra herramienta, [CommonLook PDF GlobalAccess](https://commonlook.com/accessibility-software/pdf/)⁴⁸, que es también local, de pago y bastante potente, y que permite revisar y modificar ciertos aspectos del PDF.

Como alternativa gratuita puedes probar [PAVE](http://pave-pdf.org)⁴⁹, una aplicación online que permite abrir y evaluar un PDF y corregir algunos de sus problemas de accesibilidad.



Captura de pantalla 2 Resultados de una evaluación con PAVE

⁴⁸ <https://commonlook.com/accessibility-software/pdf/>

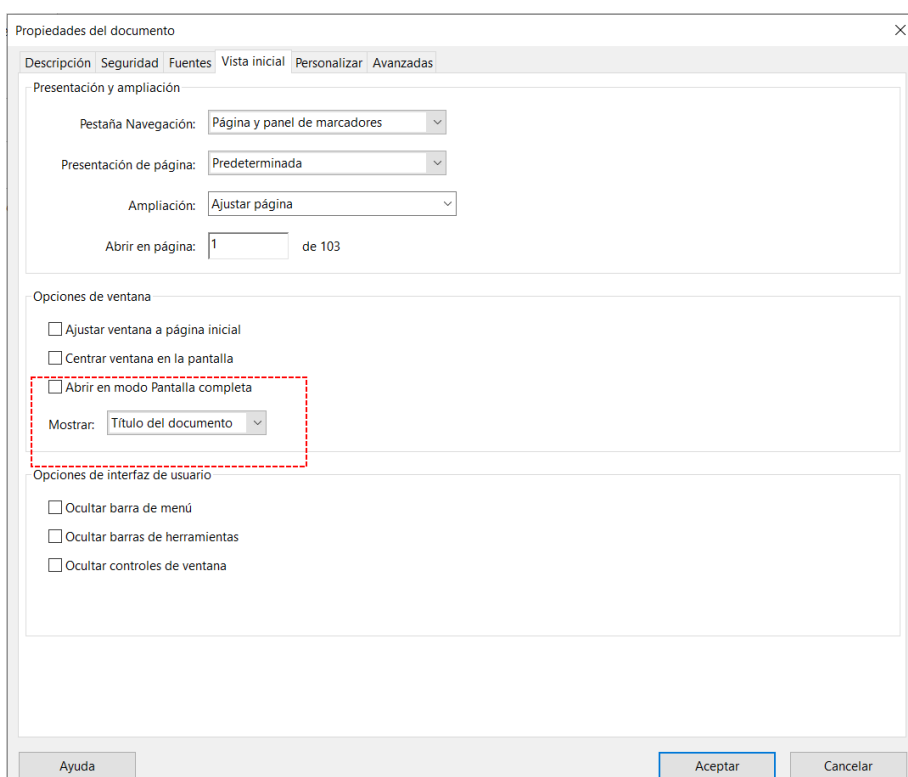
⁴⁹ <http://pave-pdf.org>

Título del documento

El título del documento es lo primero que anuncia el lector de pantalla a la persona usuaria, quien puede consultarlo 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”. Debes tener en cuenta que en Word nunca se muestra el título del documento en la barra de título, sino siempre el nombre del fichero, y será éste el que anuncie el lector de pantalla, pero el título que incluyas en las propiedades del documento se exportará al PDF.

En **Acrobat**, puedes incluir o modificar el título en el menú “Archivo > Propiedades > Descripción > Título”. Además, debes indicar que sea el título, y no el nombre del fichero, el que se muestre en la **barra de título** del lector de PDF. Esto se define en el menú “Archivo > Propiedades > Vista inicial > Mostrar > Título del documento”.



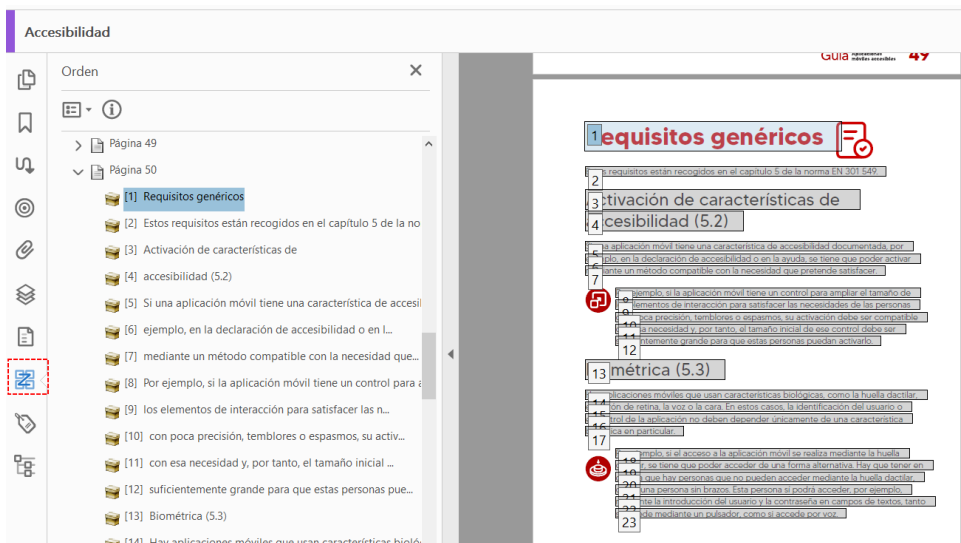
Captura de pantalla 3 “Vista inicial” de las propiedades de un documento en Adobe Acrobat Profesional DC

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 incomprensible.

El usuario de lector de pantalla puede decidir, en las opciones de lectura de Acrobat, que el contenido se lea según el orden definido en el panel “Orden” o en el panel “Etiquetas”, por ello debemos asegurarnos de que el orden sea correcto en ambos paneles.

Cuando activamos el panel “Orden”, o la herramienta “Accesibilidad > Retocar orden de lectura”, cada contenido de la página se muestra con un número, que indica su orden de lectura



Captura de pantalla 4 Panel “Orden” de Adobe Acrobat Profesional

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 cambia para reflejar su nuevo orden de lectura.

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 utilices cuadros flotantes.
- Si maquetas en varias columnas, no las simules con tabuladores, cuadros flotantes o tablas sin bordes, sino utilizando la herramienta “Columnas”.
- Inserta las imágenes a medida que redactas el texto, en línea con el contenido.

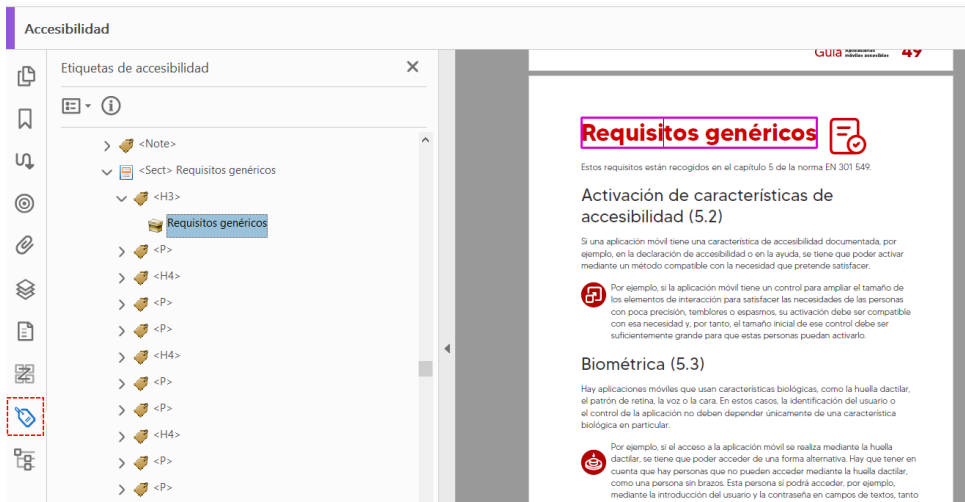
En **InDesign**, el orden de lectura viene determinado por el orden en el que se crearon las cajas. Este orden puede modificarse desde el panel Capas, ordenando los elementos de abajo a arriba.

Si cambias el orden de lectura y un elemento desaparece, y no sabes por qué o cómo solucionarlo, puedes consultar el video de Olga Carreras que referenciamos al comienzo del capítulo, donde se explica este problema común.

Etiquetado semántico

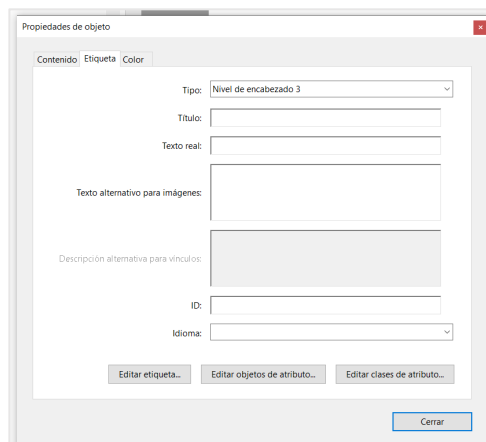
Cada contenido debe tener **una etiqueta asociada internamente** que indique qué tipo de contenido es: un encabezado, un párrafo, una lista, una tabla, una imagen, etc. Esto permite, entre otras cosas, que el lector de pantalla anuncie correctamente cada contenido según la etiqueta que tiene asociada, por ejemplo "Lista con cuatro elementos" o "Encabezado de primer nivel". Además, facilita que las personas usuarias de un 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 Profesional, 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 el panel, sin embargo, se recomienda generar siempre el PDF etiquetado desde el programa de origen para evitar problemas.



Captura de pantalla 5 Panel "Etiquetas" de Adobe Acrobat Profesional

Desde el árbol del panel "Etiquetas" puedes crear y eliminar etiquetas, o seleccionar cualquiera de ellas para acceder a sus propiedades e indicar, entre otras cosas, de qué tipo es.



Captura de pantalla 6 Propiedades de una etiqueta en Adobe Acrobat Profesional

Como ya hemos comentado, las personas usuarias de un lector de pantalla pueden decidir que el orden de lectura sea el orden del etiquetado, por ello, **el orden de las etiquetas en este árbol también debe ser adecuado** para reflejar en qué orden debe leerse el contenido.



Para poder revisar y corregir el etiquetado, necesitas conocer las etiquetas del estándar PDF.

[Descarga "Cheat Sheet Etiquetas Estándar en PDF accesibles"](#) de Olga Carreras⁵⁰

Si el documento se maqueta con las herramientas adecuadas en el programa de origen (encabezados, listas, tablas, índice de contenidos, etc.), el etiquetado del documento será correcto y apenas será necesario retocarlo en Acrobat.

Las **buenas prácticas** a seguir en el programa de origen son:

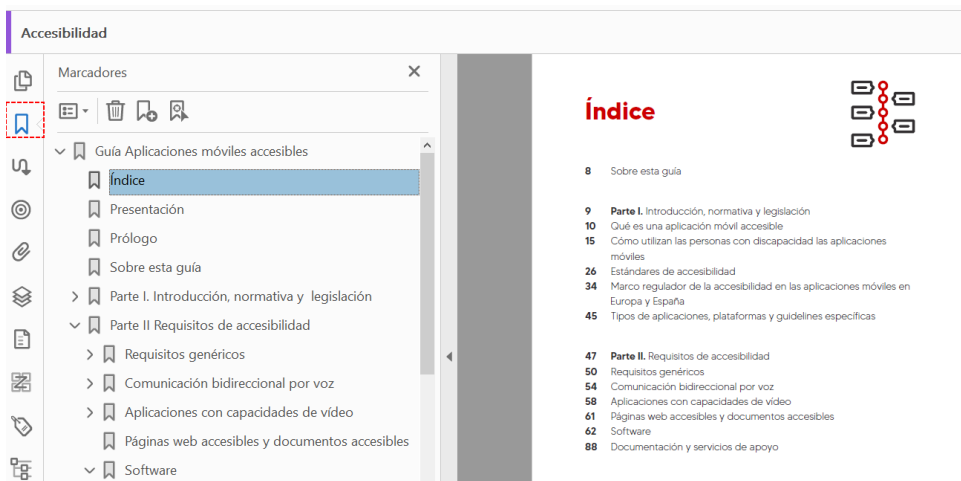
- Define los títulos mediante estilos de título, nunca los simules cambiando simplemente su tamaño o su color. En InDesign debes crear estilos de párrafo e indicar en sus propiedades la etiqueta de exportación a PDF.
- Asigna los títulos de forma coherente y no en base a su aspecto. No te saltes niveles.
- Define las listas mediante la herramienta de lista con viñetas o lista numerada.
- No incluyas retornos de carro para separar párrafos o saltar de página, ya que se exportan como párrafos vacíos que confunden a las personas usuarias de un lector de pantalla. La separación entre los contenidos debe definirse mediante sus márgenes, espaciados e interlineados. De igual modo, los saltos de página deben incluirse con la herramienta "Salto de página".
- Si incluyes notas, un índice de contenidos, referencias bibliográficas, u otro contenido similar, se debe hacer siempre con la herramienta correspondiente.

50

https://www.usableyaccessible.com/archivos/AA_Cheatsheet_Etiquetas_Estandar_PDF_accesible.pdf

Marcadores

El panel “Marcadores” de Acrobat Profesional 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 de éste.



Captura de pantalla 7 Panel “Marcadores” de Adobe Acrobat Profesional

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”.

Por otra parte, en las propiedades del documento puedes indicar que el índice de marcadores esté visible por defecto cuando se abra el documento.

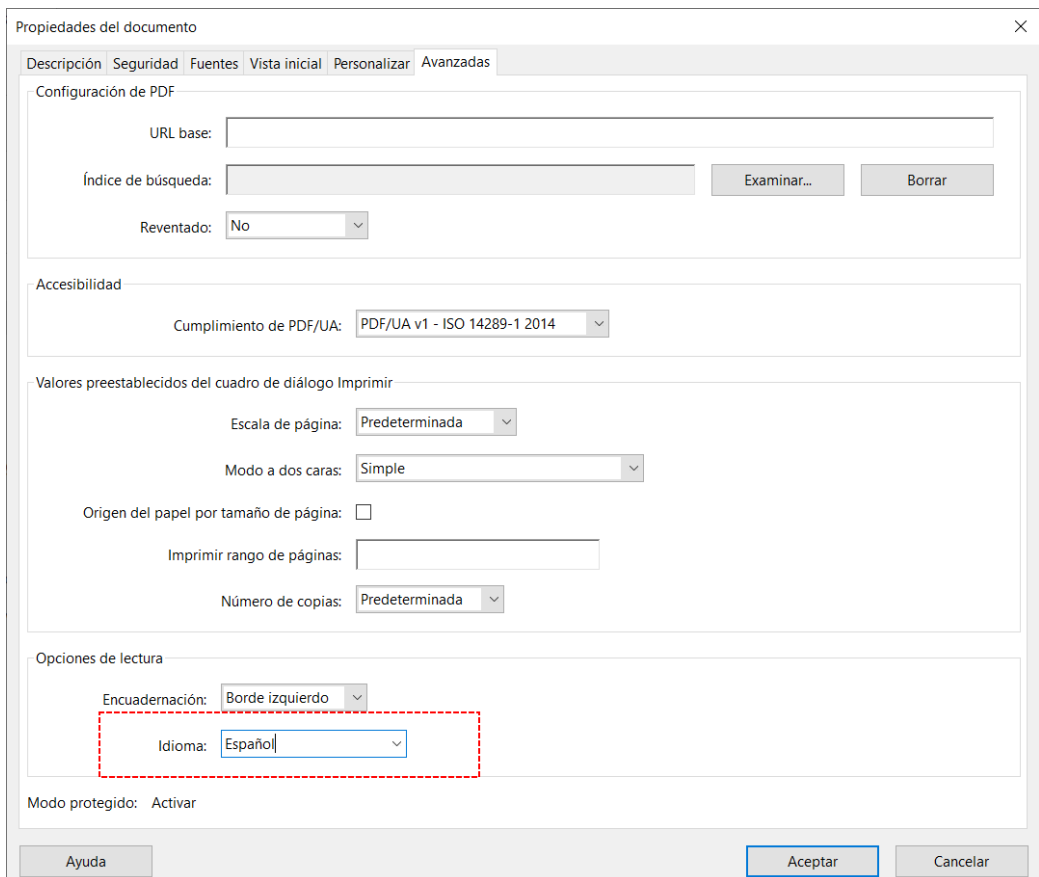
Idioma del documento

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 con la fonética adecuada.

Puedes indicar el idioma del documento en el programa de origen:

- en Microsoft Word desde el menú “Revisar > Idioma”;
- en OpenOffice Writer desde el menú “Herramienta > Idioma”;
- en Google Docs desde el menú “File > Language”.

En Acrobat Profesional, el idioma del documento se define en el menú “Archivo > Propiedades > Avanzadas > Idioma”. 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.



Captura de pantalla 8 Selección de idioma en el menú “Propiedades del documento” de Adobe Acrobat Profesional

Si un contenido se encuentra en un idioma diferente al del idioma principal, se puede indicar en las propiedades de su etiqueta, desde el árbol del panel “Etiquetas”. El idioma de un contenido también se puede definir desde el panel “Contenidos”.



Ten en cuenta que, si un contenido tiene definido un idioma diferente en el panel “Contenidos”, prevalecerá sobre el idioma definido en las propiedades de su etiqueta (en el panel “Etiquetas”) y sobre el idioma definido en las propiedades del documento.

En Word puedes cambiar el idioma de un contenido con la herramienta de idioma, disponible desde la barra de estado del programa.

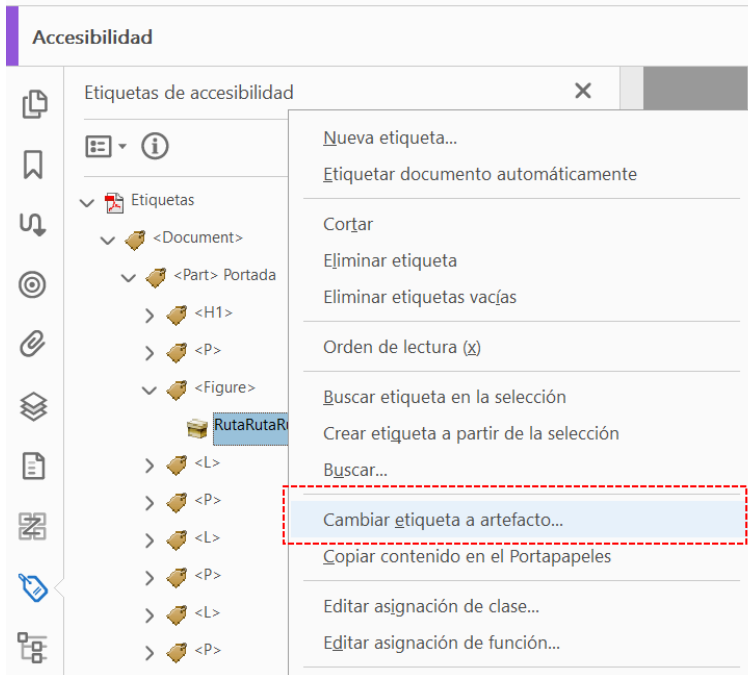
En InDesign puedes asignar un idioma a las cajas o a los estilos.

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 Profesional 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 9 Convertir una etiqueta en “artefacto” con Adobe Acrobat Profesional

La **cabecera** y el **pie de los documentos** deben incluirse siempre con la herramienta específica para ello. Esto permitirá que sean exportados como “artefactos de página”, evitando que el lector de pantalla los anuncie en todas las páginas. En InDesign, incluye estos elementos en la página maestra.

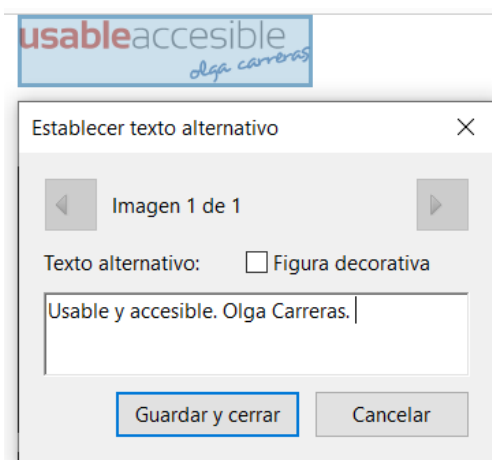


Nunca debemos eliminar un contenido del árbol de etiquetas.
Solo podemos eliminar etiquetas vacías.

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 las personas usuarias de productos de apoyo.

En Acrobat Profesional, 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 “Accesibilidad > Establecer texto alternativo” que permite recorrer todas las imágenes del documento para revisar y modificar su texto alternativo, o incluso indicar si son decorativas.



Captura de pantalla 10 Herramienta “Accesibilidad > Establecer texto alternativo” de Adobe Acrobat Profesional

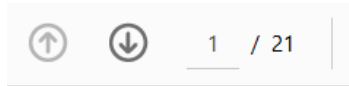
Desde la mayoría de los programas de origen se permite incluir textos alternativos a las imágenes y estos serán exportados con las imágenes al PDF.

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 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.



Captura de pantalla 11 Barra de paginación de Acrobat

Si ambas paginaciones no coinciden, se pueden cambiar los números de página de la cabecera o el pie desde el programa de origen, o bien modificar la paginación en Acrobat desde el panel “Miniaturas de páginas”. Para ello, selecciona la página o rango de páginas a las que se desea cambiar la paginación y accede a la opción “Numerar páginas” o “Etiquetas de página”, según la versión de Acrobat.

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 Profesional, 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**, es decir, el producto de apoyo anuncia 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.

En las últimas versiones de Acrobat, las propiedades de los enlaces también tienen un campo **“Descripción alternativa para vínculos”**, que es el equivalente al *title* de los enlaces en HTML. Esta descripción se muestra visualmente como un *tooltip*, pero hay lectores de pantalla, como NVDA, que no lo leen todavía en ningún contexto de acceso.

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 la tabla se puede simplificar, si se puede dividir en varias más sencillas o si la información se debería presentar en otro formato.

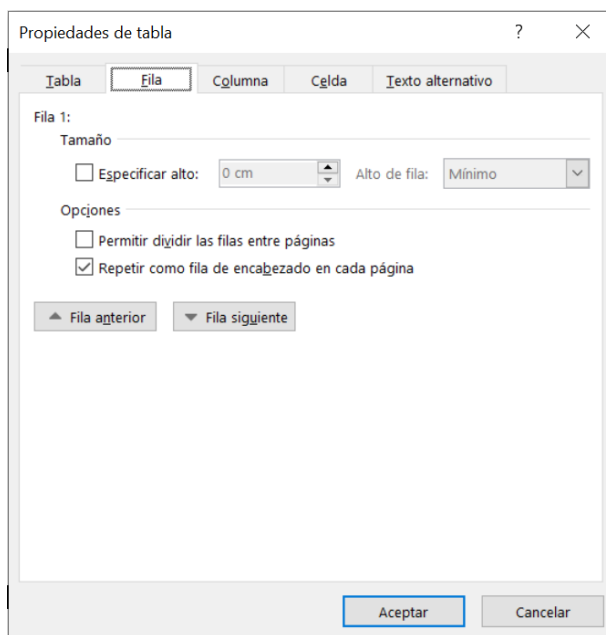
Es muy recomendable que **no combines las celdas**, de este modo, la tabla será más fácil de comprender y de acceder con un lector de pantalla. En la siguiente tabla, hay celdas con el mismo dato, sin embargo, no se unen, sino que se dejan como celdas separadas para que todas las personas puedan comprenderlas más fácilmente.

Tabla 9 Tabla con celdas que tienen el mismo dato, pero que no se han unido para mejorar su comprensión

Tarea	Número de horas	Precio (Euros)
Tarea 1	8	400
Tarea 2	2	100
Tarea 3	2	100

Si la tabla se queda dividida en varias páginas, es importante que repitas las celdas de encabezado en cada página y que no permitas que el contenido de una celda quede dividido entre las páginas. En Microsoft Word, esto se define de la siguiente manera:

- Selecciona la fila de encabezado y marca la opción **“Propiedades de la tabla > Fila > Repetir como fila de encabezado en cada página”**
- Desactiva la opción **“Propiedades de la tabla > Fila > Permitir dividir las filas entre página”**



Captura de pantalla 12 “Propiedades de tabla” en Word

Etiquetar correctamente una tabla

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), de este modo, el lector de pantalla anunciará las celdas de encabezado antes de leer cada celda de datos.

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.

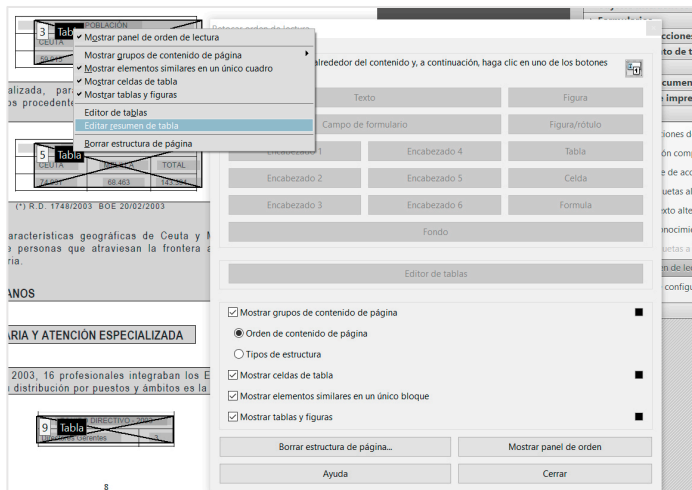
El **título de la tabla** debe tener la etiqueta *Caption*. El estándar PDF, que debemos respetar, indica que la etiqueta *Caption* ha de ser la primera etiqueta anidada dentro de la etiqueta *Table*. Esto permitirá que el lector de pantalla lea el título de la tabla cuando ésta coja el foco. Es muy recomendable que las tablas tengan siempre un título, incluido visualmente antes de la tabla. El título debe ser conciso e identificar la tabla.

Las **buenas prácticas en el programa de origen** para que las tablas se generen con un etiquetado correcto son:

- Inserta las tablas con la herramienta “Tabla”.
- Indica la fila de encabezado.
 - En Microsoft Word se define con la opción “Diseño > Fila de encabezado” y “Diseño > Primera columna”. Sin embargo, no se pueden identificar varios niveles de celdas de encabezado, esto debería hacerse desde Acrobat o simplificar la tabla.
 - En InDesign, cuando creas una tabla también hay una opción para indicar que tiene una fila con celdas de encabezado.
- Inserta el título de la tabla con la herramienta correspondiente, por ejemplo, en Word mediante la opción “Insertar título” del menú contextual de la tabla.

El campo "Descripción" en las "Propiedades de tabla > Texto alternativo" de Microsoft Word permite asociar una descripción a las tablas, sin embargo, la descripción no se exporta al PDF.

En Acrobat Profesional, 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 13 "Editar resumen de tabla" en Adobe Acrobat Profesional

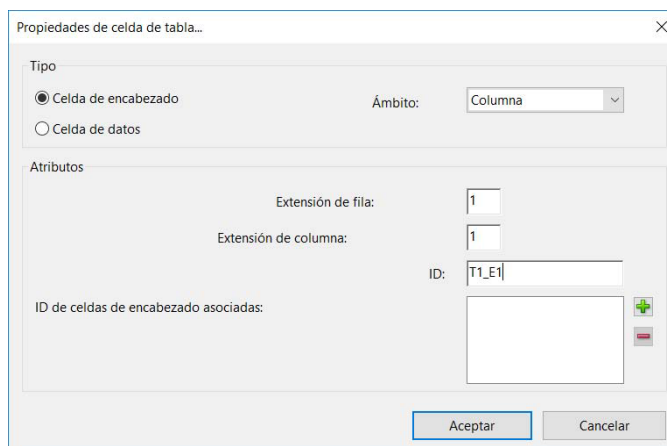
Indicar la relación entre las celdas de encabezado y las celdas de datos

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 Profesional 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.



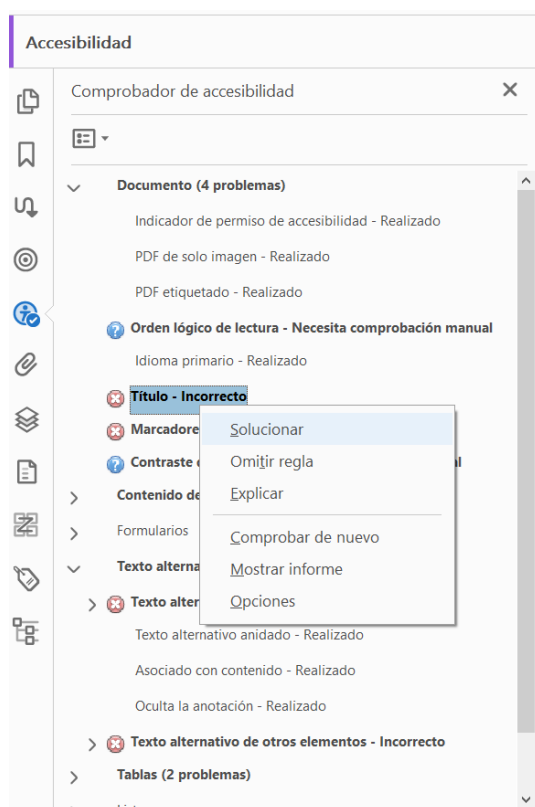
Captura de pantalla 14. "Editor de tablas" de Adobe Acrobat Profesional

Validar la accesibilidad del PDF

Nada puede sustituir a la evaluación manual de la accesibilidad del documento y su escucha con el lector de pantalla, sin embargo, como ayuda, pueden ser muy útiles los validadores automáticos de accesibilidad.

Desde la versión 2010 de **Microsoft Office** puedes revisar tu documento con el validador de accesibilidad que se encuentra en “Archivo > Información > 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.

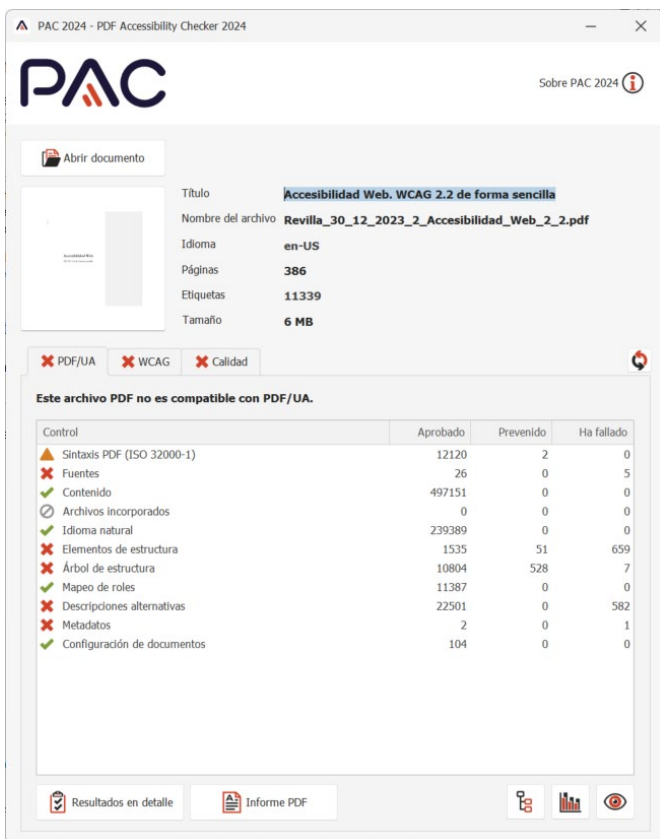
Adobe Acrobat Profesional tiene un validador de accesibilidad en “Accesibilidad > Comprobación completa”, que seguramente es el más fiable del mercado. Aunque no puede evaluar todos los requisitos de accesibilidad que debe cumplir el PDF, los errores que encuentra tienen siempre su razón de ser y deben corregirse.



Captura de pantalla 15 Resultados del validador de accesibilidad de Adobe Acrobat Profesional

Adobe Acrobat Profesional 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.

Aunque recomendamos validar siempre con la herramienta de Adobe Acrobat Profesional, hay muchos validadores automáticos de accesibilidad, como por ejemplo [PAC \(PDF Accessibility Checker\)](#)⁵¹ que es gratuito.



Captura de pantalla 16 Resultados de una evaluación con PAC



Puedes consultar un listado de validadores de PDF en "[Validadores y herramientas para consultorías de accesibilidad y usabilidad. Validar PDF](#)"⁵² de Olga Carreras.

⁵¹ <https://pac.pdf-accessibility.org/en/check>

⁵² https://www.usableyaccessible.com/recurso_misvalidadores.php#accesibilidadpdf

Exportar a PDF

Una vez validada la accesibilidad del documento de origen es el momento de guardarlo o exportarlo 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 que, como hemos visto, es la base de la accesibilidad del PDF.
- “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.

En las opciones de exportación es habitual que puedas seleccionar la **versión de PDF** con la que deseas que sea compatible. 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 o superior / PDF 1.7. No elijas la última versión, porque si seleccionas PDF 2.0, el PDF no podrá abrirse con Adobe Acrobat XI o versiones anteriores, solo con la versión DC.

Recuerda que **un documento escaneado como imagen nunca podrá ser un documento accesible**: no se puede buscar dentro de su texto; los productos de apoyo no pueden interpretar su estructura ni pueden 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 Profesional incluye una herramienta para ello, llamada “Reconocimiento de texto” o “Digitalizar y OCR”, según la versión de Adobe.

PDF/A y PDF/UA

Aunque el estándar PDF fue desarrollado por Adobe, desde 2008 es un estándar formal y abierto publicado por la ISO como ISO 32000⁵³. Existen diferentes subestándares de PDF: PDF/A, PDF/X, PDF/E o PDF/UA, que definen un PDF que debe cumplir ciertos requisitos.

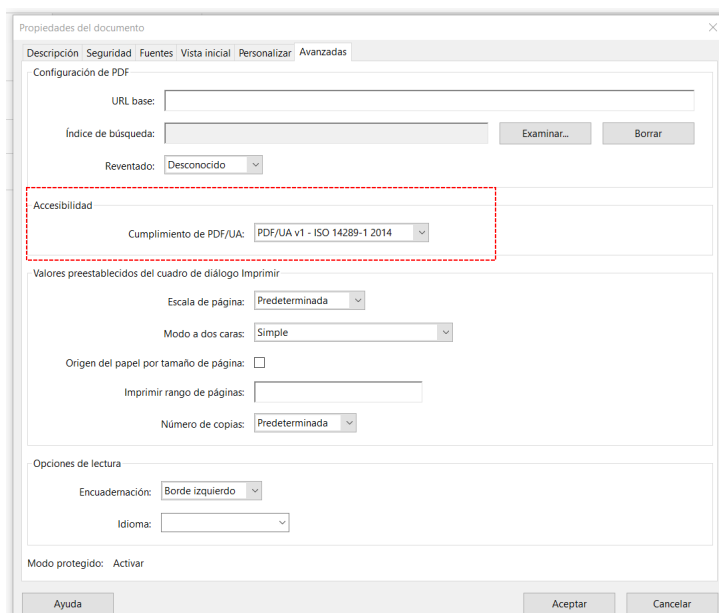
Es una confusión habitual creer que un **PDF/A** es un PDF accesible. La opción de guardar o exportar un PDF como PDF/A es una opción habitual en muchos programas. Un PDF/A no es un PDF accesible, simplemente es un PDF que está pensado para que se conserve igual a largo plazo, es decir, que pueda ser 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.

Las últimas versiones de Acrobat Profesional permiten al autor del documento indicar en sus propiedades si es un **PDF conforme con el estándar PDF/UA** ([ISO 14289-1:2014 Document management applications – Electronic document file format enhancement for accessibility – Part 1: Use of ISO 32000-1 \(PDF/UA-1\)](https://www.iso.org/standard/63534.html))⁵⁴.

⁵³ ISO 32000 <https://www.iso.org/standard/63534.html>

⁵⁴ <https://www.iso.org/standard/64599.html>



Captura de pantalla 17 “Propiedades del documento” de Adobe Acrobat Profesional

Ni el validador de accesibilidad de Adobe, ni su herramienta de validación de estándares, que incluye la validación de acuerdo al estándar PDF/UA, podrán indicar si el PDF es accesible, solo realizar algunas comprobaciones de sintaxis. Por tanto, solo una evaluación manual puede determinar si el PDF es accesible y cumple con el estándar PDF/UA.

El estándar PDF/UA incluye muchos requisitos comunes a las WCAG 2.2 y algunos específicos. Los requisitos adicionales del estándar PDF/UA son:

1. **Todas las fuentes deben estar embebidas** en el PDF.
2. **Todos los caracteres deben tener su correspondencia con caracteres Unicode**, lo cual asegura que el lector de pantalla puede interpretar y leer todo el texto correctamente.
3. Las **opciones de seguridad del documento** (definidas en “Archivo > Propiedades > Seguridad”) no deben impedir a los productos de apoyo acceder al contenido.



Puedes ampliar información sobre el estándar en “[PDF/UA. Descripción de la norma. Comparativa y relación con las WCAG](#)”⁵⁵ de Olga Carreras

⁵⁵ <https://olgacarreras.blogspot.com/2012/09/pdfua-descripcion-de-la-norma.html>

Guías de interés de documentos accesibles

- [*Documentación oficial de técnicas PDF accesibles*](#)⁵⁶.
- [*Guía “PDF accesibles con Adobe Acrobat Profesional”*](#)⁵⁷ de Olga Carreras
- [*Guía “Documentos PowerPoint accesibles”*](#)⁵⁸ de Olga Carreras
- [*“Accesibilidad en documentos PDF, ePub, Office y OpenOffice”*](#)⁵⁹ de Usable y accesible

⁵⁶ <https://www.w3.org/WAI/WCAG22/Techniques/#pdf>

⁵⁷ https://www.usableyaccesible.com/archivos/68067-ES-Olga_Carreras_eBook_Accessibility_in_PDF-D4%20FINAL-ua.pdf

⁵⁸

https://www.usableyaccesible.com/archivos/ES_Q421_eBook_Accessible_PowerPoint_Documents_with_Olga_Carreras_IAN_AD_A11y.pdf

⁵⁹ https://olgacarreras.blogspot.com.es/2009/04/dos-anos-de-usable-y-accesible.html#accesibilidad_pdf



ARIA, el aliado del HTML accesible

Los sitios web cuentan muchas veces con controles no nativos, habitualmente desarrollados con JavaScript, que los lectores de pantalla no pueden interpretar correctamente.

Esto provoca que las personas que los utilizan no los comprendan o no puedan usar dichos controles para interactuar con el sitio web.

Para solucionar este problema, el W3C desarrolló la tecnología ARIA, que permite añadir información semántica a cualquier elemento de la interfaz.

A través de la API de accesibilidad, el navegador transmite esa información semántica al producto de apoyo, y éste a su vez la traslada a la persona que lo utiliza.

De este modo las personas que usan lectores de pantalla u otros productos de apoyo pueden interactuar con el sitio web con normalidad.

Qué es ARIA

En un mundo ideal, los elementos HTML se usan y funcionan de la manera para la que están pensados: un botón es un botón, un enlace es un enlace y una lista es solo eso, una lista de elementos. Sin embargo, HTML no tiene una etiqueta para definir un árbol desplegable, unas pestañas o un mensaje de alerta. Por ello, no nos queda más remedio que crear estos nuevos componentes partiendo de elementos estándar de HTML más programación JavaScript (o AJAX o cualquier *framework* JavaScript).

Otras veces, a menudo por razones de diseño, utilizamos un elemento HTML para una función que no es para la que fue definido, a pesar de que existe un elemento HTML que podríamos usar en su lugar. Por ejemplo, podemos hacer que un *div* se comporte como un elemento de formulario (una casilla de verificación, un desplegable o un botón) modificando su apariencia mediante estilos 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 la persona usuaria de un 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 a la persona. 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 a la persona que es una lista de elementos.



Ilustración 30. Esquema de comunicación entre el navegador y el producto de apoyo a través de la API de accesibilidad. Ejemplo de lista de elementos.

En el siguiente ejemplo, el lector de pantalla anunciará “Lista con 3 elementos”.

```


- Proyectos
- Recursos
- Presupuestos

```

```

<ul>
  <li>Proyectos <ul>[...]</ul></li>
  <li>Recursos <ul>[...]</ul></li>
  <li>Presupuestos <ul>[...]</ul></li>
</ul>

```

¿Cómo hacemos que el producto de apoyo anuncie a la persona 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 y estados, por ejemplo, el nombre con el que queremos que se anuncie el árbol, que se indique si está abierto o cerrado, etc.

```


- Proyectos
    [...]

```

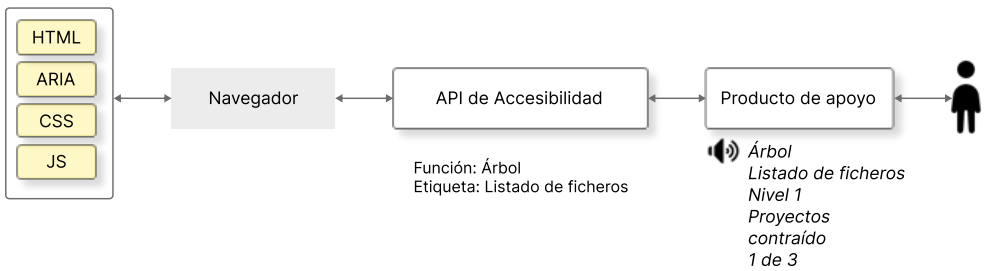


Ilustración 31. Esquema de comunicación entre el navegador y el producto de apoyo a través de la API de accesibilidad. Ejemplo de árbol definido con ARIA.

De esta manera, la persona usuaria del producto de apoyo tendrá toda la información para poder comprender y manejar el árbol. En este caso, un lector de pantalla como NVDA dirá “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.

ARIA nació en 2008. La primera recomendación, WAI-ARIA 1.0, se publicó en 2014. Actualmente, la última recomendación es WAI-ARIA 1.2 de 2023 y prácticamente todos los [navegadores y productos de apoyo dan soporte a ARIA](https://caniuse.com/#feat=wai-aria)⁶⁰.

Poco a poco va siendo una tecnología más conocida, pero a menudo mal implementada, posiblemente porque, como hemos visto, sólo conociendo la problemática de las personas que acceden con productos de apoyo y utilizando un lector de pantalla puedes realmente comprender su importancia y aplicarla correctamente.

Las WCAG 2.2 proponen usar ARIA como una forma óptima para cumplir con algunos de sus criterios de conformidad, ya que, bien utilizada, permite que diseños y funcionalidades muy complejos sean perfectamente accesibles.

⁶⁰ <https://caniuse.com/#feat=wai-aria>

Con ARIA se pueden crear controles complejos accesibles, como barras de progreso, deslizadores, desplegados, *tooltips* flotantes, alertas, *pop-hover* con opciones de diálogo, ordenación de listas de elementos, árboles de contenido contraíbles y desplegados, elementos sobre los que se puede hacer *drag-and-drop*, carruseles, pestañas, acordeones, barras de herramientas y de menú, *grids* dinámicas...

Con ARIA puedes definir áreas de contenido que se actualizan sin intervención de la persona usuaria y configurar cómo queremos que sean anunciadas por el lector de pantalla. Por ejemplo, podemos marcar un *timeline* de Twitter-X, un reloj que indica el tiempo que nos queda para completar un formulario o una notificación, de tal forma que las actualizaciones no le pasen desapercibidas a la persona que no puede verlas.

Pero también puedes utilizar ARIA para cosas más sencillas, marcar la función de una región de la página, dar nombre a los componentes de la interfaz o asociarles una descripción adicional, indicar la obligatoriedad de un campo de formulario o indicar que es inválido porque ha dado error de validación.

ARIA no es un lenguaje en sí mismo, sino un complemento de HTML5 y de SVG, ya que, aunque no lo tratamos en el libro, también permite dotar de accesibilidad a los contenidos SVG.

Ejemplo 1. Cambiar la semántica y el funcionamiento: una capa que se comporta como un botón

Imagina que, en lugar de poner un botón “Enviar” en un formulario, incluimos una imagen que realice la misma función con el siguiente código:

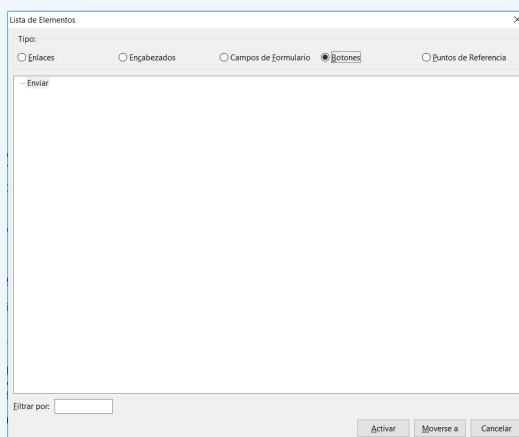


```
<div></div>
```

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 la nueva funcionalidad del `<div>` y que traslade la información al lector de pantalla, a través de la API de accesibilidad, debemos añadir el atributo `role="button"` a la etiqueta.

```
<div role="button"></div>
```

A partir de este momento, para el lector de pantalla es un botón: lo anuncia como tal (“Botón gráfico Enviar”); aparece en el listado de botones del lector de pantalla; y podemos llegar hasta él pulsando el atajo de teclado del lector de pantalla para los botones (habitualmente la letra “b”).



Captura de pantalla 18 Listado de botones de NVDA (insert+F7)

Sin embargo, sólo por decirle que es un botón, **no va a comportarse como un botón**, tendremos que añadir su comportamiento incluyendo los eventos correspondientes con *JavaScript* no intrusivo.

Aunque le asociemos los **eventos *onclick* y *onkeypress***, el `<div>` no tomará el foco cuando navegemos con el tabulador del teclado o un pulsador, como sí lo haría un `<button>`. Por tanto, no será accesible por teclado, sólo por ratón.

Para conseguir que tome el foco de teclado tendremos que **añadirle el atributo `tabindex="0"`**, que indica al navegador que incluya el `<div>` en la lista de elementos con los que la persona 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 (se presuponen añadidos los eventos *onclick* y *onkeypress* mediante *JavaScript* no intrusivo):

```
<div role="button" tabindex="0" id="buttonEnviar"></div>
```

El caso que acabamos de explicar es un ejemplo de algo que se puede hacer, pero **¿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 un determinado estilo, pero no podríamos diseñar un elemento nativo con ese aspecto visual.

Roles

Los roles en ARIA permiten indicar la función de un elemento de la interfaz. La función se asigna con el atributo *role*. **ARIA 1.2 define 94 roles**, doce más que la versión ARIA 1.1, categorizados de la siguiente manera:

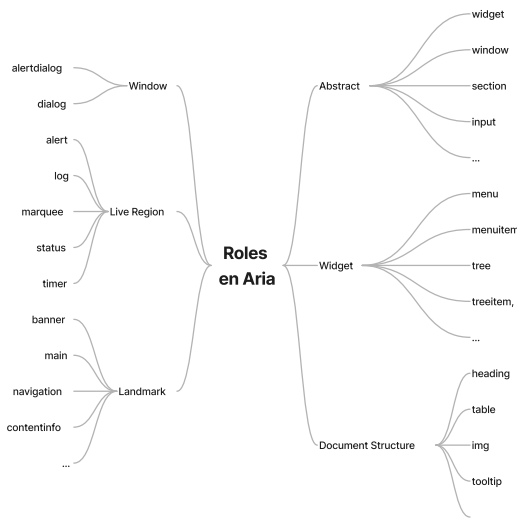


Ilustración 32. Esquema de roles en ARIA

- **Abstract:** *widget, window, section, input*, etc. Se usan solo para definir tipos de roles generales, no se aplican en los componentes y, por tanto, nunca deberías encontrarlos en una página HTML.
- **Widget:** elementos interactivos con funcionalidades más complejas que los elementos nativos de HTML: *menu, menuitem, tree, treeitem, tab, tablist*, etc.
- **Document Structure:** *heading, table, img, tooltip*, etc. No suelen ser interactivos.
- **Landmark:** *banner, main, navigation, contentinfo*, etc. Permiten definir las grandes regiones de la página, como las etiquetas de HTML5 (*header, main, nav, footer*, etc.). Las personas usuarias de un lector de pantalla tienen atajos para saltar de región en región y pueden 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, aquellas que se cambian automáticamente, sin intervención de la persona usuaria, y sin tomar el foco.
- **Window:** *alertdialog* y *dialog*, para las capas que se abren como ventanas y reciben el foco, esperando alguna acción de la persona usuaria.

Puedes acceder a la **definición de cada rol** y su detalle en el apartado “[Definition of Roles](#)”⁶¹ de la especificación WAI-ARIA 1.2 o consultar el [gráfico esquemático de relaciones entre los roles](#).⁶²

⁶¹ https://www.w3.org/TR/wai-aria-1.2/#role_definitions

⁶² https://www.w3.org/TR/2023/REC-wai-aria-1.2-20230606/img/rdf_model.svg

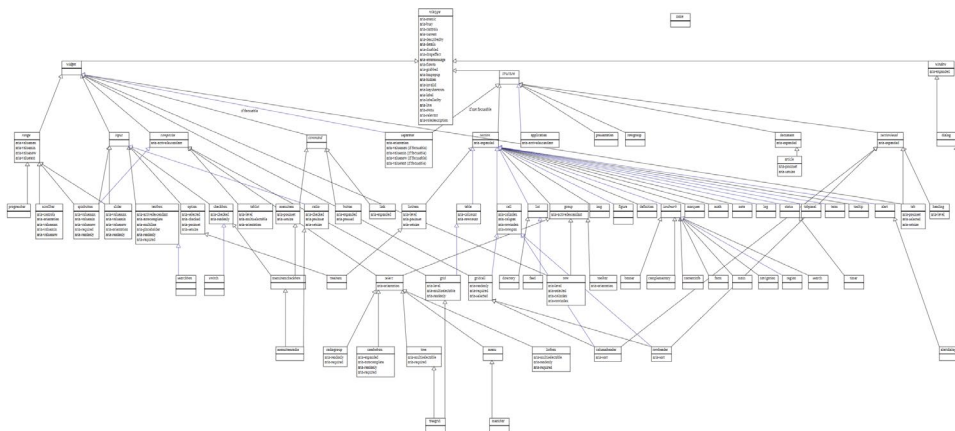


Ilustración 33 [Gráfico esquemático de relaciones entre roles de WAI-ARIA 1.2](#)

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 de la persona. En la práctica, no es necesario diferenciar una propiedad y un estado, y todos ellos comenzarán por *aria-*.

Hay **46 estados y propiedades** que se dividen en cuatro categorías:

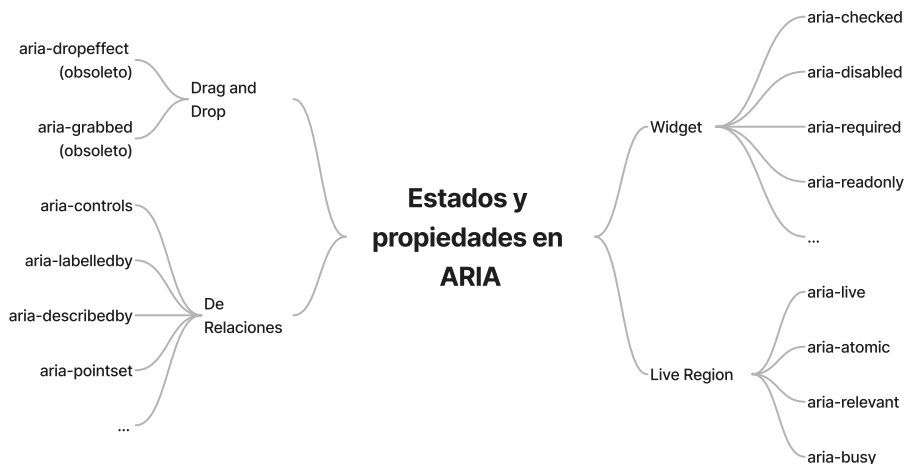


Ilustración 34 Esquema de estados y propiedades en ARIA

- **Atributos de Widget:** *aria-checked*, *aria-disabled*, *aria-required*, *aria-selected*, *aria-readonly*, *aria-expanded*, *aria-pressed*, *aria-label*, etc. para expresar atributos de componentes comunes que suelen recibir entradas de la persona usuaria o procesar sus acciones (chequeado, deshabilitado, obligatorio, de solo lectura, expandido, presionado, su nombre, etc.)
- **Atributos de Live Region:** *aria-live*, *aria-atomic*, *aria-relevant* y *aria-busy* que permiten definir cuándo se anunciarán a la persona usuaria del producto de apoyo los cambios producidos en las zonas que se actualizan solas sin recibir el foco; qué parte se anunciará; qué tipo de actualización queremos que se anuncie; o si queremos que temporalmente dejen de anunciarse. Puedes ampliar información y consultar ejemplos en el [artículo “Live Regions y WAI-ARIA”](#)⁶³.
- **Atributos de Relaciones:** 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-setsizes*, etc.
- **Atributos de Drag-and-Drop:** es decir, de componentes que se arrastran y se sueltan por la pantalla. Solo son dos atributos (*aria-dropeffect* y *aria-grabbed*) y actualmente se consideran obsoletos, en espera de que sean sustituidos por otros en el futuro.

Los estados y propiedades pueden cambiarse con *JavaScript*, de hecho, será muy importante **cambiarlos cuando la persona interactúe** con ellos.

⁶³ <https://olgacarreras.blogspot.com.es/2013/11/live-regions-y-wai-aria-como-mejorar-la.html>

Si la persona 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 por el lector de pantalla a la persona. Por el contrario, el valor del atributo *role*, que indica la función del componente, debe permanecer fijo.

Hay **tres propiedades fundamentales**, a las que hacen referencia muchas de las técnicas ARIA de las WCAG 2.2, que debemos utilizar continuamente y hacerlo bien. Son aquellas que permiten etiquetar o asociar descripciones a los elementos:

- *aria-label*,
- *aria-labelledby*
- *aria-describedby*.

Las veremos en profundidad en las próximas páginas.



Puedes acceder a la definición y el detalle de cada estado o propiedad en el apartado "[Definitions of States and Properties \(all aria-* attributes\)](#)"⁶⁴ de la especificación WAI-ARIA 1.2.

⁶⁴ https://www.w3.org/TR/wai-aria-1.2/#state_prop_def

aria-label

Con este atributo indicamos directamente **el texto 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 que 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>
```

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>
```

Otro uso habitual es para **distinguir regiones de la página** del mismo tipo, por ejemplo, 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 2.2 admiten el uso de *aria-label* como técnica suficiente.

Es habitual encontrar textos inadecuados o en un idioma incorrecto dentro del atributo *aria-label*. Es necesario comprobar que se utiliza correctamente, que el texto que incluye es adecuado y está en el idioma correcto.

aria-labelledby

Tanto *aria-label* como *aria-labelledby* se usan para **etiquetar un elemento**, es decir, para darle un nombre accesible que sea anunciado por los productos de apoyo. La diferencia es que con *aria-label* incluimos directamente el texto que queremos que funcione que nombre accesible, mientras que con *aria-labelledby* hacemos referencia al id del elemento (o los elementos) que funcionan como etiqueta de nuestro componente.

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 el siguiente ejemplo, el lector de pantalla anuncia el enlace como “Informe anual 2017 Descargar PDF (25 KB)” en vez de anunciarlo como “enlace PDF (25 KB)”, es decir, concatena la etiqueta de los dos elementos referenciados en el atributo *aria-labelledby*:

```
<h2 id="informe">Informe anual 2017</h2>
<p><a aria-labelledby="informe pdf" href="" id="pdf">
  Descargar PDF (25 KB)</a>
</p>
```

Como vemos, el atributo *aria-labelledby* puede hacer referencia a un id o a varios, en cuyo caso, se separan con espacios.

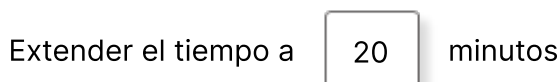
Es muy importante que siempre se referencie primero al id del propio elemento, para asegurar que el enlace es accesible en el acceso por voz (criterio 2.5.3 A).

Otro ejemplo de uso habitual se da en los **formularios de búsqueda**. En el siguiente caso, el lector de pantalla anuncia el campo como “Buscar”, aprovechando una etiqueta ya disponible en la página, esto es, la del botón adyacente.

Un formulario de búsqueda simple que consiste en un campo de texto rectangular vacío a la izquierda y un botón rectangular gris con el texto "Buscar" a la derecha.

```
<input name="searchtxt" type="text" aria-labelledby="btn">
<input name="searchbtn" id="btn" type="submit" value="Buscar">
```

También podemos usar *aria-labelledby* aprovechando el **valor de un campo de formulario**. En este ejemplo, el lector de pantalla anuncia el campo como “Extender el tiempo a 20 minutos” (en vez de “Extender el tiempo a 20”).

Un formulario que muestra el texto "Extender el tiempo a" a la izquierda de un campo de texto que contiene el número "20". A la derecha del campo de texto se encuentra el texto "minutos".

```
<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" aria-hidden="true">minutos</span>
```

Además, podemos usar *aria-labelledby* para **etiquetar zonas de la página**, o también para etiquetar **imágenes**, como en el siguiente ejemplo:



4 de 5

```
<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 de pantalla ignora las estrellas (porque tienen `alt=""`) y anuncia la capa como una imagen (gracias a `role="img"`) con la etiqueta "puntuación 4 de 5", que es la etiqueta que la imagen tiene asociada mediante el id del atributo `aria-labelledby`.

Se utiliza el atributo `aria-hidden="true"` para que el producto de apoyo no anuncie después de la imagen el contenido del `div`, pues sería innecesario y redundante. Además, se oculta visualmente el texto "puntuación".

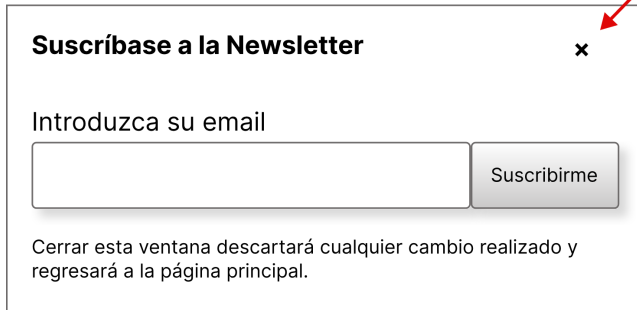
Este ejemplo también nos permite explicar que, aunque los elementos a los que referencie el atributo `aria-labelledby` estén ocultos con cualquier técnica, esto no afecta a su comportamiento, el lector de pantalla seguirá anunciando "puntuación 4 de 5".

Estos son varios de los ejemplos que incluyen las WCAG 2.2 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.

Imaginemos una ventana con un botón para cerrar, cierto contenido y al final de la ventana un mensaje de aviso.

A screenshot of a web form titled "Suscríbese a la Newsletter". The form contains a text input field labeled "Introduzca su email" and a "Suscribirme" button. At the top right of the form is a close button with an "x" icon, which is highlighted by a red arrow. Below the input field and button, there is a paragraph of text: "Cerrar esta ventana descartará cualquier cambio realizado y regresará a la página principal."

```
<button aria-label="Cerrar" aria-describedby="descClose">X</button>
```

[...]

```
<div id="descClose">  
<p>Cerrar esta ventana descartará cualquier cambio realizado y regresará  
a la página principal.</p>  
</div>
```

El atributo *aria-describedby* del botón cerrar nos permite asociarle una información adicional a su etiqueta, de modo que, cuando toma el foco, el lector de pantalla anunciará esa descripción adicional para que no le pase desapercibida a la persona usuaria: "Botón Cerrar. Cerrar esta ventana descartará cualquier cambio realizado y regresará a la página principal."

También puede usarse *aria-describedby* para asociar la **descripción extensa de una imagen**:

```
  
  
<div id="descWCAG">  
<p>Las pautas WCAG 2.2 se componen de 4 principios:</p>  
<ol>  
<li>Perceptible</li>  
<li>Operable</li>  
<li>Comprensible</li>  
<li>Robusto</li>  
</ol>  
</div>
```

Hay que tener en cuenta que el lector de pantalla leerá la descripción seguida y sin anunciar su marcado semántico.

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" />

<p id="descripcionContra" class="ayuda">
  La contraseña debe tener mínimo 6 caracteres
</p>
```

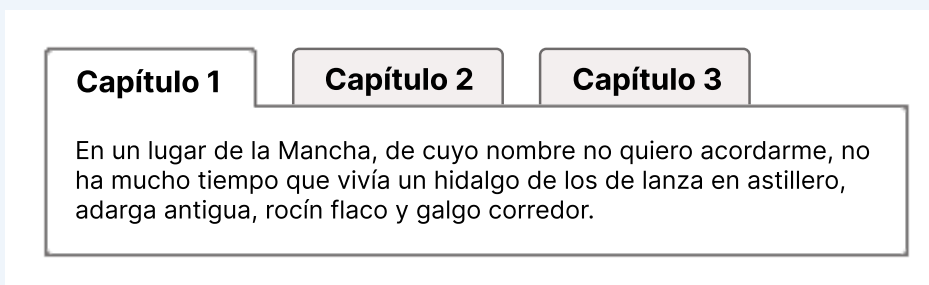
El lector de pantalla anuncia el campo con la etiqueta "Contraseña (obligatorio)" y le añade la descripción "La contraseña debe tener mínimo 6 caracteres."⁶⁵. De esta manera, los requisitos del campo no le pasarán desapercibidos a la persona usuaria de un lector de pantalla, que suele acceder a los formularios saltando de campo en campo.

Estos son varios de los ejemplos que incluyen las WCAG 2.2 para el uso de *aria-describedby* como técnica suficiente.

⁶⁵ El lector de pantalla NVDA anuncia exactamente ese campo como: "Contraseña obligatorio. Campo de edición contraseña. La contraseña debe tener mínimo 6 caracteres. En blanco". Según la combinación de lector de pantalla y navegador, el anuncio puede ser diferente.

Ejemplo 2. Navegación accesible mediante pestañas

Supongamos que tenemos un contenido estructurado visualmente mediante pestañas y maquetado con el código que se incluye a continuación:



```
<!-- pestañas -->
<ul>
  <li><a href="">Capítulo 1</a></li>
  <li><a href="">Capítulo 2</a></li>
  <li><a href="">Capítulo 3</a></li>
</ul>
<!-- contenidos -->
<section id="cap1">
  <p>En un lugar de la Mancha...</p>
</section>
<section id="cap2">
  ...
</section>
<section id="cap3">
  ...
</section>
```

Cuando se pulsa en una pestaña, se cambia su aspecto, para reflejar visualmente que está seleccionada, y se muestra el contenido del capítulo correspondiente. Para ello, nos valemos de *JavaScript* no intrusivo y de estilos CSS.

Sin embargo, las personas que acceden con un producto de apoyo, como un lector de pantalla, no entenderán la verdadera función de la lista de enlaces, qué ocurre cuando se pulsan, 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 (`<ul role="tablist">`) y darle un nombre accesible con `aria-label` o `aria-labelledby`. Partimos de una lista `` y no de unos `<div>` anidados porque, de este modo, conservarán su relación aunque se modifique el modo de presentación, por ejemplo, en el acceso sin CSS.
- Luego, debe indicarse que cada elemento de la lista (``) cumple el rol de presentación de contenidos (`<li role="presentation">`), así, el lector anunciará su contenido, pero no su rol nativo. Cuando `role="presentation"` se aplica a una imagen, es como aplicar un `alt=""` o un `aria-hidden="true"`, es decir, es ignorada por el lector de pantalla. Pero cuando se aplica a otro elemento, el lector de pantalla anuncia su contenido textual pero eliminando la información semántica de su rol, es decir, sin anunciar que es una lista, una tabla o un encabezado.

- Ahora, vamos a añadir varios atributos a los enlaces que hay dentro de cada elemento de lista:
 - el rol de pestaña (``) que anula su rol nativo de enlace.
 - el atributo `aria-controls="cap1"` que relaciona la pestaña con la sección que controla.
 - el atributo `aria-selected="true"` en la pestaña seleccionada y `aria-selected="false"` en las demás. El valor de este atributo deberá cambiarse dinámicamente por JavaScript cada vez que la persona seleccione una pestaña.
 - los atributos `aria-posinset` y `aria-setsizes` para indicar qué posición ocupa la pestaña y cuántas pestañas hay.
 - siguiendo el principio de mejora progresiva, el enlace por defecto de las pestañas (anulado luego por el Javascript) será un ancla a su sección (``).
- Por último, debemos añadir varios atributos a cada sección de contenido:
 - `role="tabpanel"` para indicar su función.
 - el atributo `aria-labelledby` para etiquetarla.
 - el atributo `tabindex="0"` para que pueda tomar el foco por teclado (más adelante explicamos cómo debe ser el acceso por teclado en el patrón de pestañas).
 - el atributo `aria-hidden="true"` en los `tabpanel` ocultos y `aria-hidden="false"` en el visible. El valor de este atributo deberá cambiarse dinámicamente por JavaScript cada vez que la persona seleccione una pestaña. Este paso no es necesario si se ocultan con el estilo `display:none`, pero sí es necesario si se ocultan con otros estilos, como `text-indent:-1000px`).
 - vamos a incluir un encabezado dentro de cada sección, que puede estar oculto visualmente, para que, desde otros contextos de acceso, como sin CSS, quede claro cuál es el contenido de la sección.

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-
  setsizes="3"><strong>Capítulo 1</strong></a></li>

  <li role="presentation"><a href="#cap2" tabindex="-1" role="tab"
  aria-controls="cap2" aria-selected="false" aria-posinset="2"
  aria-setsizes="3"> Capítulo 2</a></li>

  <li role="presentation"><a href="#cap3" tabindex="-1" role="tab"
  aria-controls="cap3" aria-selected="false" aria-posinset="3"
  aria-setsizes="3">Capítulo 3</a></li>
</ul>

<!-- contenidos -->
<section id="cap1" role="tabpanel" aria-hidden="false"
  tabindex="0" aria-labelledby="cap1">
<h2>Capítulo 1</h2>
<p>En un lugar de la Mancha...</p>
</section>
<section id="cap2" role="tabpanel" aria-hidden="true"
  tabindex="0" aria-labelledby="cap2">

```

```

<h2>Capítulo 2</h2>
...
</section>
<section id="cap3" role="tabpanel" aria-hidden="true"
tabindex="0" aria-labelledby="cap3">
<h2>Capítulo 3</h2>
...
</section>

```

¿Por qué se añade `tabindex="0"` y `tabindex="-1"` a los enlaces de las pestañas si son elementos que toman el foco por defecto? La razón es por cómo deben comportarse las pestañas en el acceso por teclado.

Si un mismo tipo de componente, como las pestañas, se maneja de forma diferente en cada sitio web, las personas que acceden sólo con el teclado tendrán que aprender su uso en cada página. Por esta razón, **el estándar ARIA también define cómo debe ser el acceso por teclado** en los componentes de tipo pestañas, menú o árbol, entre otros.

En el caso de las pestañas, una vez que toma el foco la primera pestaña, la manera de moverse a las demás pestañas debe ser con la flecha derecha e izquierda del teclado (no con el tabulador); 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; y con el tabulador puedes acceder al contenido de la pestaña.

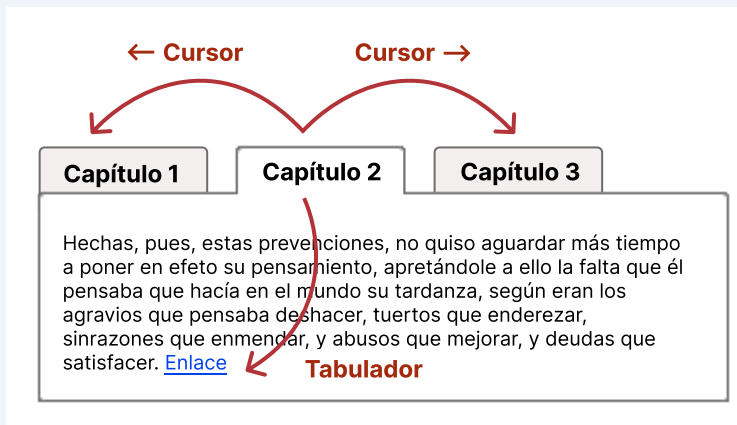


Ilustración 35 Acceso por teclado en las pestañas.

Por eso, a la primera pestaña se le incluye el atributo `tabindex="0"`, para que coja el foco, y a las demás se las elimina de la secuencia de tabulación con `tabindex="-1"` (atributo que indica que el elemento sólo puede tomar el foco por programación). A medida que la persona selecciona las pestañas con las flechas del teclado, se va cambiando dinámicamente por JavaScript el atributo de `tabindex="-1"` a `tabindex="0"`. Es lo que se llama el "roving `tabindex`". En otros componentes, como veremos más adelante, el foco se trabaja con el atributo `aria-activedescendant`.

Con el código mejorado, cuando la primera pestaña del ejemplo toma el foco, el lector de pantalla NVDA anuncia: "Capítulos del Quijote, pestañas. Capítulo 1 pestaña seleccionada 1 de 3." Según la combinación de lector de pantalla y navegador, el anuncio puede ser diferente. La persona usuaria de lector de pantalla, por tanto, podrá navegar de una pestaña a otra mediante el teclado, teniendo perfecto conocimiento de dónde está y qué está pasando en la página en todo momento.



Recomendamos el artículo [“Developing a Keyboard Interface”](#)⁶⁶ para conocer más sobre cómo el navegador interactúa con el teclado.

En la documentación [“Tabs Pattern” de ARIA Authoring Practices Guide](#)⁶⁷ del W3C y el artículo [“Tabbed interfaces”](#)⁶⁸ de Heydon Pickering encontrarás detallado el patrón de pestañas.

⁶⁶ <https://www.w3.org/WAI/ARIA/apg/practices/keyboard-interface/>

⁶⁷ <https://www.w3.org/WAI/ARIA/apg/patterns/tabs/>

⁶⁸ <https://inclusive-components.design/tabbed-interfaces/>

Buenas prácticas en ARIA

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.** Utiliza siempre que puedas etiquetas 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, sólo conseguimos que sea anunciado como un botón, pero será responsabilidad tuya que se comporte como tal.
3. **Utiliza los roles y las propiedades según la especificación.** Debes marcar la estructura del widget (*menubar, tablist...*) y las relaciones entre sus elementos (*aria-labelledby, aria-control...*). Recuerda que el rol no debe cambiar dinámicamente, solo se cambian las propiedades y estados.
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 *radiobutton* 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 redundante. Por ejemplo, estos ejemplos serían absurdos:

```
<input type="checkbox" role="checkbox" />
<img alt="Enviar" aria-label="Enviar" />
```

6. **Cambia los estados y las propiedades en respuesta a los eventos.** Si indicamos que un árbol está abierto con `aria-expanded="true"`, cuando la persona 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 tomar el foco de teclado por defecto son los elementos de formulario, los botones y los 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 únicamente 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 con 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”, lo cual incluye, de manera obligatoria, escuchar la página con un lector de pantalla.



En [ARIA Authoring Practices Guide \(APG\)](#)⁶⁹ encontrarás ejemplos para cada tipo de *widget*, con la correcta aplicación de los roles, propiedades y estados relacionados, y su adecuado comportamiento por teclado.

⁶⁹ <https://www.w3.org/WAI/ARIA/apg/>

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, es decir, una validación campo a campo.

Para que sea lo más accesible posible podemos utilizar ARIA de la siguiente manera:

```
<!--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="" aria-errormessage="error">
<!--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>
```

La capa de error tiene *role="alert"* y el atributo *aria-live="assertive"*, lo que provoca que en cuanto se modifica el contenido de la capa el lector de pantalla anuncia "Alerta" y lee su contenido (aunque el foco ya está en el siguiente campo).

El mensaje del error va precedido de un icono vectorial en formato SVG. Este elemento tiene los atributos *role="img"* y *aria-label="error:"*. Esto hace que el icono 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 de pantalla anuncie que el campo es obligatorio y que su entrada es inválida.

Asociamos el error al campo mediante el atributo *aria-errormessage* para que, cuando el campo coja el foco de nuevo, el lector me anuncie su error.

Es importante que la descripción del error incluya el nombre del campo "El campo Nombre es obligatorio", porque lo lee cuando el siguiente campo toma el foco. Si el mensaje de error es "El campo es obligatorio" parece que se refiere al campo actual, y no al campo que ha perdido el foco, que es el que se ha validado.

Cómo revisar ARIA

Con un lector de pantalla

La manera más recomendable de revisar nuestra página es **acceder con un lector de pantalla** como NVDA, JAWS o VoiceOver. Sólo así podemos escuchar cómo se anuncia cada elemento, si es comprensible, si necesita más información, y las diferencias al incluir roles y atributos ARIA. En el capítulo [Herramientas de trabajo](#) te recomendamos diferentes lectores de pantalla según la plataforma.

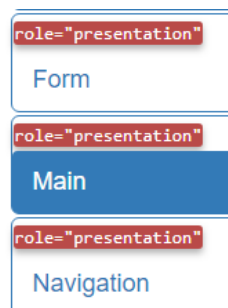
Revisando los atributos de ARIA

Otra opción es **auditar** si estamos aplicando correctamente ARIA según la especificación. Por ejemplo, las herramientas [Lighthouse](#)⁷⁰ o [AXE DevTools](#)⁷¹, que se instalan como pestañas en las “Herramientas para desarrolladores” del navegador Chrome, evalúan la corrección del código ARIA, es decir, 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.

También tenemos extensiones de navegador que **resaltan los roles y atributos ARIA** que se han aplicado en la página, como [Visual ARIA](#)⁷² o la [Web Developer Toolbar](#)⁷³.



Captura de pantalla 19 Visual ARIA resaltando roles, estados y propiedades



Captura de pantalla 20 Web Developer Toolbar resaltando roles

⁷⁰ <https://chrome.google.com/webstore/detail/lighthouse/blipmdconlkpinefehnmjammfjpmpbjk?hl=es>

⁷¹ <https://chrome.google.com/webstore/detail/axe-devtools-web-accessib/lhdoppojpmngadmndnejejpokejbdd>

⁷² <https://whatsock.com/>

⁷³ <https://chrispederick.com/work/web-developer/>

Inspeccionando el árbol de accesibilidad

Otra herramienta para revisar ARIA es la inspección directa del árbol de accesibilidad. Para introducir este concepto, debemos explicar antes cómo funcionan los navegadores más en profundidad.

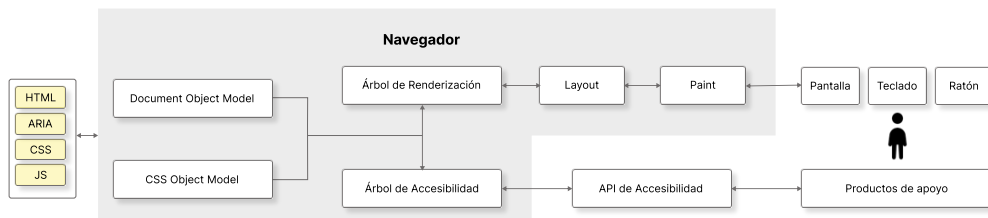


Ilustración 36 Esquema completo de funcionamiento del navegador y el producto de apoyo a través de la API de accesibilidad; y el resto de los periféricos a través del árbol de renderización.

En primer lugar, los navegadores convierten el marcado HTML en una representación interna llamada **DOM** (*Document Object Model*). El DOM contiene objetos que representan todos los elementos, atributos y nodos de texto del marcado.

Del mismo modo los navegadores convierten las propiedades CSS en una representación interna llamada **CSSOM** (*CSS Object Model*) que permiten la manipulación del CSS por parte de JavaScript.

A continuación, los navegadores crean un **árbol de renderización** y un **árbol de accesibilidad** basado en el árbol DOM y del CSSOM.

Para crear el **árbol de renderización**, el navegador recorre cada nodo visible empezando desde la raíz. Los nodos no visibles (como los metadatos) u ocultos por CSS se omiten porque no se reflejan en la salida representada.

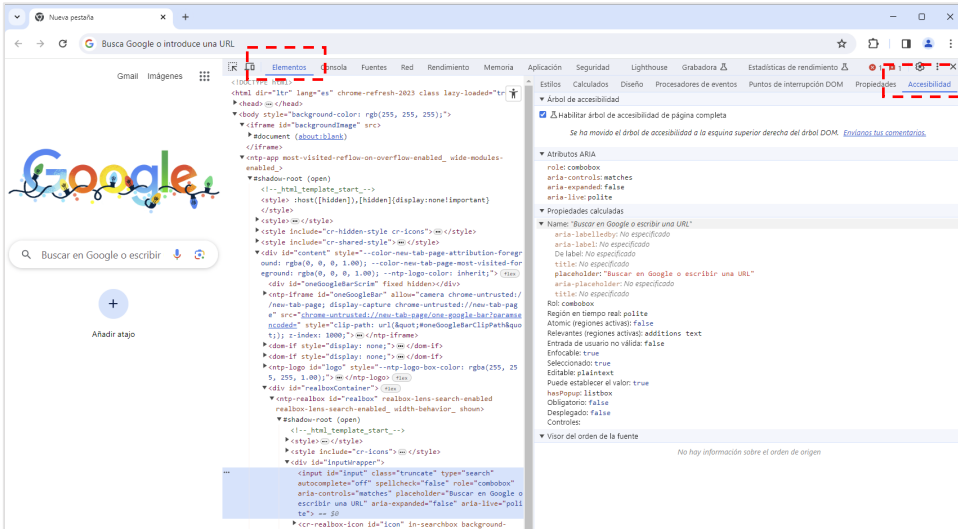
A continuación, el árbol de renderización pasa a la fase de “**Layout**” o definición de las posiciones y tamaños de los elementos que van a mostrarse, y posteriormente a la fase de “**Paint**” donde se cambian de color los píxeles de pantalla de acuerdo con lo que mande el árbol de renderización.

Los navegadores crean un árbol de accesibilidad, basado en el árbol DOM, que utilizan las API de accesibilidad específicas de cada plataforma para proporcionar una representación que pueda ser entendida por los productos de apoyo, como los lectores de pantalla. Para crear el **árbol de accesibilidad**, el navegador recoge la información relacionada con la accesibilidad de la mayoría de los elementos del DOM. Ten en cuenta que cuando los roles ARIA anulan la semántica del lenguaje nativo, no hay cambios en el DOM, solo en el árbol de accesibilidad.

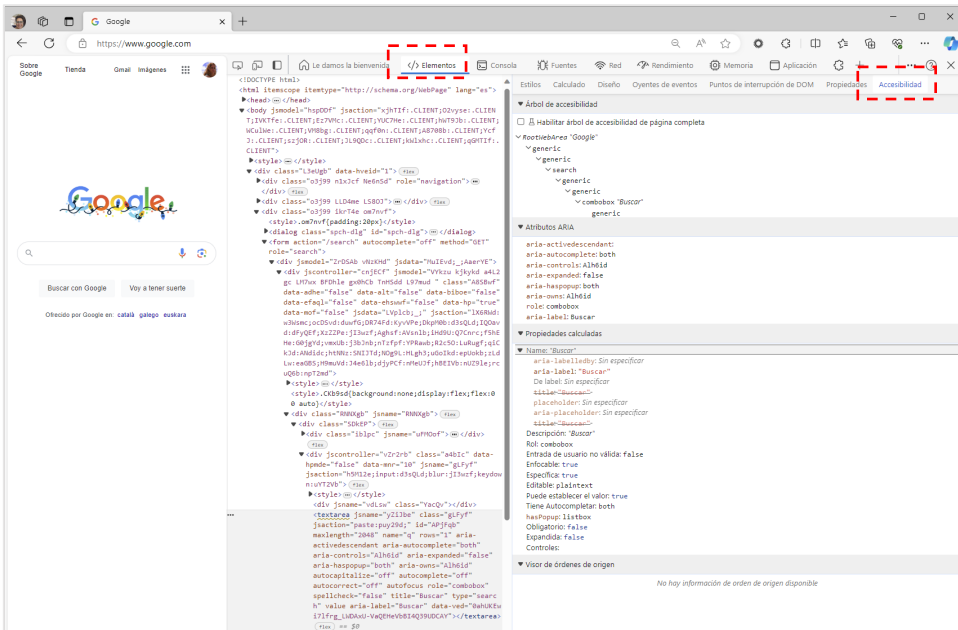
En concreto, hay cuatro propiedades relevantes en los objetos del árbol de accesibilidad:

- **Nombre del objeto.** Por ejemplo, un enlace con el texto “Leer más” tendrá “Leer más” como nombre (si no se ha indicado otro nombre con ARIA).
- **Descripción.** Por ejemplo, una tabla puede tener una descripción con el atributo “*summary*” (obsoleto en HTML 5) o el atributo *aria-describedby*.
- **Rol.** Por ejemplo, un objeto puede tener asociada una función de botón, de imagen, de enlace, de encabezado, etc.
- **Estado.** Por ejemplo, en una casilla de verificación si está activada o desactivada.

Es muy útil inspeccionar manualmente el árbol de accesibilidad para comprobar toda la información asociada a un objeto. Para hacerlo, debemos abrir las opciones para desarrolladores que nos ofrecen los navegadores. Tanto en Google Chrome como en Microsoft Edge lo puedes encontrar en las pestañas “Elementos > Accesibilidad”.



Captura de pantalla 21 Árbol de accesibilidad de Google Chrome



Captura de pantalla 22 Árbol de accesibilidad de Microsoft Edge

En la siguiente imagen mostramos las propiedades calculadas del [campo de búsqueda de Google](#).

The image shows a screenshot of a web browser's developer tools. On the left, the HTML code for a search input field is displayed. The code is as follows:

```
<textarea jsname="yZiJbe" class="glFyf"
jsaction="paste:puy29d;" id="APjFqb"
maxlength="2048" name="q" rows="1" aria-
activedescendant aria-autocomplete="both"
aria-controls="Alh6id" aria-expanded="false"
aria-haspopup="both" aria-owns="Alh6id"
autocapitalize="off" autocomplete="off"
autocorrect="off" autofocus role="combobox"
spellcheck="false" title="Buscar" type="searc
h" value aria-label="Buscar" data-ved="0ahUKEw
i7lfrg_LWDAXU-VaQEHeVbBI4Q39UDCAY"></textarea>
+1ex == $0
/div>
div class="dRYYxd"> +1ex
iiv>
/>
jscontroller="Dvn7fe" class="UUbT9 EyBRub"
!="display:none" jsname="UUbT9" jsaction="mouse
IzDCd;mouseleave:Mwfikb;hBEIVb:nUZ9le;YMFC3:VK
;vklu5c:k02QY;ldyIye:CmVOgc" data-ved="0ahUKEwi
z_LWDAXU-VaQEHeVbBI4Q4tUDCAo"></div>
jsname="mvaK7d" class="M8H8pb" data-ved="0ahUK
lfrg_LWDAXU-VaQEHeVbBI4Q4d8ICBM"></div>
```

On the right, the calculated ARIA properties for the search field are shown:

- ▼ Propiedades calculadas
- ▼ Name: "Buscar"
- aria-labelledby: Sin especificar
- aria-label: "Buscar"
- De label: Sin especificar
- title: "Buscar"
- placeholder: Sin especificar
- aria-placeholder: Sin especificar
- title: "Buscar"
- Descripción: "Buscar"
- Rol: combobox
- Entrada de usuario no válida: false
- Enfocable: true
- Específica: true
- Editable: plaintext
- Puede establecer el valor: true
- Tiene Autocompletar: both
- hasPopup: listbox
- Obligatorio: false
- Expandida: false
- Controles:

Captura de pantalla 23 Detalle de las propiedades del campo de búsqueda de Google

El campo cuenta con un atributo "title" y con un atributo ARIA "aria-label" con el mismo valor "Buscar". Para determinar el nombre tiene prioridad "aria-label", mientras que en este caso la descripción proviene del atributo "title".

Para comprender este proceso son imprescindibles dos recomendaciones del W3C referenciadas desde el estándar WAI-ARIA que todos los agentes de usuario deberían respetar para que la experiencia fuera consistente en todos ellos:

- [Accessible Name and Description Computation 1.1](#)⁷⁴ describe cómo los agentes de usuario determinan el nombre y la descripción de los objetos.
- [Core Accessibility API Mappings 1.1](#)⁷⁵ describe cómo los agentes de usuario deben exponer la semántica de los lenguajes de contenido web a la API de accesibilidad.

Seguimos analizando las propiedades del campo y vemos que su rol es "combobox". La etiqueta utilizada es "textarea", no "input" como cabría esperar, pero desde el momento en el cual le aplicamos con ARIA el atributo `role="combobox"` se convierte en un "combobox".

Un "combobox" es un "input" que controla otro elemento, como una lista desplegable o una tabla dinámica, que puede aparecer dinámicamente para ayudar al usuario a establecer el valor de esa entrada. En este caso, el campo de Google muestra sugerencias cuando el campo toma el foco o se escribe dentro de él.

⁷⁴ <https://www.w3.org/TR/acname-1.1/>

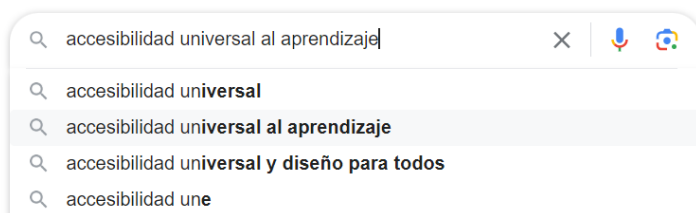
⁷⁵ <https://www.w3.org/TR/core-aam-1.1/>

Algunos de sus atributos se deducen, por ejemplo, no se considera un control obligatorio porque no se incluye este atributo, cuyo valor por defecto es *false* (no obligatorio). Otras propiedades se definen explícitamente con los atributos incluidos en el campo.

Repasemos sus atributos ARIA:

- **aria-autocomplete:** para indicar que ingresar texto desencadenará la visualización de sugerencias. Su valor *both* significa que ofrece los dos modelos posibles, una predicción de finalización del valor escrito en el campo (*inline*) y el modelo de lista (*list*).
- **aria-controls:** para asociar por código el campo con el componente que despliega.
- **aria-expanded:** para indicar si está expandido o contraído, valor que cambia dinámicamente.
- **aria-haspopup:** para indicar el tipo de elemento que despliega. Tiene el valor *both*, que no es uno de los valores admitidos por el estándar, así que el navegador interpreta que su valor es *listbox* y así lo refleja en el árbol de accesibilidad. Puedes detectar este error de sintaxis con el validador *AXE DevTools*, que hemos comentado anteriormente.
- **aria-owns:** para reflejar la relación padre/hijo visual, funcional o contextual con otro elemento (en este caso el elemento que despliega) cuando esta relación no se puede representar en la jerarquía del DOM.
- **aria-activedescendant:** en el ejemplo de las pestañas explicamos cómo manejar el foco de teclado mediante *"roving tabindex"*, bien, pues *aria-activedescendant* proporciona un método alternativo para gestionar el foco de teclado en los elementos interactivos que pueden contener varios descendientes enfocables, como los *combobox*. En vez de mover el foco entre los elementos, establecemos el foco DOM en el *combobox*, y mediante *aria-activedescendant* indicamos el elemento activo de la lista.

Cuando accedo con el teclado al componente, el foco permanece en el campo, aunque con las flechas del teclado me desplazo por las opciones sugeridas del listado:



Cuando me desplazo con las flechas del teclado por los elementos de la lista, el atributo *aria-activedescendant* cambia para reflejar dónde está el foco efectivo, aunque el foco DOM sigue en el campo:



Inspeccionando la información que llega a la API de accesibilidad

En el apartado anterior hemos explicado cómo podemos consultar el árbol de accesibilidad que genera el navegador.

Otra opción es consultar la información que tienen las API de accesibilidad en los diferentes sistemas operativos.

En Windows hay varias API de accesibilidad, destacan *UI Automation* y *Microsoft Active Accessibility (MSAA)*. Puedes probar la herramienta gratuita de Microsoft [Inspect](#)⁷⁶ para inspeccionar la información.

[“Accessibility Inspector”](#)⁷⁷ es una herramienta de desarrollo proporcionada por Apple como parte de las herramientas de desarrollo de Xcode, que proporciona información detallada sobre los elementos de la interfaz de usuario y sus propiedades.

En entornos Linux, [Accerciser](#)⁷⁸ es una herramienta de código abierto de depuración de accesibilidad para GNOME que permite probar y verificar la accesibilidad de las aplicaciones.

⁷⁶ <https://accessibilityinsights.io/docs/windows/getstarted/inspect/>

⁷⁷ <https://www.deque.com/blog/intro-accessibility-inspector-tool-ios-native-apps/>

⁷⁸ <https://help.gnome.org/users/accerciser/stable/introduction.html.en>

Las técnicas específicas de ARIA

Las WCAG 2.2 incluyen 23 técnicas⁷⁹ específicas para la tecnología ARIA:

1. Describe los controles de interfaz de usuario con la propiedad *aria-describedby*.
2. Identifica los campos obligatorios con la propiedad *aria-required*.
4. Utiliza los roles para informar de la función de cada componente de la interfaz de usuario.
5. Informa del estado de cada componente de la interfaz con los estados y propiedades ARIA.
6. Etiqueta objetos con *aria-label*.
7. Define el propósito de los enlaces con *aria-labelledby*.
8. Define el propósito de los enlaces con *aria-label*.
9. Crea una etiqueta concatenando varios nodos de texto con *aria-labelledby*.
10. Utiliza *aria-labelledby* para dar un texto alternativo al contenido no textual.
11. Utiliza los roles *landmarks* para identificar las zonas de la página.
12. Identifica encabezados con *role="heading"*.
13. Nombra las regiones y *landmarks* con *aria-labelledby*.
14. Provee etiquetas invisibles con *aria-label* cuando no puedas utilizar etiquetas visibles.
15. Describe las imágenes con *aria-describedby*.
16. Proporciona un nombre a los controles de la interfaz de usuario con *aria-labelledby*.
17. Utiliza roles de agrupación (como *role="group"* y *role="radiogroup"*) 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*.
22. Presenta mensajes de estado con *role="status"*.
23. Identifica actualizaciones de información secuenciales con *role="log"*.
24. Identifica semánticamente iconos de fuentes con *role="img"*.

⁷⁹ La técnica 3 se integró dentro de la técnica 5 en 2014.

Guías de interés de ARIA

- [*Especificación completa WAI-ARIA 1.2*](#)⁸⁰.
- [*Documentación oficial de las técnicas de ARIA*](#)⁸¹
- [*Artículos y reseñas de libro sobre WAI-ARIA*](#)⁸²
- [*"Novedades WAI-ARIA 1.2" de Olga Carreras, 2023*](#)⁸³

⁸⁰ <https://www.w3.org/TR/wai-aria>

⁸¹ <https://www.w3.org/WAI/WCAG22/Techniques/#aria>

⁸² <https://olgacarreras.blogspot.com/2009/04/dos-anos-de-usable-y-accesible.html#aria>

⁸³ <https://olgacarreras.blogspot.com/2023/06/wai-aria-12-novedades-de-la-nueva.html>



Single Page Applications (SPA) accesibles

En muchos sitios web no navegas realmente cuando pulsas los enlaces de las páginas, sino que todo el contenido de la página, o buena parte de este, se modifica dinámicamente dentro de la misma URL.

Es posible que ni siquiera te des cuenta... a menos que accedas con un lector de pantalla o únicamente con el teclado.

La experiencia de una persona ciega en una *Single Page Application* (SPA) en la que no se ha trabajado la accesibilidad puede ser desastrosa: escucha solo silencio cuando navega, no sabe dónde se encuentra y el foco de teclado se vuelve impredecible.

En este capítulo explicamos cómo remediarlo de forma sencilla.

Qué es una SPA

Una *Multiple Page Application* (MPA) y una *Single Page Application* (SPA) son dos enfoques diferentes para construir sitios y aplicaciones web.

El enfoque tradicional y más habitual es el de una *Multiple Page Application* (MPA), donde cada página se carga por separado desde el servidor cuando la persona solicita una nueva página. Es decir, cuando navegas de una página a otra, la aplicación realiza una solicitud al servidor para obtener la nueva página completa, de modo que la navegación implica recargar toda la página.

Por el contrario, en una *Single Page Application* (SPA), en lugar de cargar páginas completas desde el servidor, se carga una única página en el navegador y, a medida que la persona interactúa con la aplicación, se cargan los datos necesarios y se actualiza dinámicamente la interfaz de usuario. Se puede cambiar dinámicamente sólo una parte del contenido, por ejemplo, la región principal, pero también en ocasiones todo el contenido de la página.

Por tanto, la principal diferencia radica en cómo se maneja la navegación y la carga de contenido. Una MPA implica recargar páginas completas al navegar, mientras que una SPA carga la aplicación una vez y actualiza dinámicamente el contenido según las interacciones de las personas.

Un ejemplo de aplicación web SPA es Gmail, pero también, a día de hoy, sitios web como Zara, Yoigo o Pinterest.

Los principales problemas de accesibilidad de las SPA son:

- a. No se incluyen **mensajes de estado** cuando se modifica la página, o estos no son anunciados por los lectores de pantalla.
- b. El **título de la página** no cambia cuando se carga nuevo contenido.
- c. El **foco de teclado** no se maneja adecuadamente cuando se modifica la página.

Veamos ahora cómo solucionarlos.

Mensajes de estado

Cuando accedes a un sitio web tradicional con un lector de pantalla, éste te anuncia el título de la página que se ha cargado. Sin embargo, en una **SPA únicamente hay silencio**, porque realmente no se ha cargado la página, sólo se ha modificado el contenido dinámicamente.

Para evitar este problema debemos trabajar con mensajes de estado.



Las WCAG 2.2 no nos obligan a tener mensajes de estado, sólo a marcarlos adecuadamente con ARIA para que sean anunciados por el lector de pantalla.

Si te limitas a cumplir estrictamente con las WCAG 2.2, las SPA podrían ser inaccesibles con el lector de pantalla. Así que necesitamos incluir obligatoriamente mensajes de estado.

Las buenas prácticas a seguir son:

1. **Incluye un mensaje de estado cada vez que modifiques dinámicamente el contenido.** Lo más recomendable es que el mensaje esté visible para todas las personas.

2. **Incluye atributos ARIA** para que el lector de pantalla anuncie el mensaje aunque este no tenga el foco (consulta [el criterio 4.1.3 AA Mensajes de estado](#)):

```
<div role="status" aria-live="polite"><p>Cargando datos de la página... </p></div>
```

Si el mensaje contiene sólo una animación, ésta debe tener texto alternativo:

```
<div role="status" aria-live="polite"></div>
```

3. **Modifica dinámicamente el mensaje de estado** por "La página [título de la página] se ha cargado" cuando se carguen los datos.

El mensaje de que la página ya se ha cargado no se suele dejar visible para todas las personas, sólo para las personas que acceden con un lector de pantalla. En estos casos, recuerda que **NO** debes ocultar el mensaje con los estilos `display:none` o `visibility:hidden`, porque ocultan también el contenido para el lector de pantalla.

Título de página dinámico

En un sitio web tradicional MPA cada página que se carga tiene su propio título de página en la etiqueta `<title>` del `<head>`.

Sin embargo, en una SPA solo hay una página cuyo contenido se modifica dinámicamente, por tanto, aunque te parece que se cargan diferentes páginas, solamente es una con su título inicial. Por ello, **es imprescindible que se modifique dinámicamente el título de la página**, esto es, el contenido de la etiqueta `<title>` del `<head>`.



Cambiar dinámicamente el título de la página **no provoca que el lector de pantalla lo anuncie** automáticamente como un mensaje de estado. Por eso seguimos necesitando incluir los mensajes de estado que ya hemos comentado.

Los sitios web SPA de acceso público suelen modificar dinámicamente el título de las páginas por motivos de posicionamiento en buscadores. Sin embargo, **es habitual encontrar errores cuando el título de la página no es relevante para dicho posicionamiento**. Por ejemplo, podemos encontrar varias páginas de un proceso de compra que mantienen el mismo título; o que las páginas dentro de la zona de clientes, de acceso restringido, mantienen siempre el título de la página inicial.

El título de la página es importante para una persona usuaria de lector de pantalla, que puede preguntarlo en cualquier momento para ubicarse, por ejemplo, con NVDA o JAWS mediante el atajo `Insert+t`.

Un título claro también es importante para cualquier persona que accede a un sitio web, puesto que el título es el nombre de la página que se muestra en la pestaña del navegador, al guardarla en marcadores o al compartirla en redes sociales.

Hay otra razón por la cual el título puede ser relevante en una SPA. Ya hemos indicado que debemos incluir un mensaje de página cargada: “La página [título de la página] se ha cargado”. Si utilizamos el contenido de la etiqueta `<title>` para generar dinámicamente el mensaje de página cargada, será muy confuso cuando el lector de pantalla anuncie que se han cargado diferentes páginas siempre con el mismo nombre.

Un título único y descriptivo para cada página ofrece mensajes de estado más claros cuando se utiliza el título para generarlos: “La página ‘Proceso de compra. Paso 1 de 3. Datos del cliente’ se ha cargado”, “La página ‘Proceso de compra. Paso 2 de 3 Dirección de envío’ se ha cargado”, etc.

Control del foco de teclado

Hay muchas personas que acceden a las páginas mediante pulsadores o únicamente mediante teclado, entre ellas las personas que usan un lector de pantalla, muchas de las cuales son ciegas y no pueden ver dónde está el foco. Estas personas no saben que se encuentran en una SPA ni tienen por qué saberlo.

En un sitio web tradicional, el foco de teclado se sitúa al comienzo de la página cuando ésta se carga. Sin embargo, navegar en una SPA donde no se ha trabajado la accesibilidad puede resultar muy confuso cuando dependes del teclado y no puedes ver la pantalla. A menudo, el foco de teclado se torna impredecible, dejándote con la incertidumbre de a dónde se dirigirá.

Es común encontrarse en situaciones en las que el foco de teclado se pierde cuando se carga nuevo contenido, ya que el elemento con el foco desaparece y ya no está presente. Otras veces, el foco se queda donde está después de anunciarse la carga de una nueva página, generando la expectativa de que debería ubicarse al inicio de esta.

Por tanto, la tercera clave para que una SPA sea accesible es **manejar correctamente el foco de teclado cuando el contenido se carga**. Para hacerlo bien hay que aplicar el sentido común, conocer cómo acceden las personas usuarias de lector de pantalla y de teclado; e involucrarlas siempre que se pueda en las pruebas.

Opción 1. Mandar el foco de teclado al comienzo de la página

Si se está modificando dinámicamente todo el contenido central, una opción puede ser **mandar el foco al comienzo de la página**, que es lo que la persona espera. En concreto, es recomendable situarlo en el primer enlace del <body>, esto es, el enlace “Saltar al contenido” (consultar el [criterio de conformidad 2.4.1 A “Evitar bloques”](#)).

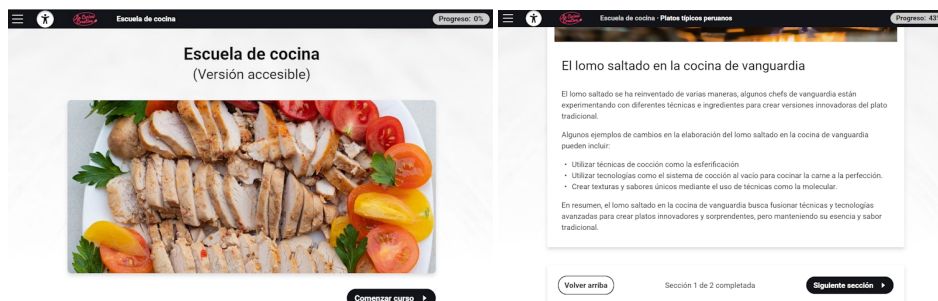
De este modo, después de que el mensaje de estado anuncie “La página [título de la página] se ha cargado”, el foco se sitúa siempre de manera predecible al comienzo de la página, en el enlace para saltar al contenido central.

Por tanto, la persona usuaria de un lector de pantalla tiene el feedback necesario y es un comportamiento predecible. Por su parte, la persona usuaria de teclado que ve la página y no utiliza un producto de apoyo, no se queda con el foco en una posición desde la que le resulte difícil regresar al inicio de la página o a la región principal. Por ejemplo, el foco de teclado no permanece en el pie de página tras pulsar uno de sus enlaces.

Opción 2. Mandar el foco al título de la página

En otros casos, puede generar una mejor experiencia de uso **enviar el foco al título de la página**, esto es, al `<h1>` en el `<main>`.

En el siguiente ejemplo de un curso accesible de *isEazy*, se pueden observar dos páginas, la página de inicio y una página interior. Cuando pulsas “Comenzar curso” o “Siguiente sección” no se navega a otra página, sino que se modifica dinámicamente el contenido de la página:



Captura de pantalla 24. Páginas de un curso accesible de *isEazy*

Teniendo en cuenta los elementos concretos de la cabecera del curso y que el objetivo del alumno es ir avanzando por los contenidos de este, una buena alternativa cuando se cambia dinámicamente el contenido de la página es enviar el foco al `<h1>`. De este modo, la persona avanza de manera rápida, fluida y consistente.

En este caso, también es adecuado simplificar el mensaje de cargando página a “La página se ha cargado” o “El contenido se ha cargado”, puesto que a continuación toma el foco el título de la página (el `<h1>`), por lo que el lector de pantalla ya lo va a anunciar y no es necesario repetirlo en el mensaje de estado.

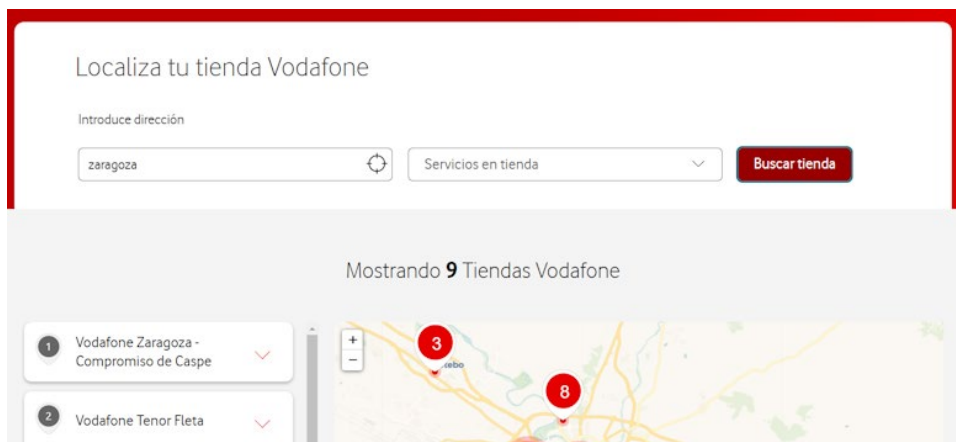
Cuando envíes el foco de teclado a un elemento ten en cuenta estos dos puntos:

- 1) Si el elemento que toma el foco **NO** es un elemento de interacción, como en este caso el `<h1>`, debe tener el atributo `tabindex="-1"`, nunca `tabindex="0"`.
Si le añades `tabindex="0"` tomará el foco al tabular por la página.
Si le añades `tabindex="-1"` solo tomará el foco por programación, pero nunca al tabular por la página, que es lo correcto.
- 2) La **indicación visual de qué elemento tiene el foco debe ser clara**. Evita limitarte a un simple cambio de color. Por ejemplo, el elemento podría estar resaltado mediante un recuadro.

Opción 3. Mantener el foco de teclado en el elemento de interacción pulsado

En otros casos, cuando se modifica dinámicamente sólo una parte del contenido, mover el foco al comienzo de la página o al título no es la mejor solución.

Imagina que tienes un buscador en la región principal de la página, de modo que solamente se modifican dinámicamente los resultados de la búsqueda, como en esta página:



Captura de pantalla 25. Buscador interior de una página que recarga dinámicamente los resultados.

En este caso, puede ser más adecuado que el **foco de teclado se quede en el botón "Buscar"**, como se muestra en el ejemplo anterior, de modo que la siguiente tabulación nos lleva al primer resultado de la búsqueda.

Por otra parte, el mensaje "Mostrando 9 tiendas" debe tener los atributos ARIA de mensaje de estado que hemos comentado anteriormente. De este modo, el lector de pantalla anuncia el mensaje, aunque no se haya movido el foco.

Recomendaciones finales

Las **tres recomendaciones claves** desarrolladas en este capítulo para hacer accesible una SPA son:

- **Informa** a la persona de lo que ocurre mediante mensajes de estado correctamente implementados con ARIA, para que sean anunciados a las personas usuarias de productos de apoyo.
- **Modifica** dinámicamente el `<title>` del `<head>` para que refleje el contenido que se está mostrando en cada caso.
- **Controla** el foco de teclado cuando se carga contenido nuevo para que sea predecible, consistente y genere la mejor experiencia de uso para las personas usuarias de teclado, que pueden o no ver la pantalla.

Otras recomendaciones a tener en cuenta a la hora de plantear la accesibilidad de una SPA son:

- Acuérdate de modificar **la miga de pan** convenientemente para que refleje en qué página estamos.
- Si marcas el **menú o paso actual**, acuérdate de marcar el correcto cuando modifiques el contenido.
- Tendrás que **programar un retardo** para que el lector de pantalla anuncie todo adecuadamente. Debemos asegurarnos de que el lector de pantalla carga el nuevo contenido en el búfer; y que le da tiempo a leer el mensaje de estado antes de leer el contenido que toma el foco, sin interrupciones ni cambios abruptos.



Sistemas de diseño accesibles

La transformación digital ha traído a las organizaciones la necesidad de crear innumerables servicios online.

Mantener la coherencia entre ellos es una tarea casi imposible sin un sistema que les ayude a reutilizar componentes de un proyecto a otros.

Si aprovechamos para hacer estos componentes accesibles, habremos dado un gran paso para construir sitios web accesibles casi sin darnos cuenta.

Qué es un Sistema de diseño

En las organizaciones grandes (y no tan grandes), es común que varios diseñadores UX/UI y desarrolladores colaboren para diferentes proyectos, y muchas veces tendrán que enfrentarse a situaciones similares: formularios, cabeceras, menús de navegación, enlaces... En ausencia de guías claras y compartidas, cada diseñador UX/UI y desarrollador se verá obligado a dedicar una considerable cantidad de tiempo a definir, decidir, diseñar y programar cada uno de los elementos de una interfaz, incluso los aparentemente simples.

Estas incertidumbres plantean desafíos a diseñadores UX/UI y desarrolladores durante la creación de productos digitales, resultando en una pérdida de tiempo en la búsqueda de estilos específicos para cada elemento o componente.

- ¿Cuánto **tiempo y esfuerzo se ha invertido** en el diseño y desarrollo de cada uno de los elementos?
- ¿Cómo de frustrados crees que se sentirán los integrantes del equipo de diseño y desarrollo? ¿Y los responsables de proyectos, al ver que el suyo **no se parece al de otros** compañeros?
- ¿Y cómo se sienten las personas que van a acceder al sitio web, que deben dedicar parte de sus recursos mentales a **aprender cómo se maneja cada interfaz**?
- ¿Es buena o **mala la imagen** que la organización está transmitiendo?

Para evitar estos problemas nacieron los sistemas de diseño, una herramienta que permite al equipo establecer patrones reutilizables para crear nuevos proyectos y extender funcionalidades. Los sistemas de diseño son mucho más que una librería de elementos compartidos por diseñadores UX/UI y desarrolladores, **es una forma de trabajo en sí, unos procesos, una documentación y un equipo, que, basándose en unos principios compartidos, permiten la escalabilidad y estandarización de los proyectos a partir de pequeñas piezas.**

Y, si estas pequeñas piezas tienen en consideración la accesibilidad, podemos decir que la creación de proyectos accesibles se convierte en algo más sencillo, ya que tendremos gran parte del camino hecho al construir las páginas con elementos ya validados.

Cómo diseñar y desarrollar un sistema de diseño accesible

Los sistemas de diseño se organizan siguiendo el modelo de diseño atómico de Brad Frost, una metodología compuesta por seis etapas, como se muestra en la Ilustración 37, de lo más pequeño a lo más grande.

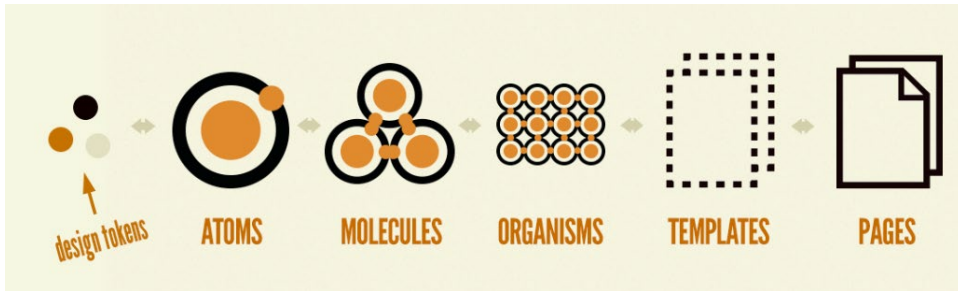


Ilustración 37 Etapas del diseño atómico. Fuente: [Extending Atomic Design | Brad Frost](#)⁸⁴

Aunque los átomos son las unidades más pequeñas de la materia, los **tokens de diseño** serían los valores específicos de las características de estos elementos, como el peso de una fuente o el valor hexadecimal de un color.

Los **átomos** son las bases sobre las que se asienta el diseño, y se correspondería con la identidad de la organización. Actuar sobre la accesibilidad de los átomos es fundamental, pues es la primera piedra en el camino. Debemos seleccionar:

- **colores** de forma y de fondo, cuya combinación debe tener suficiente contraste.
- **tipografías** que faciliten la lectura.
- **iconos** que sean explicativos y que tengan un contraste suficiente con el fondo. En desarrollo se tienen que implementar de tal manera que se pueda decidir si su texto alternativo es obligatorio o no, dependiendo del contexto en el que se use.
- un **tono de voz** sencillo y compatible con la escritura clara.
- **espaciados** suficientes para poder organizar la información con claridad y con una jerarquía evidente.

El siguiente nivel serían las **moléculas**, es decir, la unión de varios átomos que en conjunto formarían los elementos básicos de HTML. Algunas de las variables que debemos vigilar en este paso, siguiendo lo definido en las WCAG 2.2, sería:

- los **encabezados** deben ser de mayor tamaño y/o peso cuanto más importantes sean, es decir, un h1 debe ser más destacado que un h2, y subsiguientes.
- Los **textos** deben tener un tamaño suficiente para poder ser leídos y redimensionados, evitando bloques de texto largos.
- Los **enlaces**, además de ser destacados visualmente para poder distinguirlos del texto normal, deben tener en cuenta la forma de identificar su objetivo.

⁸⁴ <https://bradfrost.com/blog/post/extending-atomic-design/>

- Las **tablas de datos** accesibles suelen ser de los elementos más retadores de conseguir, sobre todo si son complejas, por las relaciones entre encabezados y celdas. Se debe tener en cuenta además cómo deben responder ante diferentes tamaños de pantalla, en especial los móviles.
- **Elementos de formulario:** campos de entrada, casillas de verificación, botones, avisos de errores... Si bien no son complicados, sí son laboriosos por las diferentes casuísticas que tienen (estado activo, inactivo, con foco, con error...). En cuanto a diseño, hay que tener en cuenta el contraste de colores, la sencillez de uso, las instrucciones de manejo, las validaciones, el autocompletar, los gestos que se deben hacer, si tienen movimiento de arrastre... En desarrollo deben cuidar las etiquetas que identifiquen el propósito de los campos, los atributos ARIA si los necesitan, cómo se comportan...
- Las **imágenes**, tanto decorativas como de contenido, cómo deben mostrarse y si deben llevar texto alternativo o no, y cómo implementarlo.

Una vez que se hayan decidido las moléculas, el siguiente paso son los **organismos**, es decir, los componentes que unen varios elementos en un único concepto, como la cabecera de una página, los menús de navegación, las migas de pan... Aquí la casuística de accesibilidad es aún mayor, ya que cada componente deberá ser evaluado por sus características propias siguiendo igualmente las WCAG 2.2. Por ejemplo, un carrusel debe poderse manejar por teclado; se debe poder parar; la estructura debe poder ser determinada por software; las transiciones no deben tener parpadeos molestos o movimientos bruscos; los cambios se tienen que anunciar por ARIA a los lectores de pantalla; las imágenes deben tener (o no) texto alternativo...

Otro ejemplo de componente complejo son los formularios. No sólo hay que tener en cuenta la complejidad de cada campo por separado, sino también cómo interactúan entre ellos, cómo se agrupan, cómo se validan, si tienen un límite de tiempo, cómo se dividen en diferentes páginas...



Si quieres comprobar los requisitos de accesibilidad propios de los carruseles, formularios y otros componentes comunes, recomendamos los [tutoriales de componentes del W3C](#).⁸⁵

El siguiente nivel sería la construcción de **plantillas**, es decir diseños prehechos y ya desarrollados listos para introducir los contenidos. Aquí debemos verificar cómo se comportan los diferentes ítems en conjunto: su jerarquía, su manejo por teclado, el orden del foco, y que éste no quede atrapado en capas que se oculten. Además, debemos comprobar que se identifica el idioma de la página, que funciona correctamente en diferentes tamaños de pantalla y orientaciones, que la ayuda se encuentra siempre en la misma posición, que el título de la plantilla describe su propósito... Es decir, lo ya visto en los criterios de las WCAG 2.2.

El último nivel serían las **páginas** que efectivamente se construyen a partir de todos los ítems creados. Si todos los pasos previos se han dado con seguridad, se habrá avanzado bastante en la accesibilidad del conjunto final, pero no se puede asegurar al 100% la misma, pues hay que comprobar que se han seguido las especificaciones dadas en la documentación y que la orquestación de los ítems es correcta. Por ello se debe comprobar, tanto con herramientas automáticas como manualmente, y, si es posible, con personas con discapacidades variadas.

⁸⁵ <https://www.w3.org/WAI/tutorials>

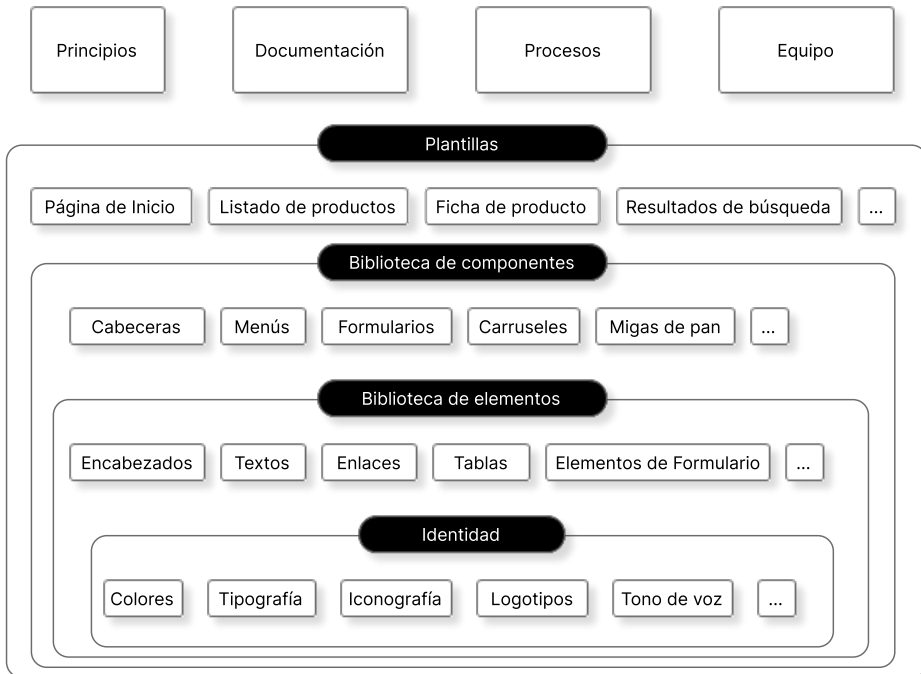


Ilustración 38 Componentes básicos de un sistema de diseño

Documentar el sistema de diseño

Los sistemas de diseño son mucho más que un kit de UI listo para utilizar en Figma, Sketch u otras herramientas de diseño de interacción. Los sistemas de diseño incluyen la documentación que explica el uso de cada ítem, tanto desde el punto de vista de diseño como de desarrollo. Su objetivo, por encima de todo, es facilitar la comunicación entre ambas partes, por lo que debe presentar la información más relevante de forma clara y fácil de entender.

Los sistemas de diseño más importantes contemplan la accesibilidad como uno de sus pilares, y hacen verdaderos esfuerzos para que así se vea reflejado en el producto final. Veámoslo cómo lo reflejan diversos sistemas de diseño a través de un elemento de uso común, como las casillas de verificación o *checkbox*.

El sistema de diseño [Material 3 de Google](#)⁸⁶ muestra información de cómo se diseña el elemento, las especificaciones técnicas, cómo se usa, y las pautas de comportamiento. Además, dedica una página a las características de accesibilidad de este elemento: cómo se utiliza con tecnología de apoyo, cómo se opera con teclado, y qué características de código deben tenerse en cuenta para que esto suceda así.

En el sistema de diseño [Fluent 2 de Microsoft](#)⁸⁷ describe el diseño, el comportamiento, y el contenido asociado a las casillas de verificación. Con respecto de la accesibilidad, solamente incluye un párrafo explicando el uso de las etiquetas *label* en los grupos de casillas. Aunque sólo dedica un breve párrafo, el resto de la página describe las buenas prácticas que se esperan de una casilla de verificación para que sea usable y accesible: textos cortos, fácil de escanear, funcionamiento predecible, orden lógico...

En las [Human Interface Guidelines de Apple](#)⁸⁸ presentan todos los elementos de tipo selector en una misma página, y de los *checkbox* apenas presentan algunos casos de uso genérico, sin comentar nada específico de accesibilidad. Sin embargo, incorpora una amplia [documentación para desarrolladores](#)⁸⁹ y cómo incluir las características de accesibilidad en general.

Otra forma de aplicar la accesibilidad en un sistema de diseño es la forma en la que lo hace el sistema de diseño [DESY del Gobierno de Aragón](#)⁹⁰, el primero publicado en abierto de una administración pública en España. Como producto de un organismo público de la Unión Europea, DESY está obligado a cumplir la norma EN 301 549, basada en las WCAG 2.1. Para ello, DESY integra numerosa casuística, comentada en lenguaje claro e incluye un apartado de buenas prácticas para cada elemento.

En el último ejemplo, el sistema de diseño [Dintel del Gobierno de España](#)⁹¹ indica que cumple los criterios necesarios según las WCAG 2.1, aunque los cambios de contenido y estilos pueden afectar a la accesibilidad. También ofrece un listado de comprobaciones muy útil para asegurarnos de que el elemento cumple los criterios de éxito asociados al mismo.

Finalmente, los sitios web que dan soporte documental a DESY y Dintel deben cumplir con la normativa pública de accesibilidad, por lo que son un gran campo de pruebas para sus propios elementos y componentes.

⁸⁶ <https://m3.material.io/components/checkbox/accessibility>

⁸⁷ <https://fluent2.microsoft.design/components/web/react/checkbox/usage>

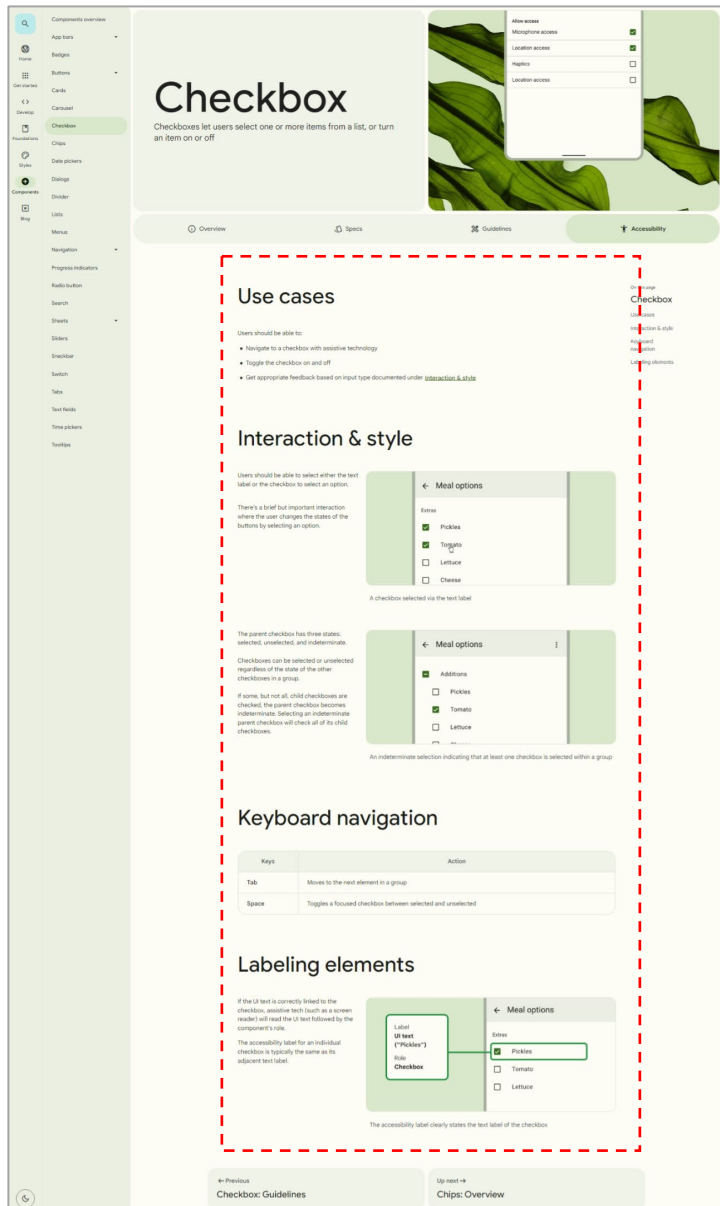
⁸⁸ <https://developer.apple.com/design/human-interface-guidelines/toggles>

⁸⁹ <https://developer.apple.com/documentation/Accessibility>

⁹⁰ <https://desy.aragon.es/componente-checkboxes.html>

⁹¹ <https://dintel.redsara.es/DINTEL/elementos-de-interfaz/componentes/checkbox.html>

Como vemos, cada organización mantiene una documentación diferente adecuada a sus necesidades, con mayor o menor detalle, y con mayor o menor foco en la accesibilidad. Lo importante es que el ítem compartido esté validado y que, al propagarse por diferentes páginas, siga siendo igual de accesible que cuando se concibió en el sistema. Además, si se hace una mejora de accesibilidad sobre el ítem, esta mejora se propagará a las páginas donde ese ítem se haya usado.



Captura de pantalla 26 Casilla de verificación en el sistema de diseño Material 3 de Google

Fluent 2 Search (7) Go to Fluent 1

- Get started
- Design language
- Guidelines
 - Web
 - Components
 - Overview
 - Accordion
 - Avatar
 - Avatar group
 - Badge
 - Button
 - Card
 - Checkbox**
 - Combobox
 - Dialog
 - Divider
 - Dropdown
 - Field
 - Fluent provider
 - Icon
 - Image
 - Info button
 - Input
 - Label
 - Link
 - Menu
 - Persona
 - Popover
 - Progress bar
 - Radio group
 - Select
 - Skeleton
 - Slider
 - Spin button

On this page

- Resources
- Layout
- Behavior
- Accessibility
- Content

Email me

- Daily updates
 - Updates from my groups
 - New followers
 - Likes on my posts
 - Messages in my inbox
- Weekly updates
 - Highlights from my organization
 - Highlights from my groups
 - Suggestions for people to follow

Indeterminate state

If one checkbox can control a subset of options, selecting or clearing the parent checkbox selects or clears all of the children. Use an indeterminate state to show when some, but not all, of the options are selected.

Create a visual separation between parent and child checkboxes to increase scannability. For example, indenting the children makes it easy to infer what will happen if someone selects the parent.

Accessibility

Checkbox groups should be associated with a descriptive `label` component. Do this by rendering a separate `Label` and either nesting the checkbox within it or using `for` and `id` attributes to associate the elements.

Passing the `label` prop within the `checkbox` provides a text label for single checkboxes.

Content

1 Preferred devices

- Surface Pro
- Surface Studio
- Surface Laptop Studio

2

1. **Group label** (optional) describes a group of checkboxes.
 2. **Checkbox label** (required) clearly communicates what will happen if the checkbox is selected.

Captura de pantalla 27 Casilla de verificación en el sistema de diseño Fluent 2 de Microsoft

Design Overview What's new Guidelines Resources

Filter

- > Platforms
- > Foundations
- > Patterns
- > Components
 - > Content
 - > Layout and organization
 - > Menus and actions
 - > Navigation and search
 - > Presentation
 - > Selection and input
 - Color wells
 - Combo boxes
 - Digit entry views
 - Image wells
 - Pickers
 - Segmented controls
 - Sliders
 - Steppers
 - Text fields
 - Toggles**
 - Virtual keyboards
 - > Status
 - > System experiences
 - > Inputs
 - Action button
 - Apple Pencil and Scribble
 - Digital Crown
 - Eyes
 - Focus and selection
 - Game controllers
 - Gestures
 - Gyroscope and accelerometer
 - Keyboards
 - Nearby interactions
 - Pointing devices
 - Remotes
 - Touch Bar
 - > Technologies

particular, avoid using these components in a toolbar or status bar.

Switches

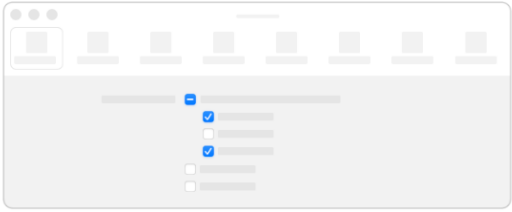
Avoid using a switch to control a single detail or a minor setting. A switch has more visual weight than a checkbox, so it looks better when it controls more functionality than a checkbox typically does. For example, you might use a switch to let people turn on or off a group of settings, instead of just one setting.

In general, don't replace a checkbox with a switch. If you're already using a checkbox in your interface, it's probably best to keep using it.

Checkboxes

A checkbox is a small square button that's empty when the button is off, contains a checkmark when the button is on, and can contain a dash when the button's state is mixed. Unless a checkbox appears in a checklist, a descriptive label follows the button.

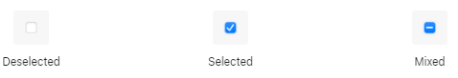
Use a checkbox instead of a switch if you need to present a hierarchy of settings. The visual style of checkboxes helps them align well and communicate grouping. By using alignment — generally along the leading edge of the checkboxes — and indentation, you can show dependencies, such as when the state of a checkbox governs the state of subordinate checkboxes.



Consider using radio buttons if you need to present a set of more than two mutually exclusive options. When people need to choose from options in addition to just "on" or "off," using multiple radio buttons can help you clarify each option with a unique label.

Consider using a label to introduce a group of checkboxes if their relationship isn't clear. Describe the set of options, and align the label's baseline with the first checkbox in the group.

Accurately reflect a checkbox's state in its appearance. A checkbox can be in a selected, deselected, or mixed state. If you use a checkbox to globally turn on and off multiple subordinate checkboxes, show a mixed state when the subordinate checkboxes have different states. For example, you might need to present a text-style setting that turns all styles on or off, but also lets people choose a subset of individual style settings like bold, italic, or underline. For developer guidance, see [allowsMixedState](#).



Radio buttons

A radio button is a small, circular button followed by a label. Typically displayed in groups of two to five, radio buttons present a set of mutually exclusive choices.

A radio button's state is either on (a filled circle) or off (an empty circle). Although a radio button can also display a mixed state (indicated by a dash), this state is rarely useful because you can

Supported platforms

- Toggles
- Best practices
- Platform considerations
- Resources
- Change log

Captura de pantalla 28 Casilla de verificación en el sistema de diseño *Human Interface Guidelines* de Apple

Componentes principales

- Botones
- Cabeceras
- Pie de página
- Formularios
- Entrada de texto
- Área de texto
- Selector
- Carga de archivos
- Botón de radio

Casilla de verificación

- Bloques de datos
- Árbol
- Navegación
- Barra de búsqueda
- Migas de pan
- Paginación
- Listado de enlaces
- Menú horizontal
- Menú vertical
- Menú de navegación
- Cabeceras
- Pie de página
- Mostrar y ocultar

- Detalles
- Acordeón
- Acordeón histórico
- Interruptor
- Pestañas
- Desplegable
- Barra de menús
- Globo de ayuda

- Datos
- Píldoras
- Listado de descripciones
- Item
- Item de estado
- Tabla

- Avisos
- Resumen de errores
- Notificaciones
- Modal

- Información visual
- Card
- Iconos
- Imágenes
- Spinner

Casilla de verificación

Este elemento ofrece la posibilidad de seleccionar varias opciones de las que se ofrecen.

Uso `Código.Nunjucks.y.HTML` `Código.Angular`

Anatomía

Se componen de una caja, un elemento cuadrado, y de una etiqueta con el nombre de la opción. La alineación puede ser tanto en vertical como en horizontal, siendo preferible siempre que sea posible la alineación vertical.

Tipos

Por defecto	Condicional	Con líneas divisorias
<input type="checkbox"/> Item 1	<input type="checkbox"/> Item 1	<input type="checkbox"/> Item 1 Lorem ipsum dolor sit amet
<input type="checkbox"/> Item 2	<input type="checkbox"/> Item 2	<input type="checkbox"/> Item 2 Lorem ipsum dolor sit amet
<input type="checkbox"/> Item 3	<input type="checkbox"/> Item 3	<input type="checkbox"/> Item 3 Lorem ipsum dolor sit amet

Por defecto

Es el que se recomienda usar en primera instancia.

Condicional

Se usa en los casos en los que hay contenido adicional, por ejemplo, si al seleccionar la opción, el usuario debe introducir algún input. En este caso, debajo del radio se introduce una línea vertical que abraza el contenido adicional.

Con líneas divisorias

En general, no se recomienda su uso. Se emplea cuando el contenido es extenso y se requiere hacer una división visual de los elementos.

Tamaños

Tamaño base	Tamaño pequeño
<input type="checkbox"/> Item 1	<input type="checkbox"/> Item 1
<input type="checkbox"/> Item 2	<input type="checkbox"/> Item 2
<input type="checkbox"/> Item 3	<input type="checkbox"/> Item 3

Tamaño base

Utilizar por defecto para cualquier tipo de situaciones y layouts. Siempre que sea posible se recomienda usar este tamaño por motivos de accesibilidad.

Tamaño pequeño

Se recomienda su uso cuando el elemento estándar sea demasiado grande para el espacio donde debe de encajar, por ejemplo dentro de tablas, cards, grupos de filtros o similares.

Buenas prácticas

- Por defecto, todas las casillas de verificación deben mostrarse sin selección o "vacías".
- Liste las opciones en un orden racional que tenga sentido para el usuario.
- Si el conjunto de opciones es mayor a 5, es preferible emplear un elemento de selección o desplegable.
- El input de entrada no debe limitarse a la caja, sino que también debe poderse seleccionar la opción pulsando en la etiqueta.

Captura de pantalla 29 Casilla de verificación en el sistema de diseño DESY del Gobierno de Aragón

Español ▾

DINTEL
SISTEMA DE DISEÑO
DISEÑO NORMALIZADO DE TRÁMITE ELECTRÓNICOS

Inicio Primeros pasos Fundamentos Elementos de interfaz Actualizaciones Contacto

Elementos de interfaz > Componentes > Casilla de verificación

Casilla de verificación

Las casillas de verificación o checkbox se utilizan cuando hay opción de seleccionar varios elementos de una lista. Los usuarios pueden seleccionar cero, uno o cualquier número de elementos.

Diseño

Vista previa [Ejemplo](#)

Label *
 Option 1 Option 2 Option 3 Option 4 Option 5 Option 6

Usos

Cuándo usar

- En formularios o página completa, modales o paneles laterales.
- En filtros de página, menú o dentro de un componente.
- En tablas de datos para edición por lotes.
- En páginas de términos y condiciones para indicar si está de acuerdo con los términos.
- En el recordatorio de contraseñas para el login.
- En listas con subselecciones.

Cuándo no usar

- Si sólo puede seleccionarse una opción de la lista, debe utilizarse el botón de opción para que el usuario pueda seleccionar sólo una opción.

Comportamiento

Tipos de casillas de verificación

Label * <input type="checkbox"/> Option 1 <input checked="" type="checkbox"/> Option 2 Simple	Label * <input type="checkbox"/> Option 01 <input checked="" type="checkbox"/> Option 02 Button
---	---

Orientación

Label * <input type="checkbox"/> Option 1 <input checked="" type="checkbox"/> Option 2 Horizontal	Label * <input type="checkbox"/> Option 01 <input checked="" type="checkbox"/> Option 02 Vertical
---	---

Tipos de casillas de verificación en formularios

<input type="checkbox"/> <input checked="" type="checkbox"/> Sin texto	<input type="checkbox"/> Option 1 <input checked="" type="checkbox"/> Option 2 Con texto
---	---

Estados

<input type="checkbox"/> Option 1 Default	<input type="checkbox"/> Option 1 Hover
<input checked="" type="checkbox"/> Option 1 Focus	<input checked="" type="checkbox"/> Option 1 Select
<input checked="" type="checkbox"/> Option 1 Select All	<input type="checkbox"/> Option 1 Disabled
<input checked="" type="checkbox"/> Option 1 Select Disabled	<input type="checkbox"/> Option 1 Select All Disabled
<input type="checkbox"/> Option 1 Error detail	Error

Accesibilidad

Este componente ha sido validado para cumplir con los requisitos de accesibilidad la UNE-EN 301549:2022 en la que se engloban los criterios de la WCAG 2.1 (AA), sin embargo, los cambios del contenido y estilos pueden afectar al cumplimiento de la accesibilidad. Debemos de asegurarnos de revisar y seguir las directrices de esta sección cuando actualice o añada nuevo contenido o estilo a este componente.

- La propiedad label del checkbox tiene role="checkbox".
- El atributo aria-label está establecido en el elemento con role="checkbox".
- Cuando está marcado, el elemento checkbox tiene el estado aria-checked establecido en true. Mientras que cuando no lo está, se establece en false.
- Si el checkbox o grupo de checkboxes incluye texto estático descriptivo adicional relevante, tiene la propiedad aria-describedby establecido en el ID del elemento que contiene la descripción. Se usa para indicar la identificación de un componente que contiene texto descriptivo y ayuda relacionado con el checkbox.
- La propiedad disabled hace que el componente checkbox esté deshabilitado. Esto evita que los usuarios puedan interactuar con el checkbox y transmite su estado inactivo a las tecnologías de asistencia.

Captura de pantalla 30 Casilla de verificación en el sistema de diseño Dintel del Gobierno de España



Cuidar la escritura

La pauta 3.1 de las WCAG 2.2 muestra cómo debemos hacer los contenidos fáciles de leer y comprender.

En este capítulo damos un paso más y explicamos tres formas diferentes y complementarias de ampliar la calidad de la información, la participación ciudadana y el respeto a la diversidad.

En primer lugar, explicamos los conceptos de comunicación y lenguaje claro, así como las claves para conseguirlo.

A continuación, mostramos un paso más allá, la lectura fácil o lenguaje fácil, una forma de llegar a casi todas las personas.

Por último, mostramos el poder de las palabras para crear una sociedad mejor a través del lenguaje inclusivo.

La importancia de las palabras

A menudo, tenemos que releer un texto porque resulta difícil de comprender. Puede que no le encontremos sentido, que incluya palabras que no conocemos o que esté construido de manera muy compleja.

Este problema afecta a diario a las personas que se tienen que enfrentar a textos jurídicos, financieros, administrativos o médicos sin ser profesionales de estos campos. Esta dificultad es mayor para las personas con discapacidad intelectual, del desarrollo o auditiva; para las personas con bajo nivel de alfabetización; para las personas que están aprendiendo el idioma; o para niños y niñas que no comprenden conceptos complejos, entre otros perfiles.

Como reacción a este problema han surgido dos movimientos en defensa de la accesibilidad cognitiva a través del lenguaje claro y la lectura fácil, que, si bien comparten origen, son conceptos diferenciados.

El **lenguaje claro** es parte de un movimiento más grande, la **comunicación clara**. Ambos métodos ponen a las personas que van a leer el texto en primer lugar considerando:

- lo que las personas quieren y necesitan saber;
- el nivel de interés, experiencia y habilidades de alfabetización de las personas;
- el contexto en el que las personas utilizarán el documento.

La **lectura fácil** (también llamada **lenguaje fácil**) es un método para hacer la información más fácil de entender para la mayoría de las personas, incluidas las personas con discapacidad intelectual o del desarrollo.

Es decir, el lenguaje claro se puede utilizar para una audiencia general, mientras que la lectura fácil se utiliza para personas que tienen dificultades con la comprensión lectora.

Por otro lado, el **lenguaje inclusivo** es una forma de expresión que busca evitar la discriminación o la exclusión de personas por sus características como consecuencia de la forma en que se usa el lenguaje. El lenguaje inclusivo pretende reflejar la diversidad y la igualdad de la sociedad, así como promover el respeto y la visibilidad de los grupos minoritarios o vulnerables. El lenguaje inclusivo es compatible tanto con el lenguaje claro como con la lectura fácil.

La comunicación clara

En la [Guía de comunicación clara](#)⁹², Estrella Montolío y Mario Tascón definen la comunicación clara como “transmitir de forma fácil, directa, transparente, simple y eficaz información relevante para la ciudadanía por cualquiera de los diferentes canales [...] y adaptada a sus particularidades”.

Dentro de este modelo, el lenguaje claro es un componente, pero no el único. El diseño, el neurolenguaje y el ámbito digital tienen también una gran participación. En la intersección de estos campos aparece un nuevo factor, la inteligencia artificial, que aprende de la interacción con las personas y conversa en nuestro propio registro.

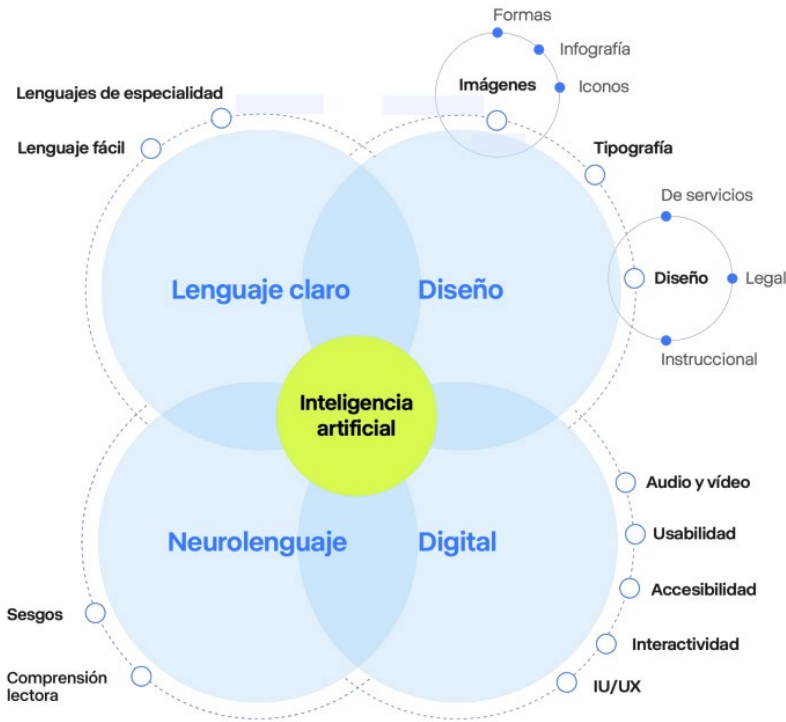


Ilustración 39 Componentes de la comunicación clara (Fuente: Guía de comunicación clara)

Para practicar una comunicación clara, Montolío y Tascón identifican 9 pasos en los que las personas deben estar en el centro del proceso y ser el foco de atención.

⁹² <https://comunicacionclara.com/docs/guia-comunicacion-clara-prodigioso-volcan.pdf>

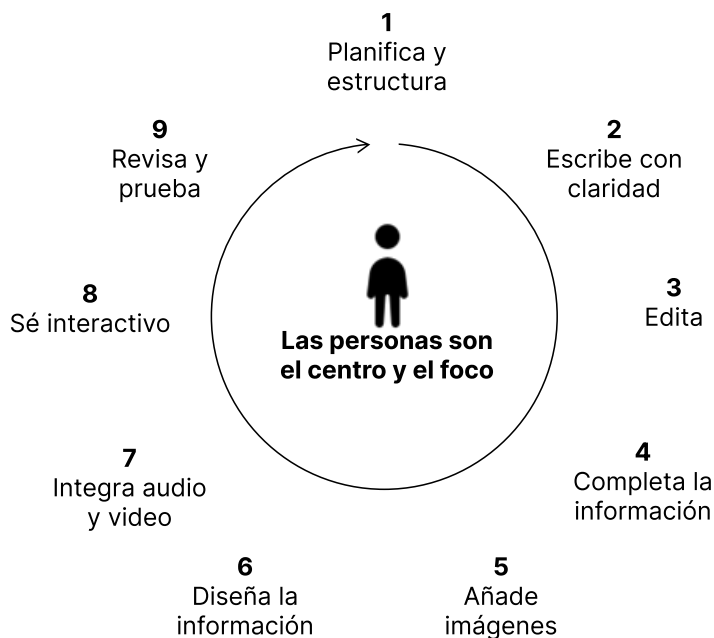


Ilustración 40 Pasos de la comunicación clara según la Guía de comunicación clara

- **Paso 1. Planifica y estructura.** En este primer momento debemos responder a las preguntas: ¿Qué queremos contar? ¿Qué elementos debe incluir esa información? ¿Cuántas partes va a tener? ¿Cómo se van a organizar las partes?
- **Paso 2. Escribe con claridad.** Escribir es difícil porque requiere trabajar con las ideas tanto como con las palabras.
- **Paso 3. Edita.** Es el momento de usar los recursos tipográficos a nuestra disposición para destacar elementos: mayúsculas, negritas, cursivas, subrayados... Siempre con mesura para que el texto no parezca un catálogo de palabras luchando por destacar.
- **Paso 4. Completa la información.** Utiliza ejemplos, notas al pie, tablas de contenidos o índices...
- **Paso 5. Añade imágenes.** Iconos, gráficos, ilustraciones o fotografías que refuercen el mensaje y hagan el contenido más atractivo.
- **Paso 6. Diseña la información.** Es el momento de que nuestra información luzca limpia y ordenada, legible y jerarquizada. La composición, el uso de los espacios, el ancho de los párrafos, o el color son algunas de las herramientas que usaremos para dar una impresión de simplicidad en el diseño.
- **Paso 7. Integra audio y video.** En la era audiovisual, complementar la información textual con alguno de estos recursos mejorará su atractivo, la usabilidad y accesibilidad.
- **Paso 8. Sé interactivo.** En la web no existe la linealidad que se le puede presumir a un libro, sino que las personas entran y salen sin control, por lo que nuestros textos deben atraparles para que decidan leerlos.
- **Paso 9. Revisa y prueba.** Ya sea con tests A/B, con test de usuarios, o grupos de enfoque, lo importante es conocer de primera mano las opiniones de las personas e integrar su *feedback* si se considera apropiado.

Redactar en lenguaje claro

La Organización Internacional de Estándares (ISO) ha publicado un estándar específico para el lenguaje claro, el [ISO 24495-1:2023](https://www.iso.org/obp/ui/en/#iso:std:iso:24495:-1:ed-1:v1:en)⁹³ (en inglés y francés). Sus recomendaciones se dividen en cuatro categorías:

1. El texto es **relevante** para lo que la persona necesita. Para ello es necesario identificar a la persona que va a leer el texto, sus objetivos y su contexto. También es necesario seleccionar el tipo o tipos de documentos que le vamos a presentar y el contenido que va a necesitar.
2. La persona puede **encontrar fácilmente** lo que necesita. Para ello hay que trabajar la estructura del documento; utilizar técnicas de diseño de la información; incluir encabezados que anticipen el contenido de los bloques de texto; y separar la información importante de la información suplementaria.
3. La persona puede **entender fácilmente** lo que encuentra. Para ello se deben emplear palabras familiares en frases y párrafos claros y concisos. El tono de voz debe ser respetuoso y se debe comprobar que el documento final es coherente. Además, se debe considerar el uso de imágenes y multimedia que complementen el contenido del texto.
4. La persona puede **usar fácilmente** la información. En este punto es necesario evaluar que lo escrito cumple las recomendaciones anteriores.

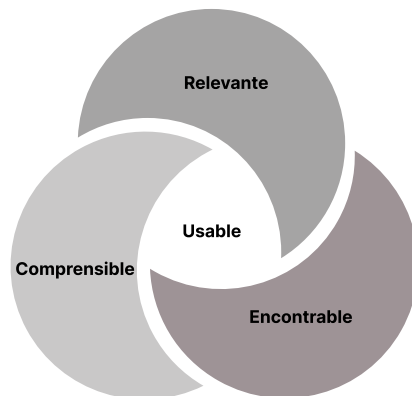


Ilustración 41 Categorías de recomendaciones de lenguaje claro según el ISO 24495-1:2023

De forma más concreta, la *Guía de comunicación clara* da los siguientes consejos más precisos para escribir de forma clara:

- Utilizar ejemplos, comparaciones, analogías, metáforas o ser concretos son algunas de las técnicas que podemos usar al escribir.
- Cuando escribas para una pantalla, sintetiza todo lo que puedas. Piensa que las personas usan sobre todo el móvil para acceder a la información.
- Controla la longitud de párrafos y oraciones.

⁹³ <https://www.iso.org/obp/ui/en/#iso:std:iso:24495:-1:ed-1:v1:en>

- Sigue el orden natural de las frases.
- Utiliza la voz activa en lugar de la pasiva.
- No abuses de los gerundios, infinitivos y participios.
- Conecta las ideas.
- Enumera con listas ordenadas.
- Escoge las palabras más adecuadas.
- Evita términos técnicos.
- Vigila el uso de abreviaturas y siglas.
- Evita la acumulación de elementos de negación.
- Respeta las normas de la lengua.
- Divide el texto en apartados y bloques.
- Introduce títulos explicativos antes de los apartados.
- Asegura una relación coherente entre el texto escrito y las imágenes.

Ejemplo de mejora comunicativa

El [Ayuntamiento de la ciudad de Madrid está revisando la información](#)⁹⁴ de sus portales institucionales y comunicaciones con los ciudadanos para aplicar un lenguaje más claro, conciso y comprensible para la ciudadanía.

A continuación, mostramos dos versiones de la misma información. Se trata de una carta que se envía a los ciudadanos explicándoles las características del pago de impuestos.

¿Eres capaz de identificar los cambios en el texto? ¿Y en el diseño de la carta?

ANTES⁹⁵

Estimado/a Sr/Sra:

Figura Vd. Como titular del Sistema de Pago a la Carta (PAC), inscripción nº 1621529500217, para el ejercicio 2017, según los datos que constan en nuestra base de datos.

Con esta comunicación ponemos en su conocimiento la situación de los recibos adheridos al mismo, con indicación de los que han sido abonados en su totalidad y los que restan por pagar, una vez aplicados los pagos a cuenta realizados por Vd. Y las bonificaciones correspondientes.

Se envían los justificantes de pago de los recibos que han resultado totalmente abonados. Los recibos que quedan pendientes de pago, ya sea en su totalidad o parcialmente, le serán remitidos a su banco o caja de ahorros, para que sean cargados en su cuenta el 15 de diciembre, siendo la entidad bancaria la que le remitirá el justificante de su pago. Le enviamos las imágenes de los recibos pendientes con todos los datos y le recomendamos que conserve el presente documento para que pueda completar los datos que le remitirá su entidad financiera.

Próximamente recibirá información sobre el calendario de pagos para el ejercicio 2018.
Atentamente,

XXX

DEVOLUCIÓN DE IMPORTES NO APLICADOS El artículo 55 de la Ordenanza Fiscal General de Gestión, Recaudación e Inspección determina que en el supuesto de que los pagos anticipados sean superiores a la suma de los importes de las liquidaciones adheridas al PAC, se procederá a la devolución de oficio mediante transferencia bancaria a la misma cuenta en la que se efectuaron los cargos.

⁹⁴ <https://www.madrid.es/portales/munimadrid/es/Inicio/Actualidad/Comunicacion-Clara/?vgnextfmt=default&vgnextoid=a01f1905bacde510VgnVCM1000001d4a900aRCRD&vgnnextchannel=59af566813946010VgnVCM100000dc0ca8c0RCRD&idCapitulo=10559854>

⁹⁵ Original completo disponible en https://www.madrid.es/UnidadesDescentralizadas/Calidad/LenguajeClaro/ComunicacionClara/Ficheros/ModelocartaliqPAC_Inicial_Web.pdf

DESPUÉS⁹⁶

Desde la Agencia Tributaria del Ayuntamiento de Madrid le informamos sobre la situación de su Pago a la Carta 2018.

Como recordará, eligió esta opción de pago, domiciliando los recibos en su banco. En este envío le explicamos en detalle:

- > Cuánto ha pagado
- > Cuánto se ha ahorrado
- > Cuánto le queda por pagar

Además, incluimos la información de los recibos pendientes y los justificantes de pago de los recibos abonados en su totalidad mediante esta opción. Le aconsejamos que guarde toda esta información junto con los justificantes de pago que le enviará el banco. En breve recibirá el calendario de pagos para el próximo año.

Atentamente,

XXXXX

Así hacemos el cálculo> Para poder ofrecerle el Pago a la Carta, hacemos el cálculo de la cuota periódica a pagar tomando como referencia las cantidades abonadas el año pasado. En caso de que la cuota de este año sea menor, no se preocupe: le haremos la devolución mediante transferencia a la cuenta bancaria en la que ha domiciliado los pagos.

2018 MADRID
Madrid, noviembre de 2017

Estimado Sr/Sra:

Figura 16 como titular del Sistema de Pago a la Carta (PAC), inscripto nº 16215250217 para el ejercicio 2017, según los datos que constan en nuestra base de datos.

Con esta comunicación le informo en su conocimiento de la situación de los recibos adscritos al mismo, con respecto a los que han sido abonados en su totalidad y los que están por pagar, una vez aplicados los pagos a cuenta realizados por Ud. y las bonificaciones correspondientes.

Se envían los justificantes de pago de los recibos que han sido totalmente abonados. Los recibos que quedan por pagar, ya sea en su totalidad o parcialmente, le serán enviados a su banco o, caso de no tener, para que sean cargados en su cuenta el 16 de diciembre, a favor de la entidad bancaria a la que le remití el justificante de su pago. Le enviamos los imágenes de los recibos pendientes con todos los datos y le recomiendo que conserve el presente documento por si que pueda completar los datos que le remití su entidad financiera.

Por último, recibirá información sobre el calendario de pagos para el ejercicio 2018.

Atentamente,

LA SUBDIRECTORA GENERAL DE RECAUDACIÓN

Ana María Hernández Mateo

REPÚBLICA ESPAÑOLA DE ESPAÑA
AYUNTAMIENTO DE MADRID



Información sobre la situación de su Pago a la Carta 2018

Madrid, noviembre de 2017

Atención: APELLIDOS, NOMBRE y DNI MADRID

Madrid, noviembre de 2017

Información

Teléfono: 91 529 82 10 (de 9 a 19 h) o 91 529 82 11 (de 9 a 19 h) o 24 h en los días festivos.

Cuanto le queda por pagar

Cuanto se ha abonado

Cuanto le queda por pagar

Para poder ofrecerle el Pago a la Carta, hacemos el cálculo de la cuota periódica a pagar tomando como referencia las cantidades abonadas el año pasado. En caso de que la cuota de este año sea menor, no se preocupe: le haremos la devolución mediante transferencia a la cuenta bancaria en la que ha domiciliado los pagos.

LA SUBDIRECTORA GENERAL DE RECAUDACIÓN

Ana María Hernández Mateo

Captura de pantalla 31 Antes y después de la comunicación

⁹⁶ Original completo disponible en https://www.madrid.es/UnidadesDescentralizadas/Calidad/LenguajeClaro/ComunicacionClara/Ficheros/ModelocartaliqPAC_Final_Web.pdf

Lectura fácil o lenguaje fácil

Tanto la ISO como su homólogo español AENOR han publicado el estándar para redactar en lectura fácil, el [ISO 23859:2023](https://www.iso.org/obp/ui/en/#iso:std:iso-iec:23859:ed-1:v1:en)⁹⁷ y la [Norma UNE 153101](https://olgacarreras.blogspot.com/2019/02/lectura-facil-pautas-y-recomendaciones.html)⁹⁸. Sus principales recomendaciones son:

Vocabulario

- Utiliza palabras sencillas y habituales, adecuadas para los lectores del documento.
- Evita palabras:
 - abstractas, muy largas o con sílabas difíciles;
 - que acaban en “mente”;
 - que exageran y terminan en “ísimo”;
 - de otro idioma a menos que sean muy conocidas;
 - abreviaturas;
 - siglas, a menos que sean muy conocidas y las expliques;
 - que no tienen un significado claro;
 - que tengan varios significados;
 - frases hechas, refranes, ironías y metáforas.
- Utiliza la misma palabra para describir el mismo objeto o concepto en todo el documento.
- Si necesitas usar palabras difíciles, utiliza glosas y glosarios para explicarlas.

Números

- Escribe los números en cifras.
- Evita:
 - los números grandes. Si tienes que usarlos, escríbelos en letra y con ejemplos de comparaciones;
 - los números ordinales;
 - las fracciones y los porcentajes. Utiliza ideas más fáciles de imaginar;
 - los números romanos. Si tienes que usarlos, explícalos;
 - escribir las horas en formato 24 horas.
- Separa los números de teléfono en números más pequeños.
- Escribe las fechas completas y con el día de la semana escrito.

⁹⁷ <https://www.iso.org/obp/ui/en/#iso:std:iso-iec:23859:ed-1:v1:en>

⁹⁸ <https://olgacarreras.blogspot.com/2019/02/lectura-facil-pautas-y-recomendaciones.html>

Signos de puntuación

- Evita:
 - escribir palabras o frases enteras con letras mayúsculas;
 - signos raros y poco habituales, como () % & ... ; “ ”;
 - “etcétera”, puedes utilizar otras palabras como “muchos más” o “y otros”.
- Utiliza los 2 puntos delante de una lista con más de 3 palabras o frases.

Frases

- Emplea frases simples y breves, cada una con una única idea. Si es necesario usar frases largas, divídelas en puntos naturales de lectura.
- Busca que cada frase ocupe una línea para facilitar la lectura.
- Asegúrate de especificar a quién te diriges, utilizando pronombres como tú, yo, ellos, usted o los nombres de las personas si los hay.
- Utiliza verbos simples y, cuando sea posible, el presente para simplificar la estructura verbal.
- Evita:
 - las explicaciones con comas dentro de una frase;
 - usar 2 o más verbos juntos;
 - los gerundios;
 - dar órdenes, si tienes que hacerlo, comprueba que está claro a quién se dirigen;
 - la voz pasiva, usa mejor la voz activa;
 - las frases negativas, usa mejor las positivas, y nunca dos frases negativas juntas.

Organizar el texto

- Proporciona toda la información necesaria sin dejar lagunas para que la persona no tenga que hacer suposiciones.
- Evita la inclusión de detalles innecesarios, manteniendo los textos concisos y evitando la longitud excesiva.
- Estructura la información de manera cronológica, utilizando términos como “antes”, “después” o “siguiente” para guiar a la persona a lo largo del texto.
- Mantén una comunicación directa con la persona siempre que sea posible.
- Agrupa información relacionada bajo los mismos temas y utiliza espacios en blanco entre las secciones para una mayor claridad.
- Emplea títulos descriptivos que indiquen el contenido del texto.
- Organiza las listas utilizando viñetas para facilitar la lectura.
- Presenta los diálogos en formato teatral, utilizando guiones y asegurándote de identificar claramente quién está hablando.

Presentar documentos

- Organiza el contenido de los documentos mediante apartados.
- Procura que las líneas finalicen aproximadamente en el mismo sitio.
- Establece un interlineado de al menos 1,5 y utiliza márgenes amplios.
- Opta por un tipo de letra sin adornos (sin serifas) y un tamaño mínimo de Arial 14, ajustándolo según la tipografía elegida.
- Elige un color de letra que contraste adecuadamente con el fondo de la página.

- Indica claramente cuando el texto continúa en la siguiente página.
- Numera las páginas, utilizando un tamaño mayor que el texto y colocándolo de manera consistente en cada página.
- Distingue los títulos del resto del texto, ya sea mediante tamaños o colores diferentes.
- Resalta las palabras explicadas en glosas utilizando negrita.
- Si hay secciones que no siguen el estilo de Lectura Fácil, adviértelo.
- Decide la orientación de las páginas (horizontal o vertical) y mantén la consistencia en todo el documento.
- Evita:
 - dividir todo el documento en apartados muy pequeños;
 - escribir en vertical;
 - las letras con adornos, en cursiva, subrayadas, con sombras, o que no se leen bien;
 - escribir sobre imágenes y sobre fondos con varios colores y tonos;
 - justificar el texto y escribir en columnas. Es mejor alinear las frases a la izquierda que en el centro o a la derecha de la página;
 - separar una palabra con un guion entre 2 líneas.

Imágenes

- Selecciona imágenes nítidas y específicas, evitando la saturación de elementos. Asegúrate de que tengan buena calidad y, preferiblemente, úsalas en color.
- Ubica las imágenes sobre o a la izquierda del texto al que acompañan, manteniéndolas cercanas pero sin obstaculizar la lectura. Evita situarlas detrás del texto.
- Utiliza la misma imagen a lo largo del documento para representar una idea consistente.
- Omite información sobre derechos de uso junto a las imágenes. En caso de que la imagen requiera explicación, facilita esta información en formato de Lectura Fácil.

Gráficos

- Incluye un título o explicación que aclare el contenido del gráfico.
- Agrega una leyenda que indique cómo interpretar la información del gráfico.
- Limita la explicación en un gráfico a solo dos elementos; si hay más información, considera la posibilidad de crear gráficos adicionales.

Mapas y planos

- Incluye un título o una explicación en el mapa para orientar sobre su contenido.
- Destaca itinerarios y caminos de forma clara.
- Indica claramente los elementos clave que la persona debe identificar, incluyendo lugares importantes; puedes emplear fotografías reales de espacios y elementos reconocibles.
- Añade una leyenda que facilite la interpretación del mapa.

El índice

- Coloca el índice al principio del documento.
- Deja claro en qué página está cada apartado. Puedes utilizar rayas.

Glosas o explicaciones al margen

- Emplea glosas para aclarar palabras complicadas que son esenciales en el documento.
- Introduce la glosa la primera vez que la palabra difícil aparezca en el texto.
- Coloca la glosa lo más cerca posible de la palabra difícil que estás explicando, asegurándote de que ambas estén en la misma página.
- Resalta la palabra complicada en negrita tanto en el texto como en la glosa, diferenciando claramente la palabra de su explicación. Puedes presentar la palabra en una línea y la explicación en líneas separadas.

Glosario

- Emplea un glosario para definir palabras que la persona debe comprender antes de abordar el documento. Recuerda que el glosario no es un compendio de todas las explicaciones en el texto.
- Sitúa el glosario al inicio del documento.
- Evita la palabra “glosario” si resulta complicada; en su lugar, opta por títulos como “Diccionario” o “Palabras clave”.
- Organiza las palabras del glosario en orden alfabético.
- Distingue claramente la palabra de su definición, posiblemente utilizando negrita y un diseño coherente.
- Ilustra las palabras difíciles con ejemplos para facilitar su comprensión

Ejercicios y actividades

- Agrega actividades de memoria y ejercicios para abordar el contenido.

Resúmenes y repeticiones

- Utiliza resúmenes en documentos extensos y repite la información clave cuando sea necesario.
- Coloca los resúmenes al final de la sección correspondiente.
- Facilita la diferenciación entre el texto principal y el resumen mediante el uso de colores, formas u otros elementos visuales.

Ejemplo de lectura fácil

Observa la siguiente noticia publicada por la ONG Inclusion-Europe escrita en lectura fácil. Inclusion-Europe es una asociación cuyo fin es la igualdad de derechos y la plena inclusión de personas con discapacidad intelectual y sus familias en todos los aspectos de la sociedad.



 **La comisaria Helena Dalli habla en la conferencia Hacia la Inclusión**

Helena Dalli es una **Comisaria** en la **Comisión Europea**.

Ella fue una de las parlantes en la conferencia sobre **desinstitucionalización**.

La conferencia fue organizada por **EEG**.

 La Comisaria Helena Dalli dijo que los **centros** son muy malos para las personas con **discapacidad intelectual**.

Ella dijo que las personas con discapacidad intelectual tienen una vida mejor cuando **viven de forma independiente** y en una comunidad.

 Helena Dalli también dijo que pronto habrá una nueva **Estrategia Europea sobre Discapacidad**.



página 25

Captura de pantalla 32 Noticia escrita en lectura fácil (Fuente: inclusion-europe.eu)

La noticia relata la intervención de una persona en una conferencia y cubre los principales interrogantes de una noticia: qué pasó, quién estuvo involucrado, y por qué es relevante para la persona que lee la noticia.

En la noticia se encuentran muchas de las características propias del estilo de "Lectura fácil":

- Texto:
 - Palabras sencillas, sin tecnicismos.
 - El estilo es muy sencillo: sujeto, verbo y complementos.
 - Frases cortas, distribuidas en varias líneas para facilitar el escaneado.
 - No se usan abreviaturas.
 - No se usan signos de puntuación poco usuales, solo el punto al final de cada oración.
 - No se dan cifras.
 - Se utiliza la voz activa, no la pasiva.
- Diseño y maquetación:
 - Dibujos sencillos en los márgenes que refuerzan los contenidos más complicados del texto.
 - Las palabras importantes se resaltan en negrita.
 - Tamaño de las letras grandes.
 - El tipo de letra, sin serifa.
 - Márgenes amplios.
 - Una única noticia por página.
 - Interlineado generoso.
 - Alineado a la izquierda.
 - Fondo de color plano, sin dibujos.
 - Alto contraste de los colores del texto y del fondo.
 - Se identifica claramente el número de la página en la que se encuentra.
 - Tamaño de la página coincide con un folio

Además, la foto de la protagonista está en primer plano, reduciendo el ruido visual alrededor y facilitando su identificación. No se usan fotos del público escuchando la conferencia, ni planos generales más difíciles de procesar.

Proceso de creación de un documento en lenguaje fácil

El proceso se compone de dos etapas: la fase de adaptación y la fase de validación. La norma incluye un anexo informativo que proporciona ejemplos de actividades para la fase de validación, la cual es fundamental en el proceso y recibe una gran atención en el contenido de la norma.

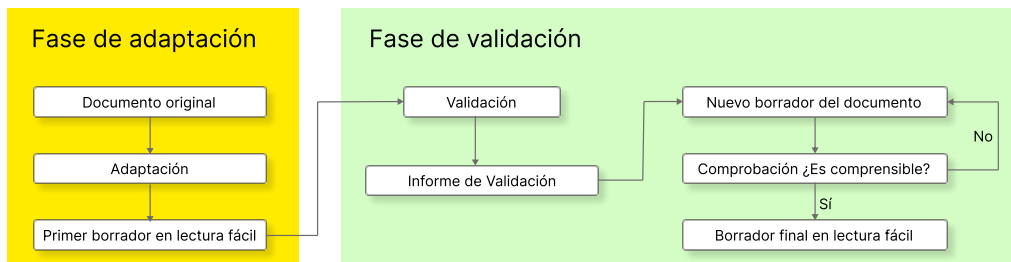


Ilustración 42 Proceso de creación de un documento en lectura fácil

En la primera fase se recoge el documento original, se adapta con la participación del diseñador y el maquetador, y se genera un primer borrador en lectura fácil.

A partir de ahí se pasa a la fase de validación, realizada por los validadores expertos y coordinada por un dinamizador. El dinamizador genera un informe de validación con las aportaciones del grupo de validadores. Estas aportaciones se integran dentro de un nuevo borrador, realizado por el adaptador, diseñador y maquetador.

Este nuevo borrador se comprueba de nuevo por los validadores, coordinados por el dinamizador y con la colaboración del adaptador, diseñador y maquetador. Esta prueba puede dar resultado negativo, y se debe volver a generar un nuevo documento con las aportaciones de la prueba.

Si el resultado es positivo, se convierte en el borrador final, listo para publicar y se puede utilizar el logo europeo de lectura fácil.



Ilustración 43 Icono de Lectura fácil

Guías de interés de lectura fácil

- [Lectura fácil: Métodos de redacción y evaluación](#) ⁹⁹ de Oscar García Muñoz.
- [Curso gratuito de lectura fácil](#) ¹⁰⁰ de la asociación "Plena Inclusión".
- [Guía "Información para todos"](#) ¹⁰¹ de la ONG Inclusion Europe.

⁹⁹ www.plenainclusion.org/sites/default/files/lectura-facil-metodos.pdf

¹⁰⁰ <https://www.plenainclusion.org/formacion/cursos/curso-de-autoformacion-lectura-facil-introduccion-a-la-adaptacion/>

¹⁰¹ https://www.inclusion-europe.eu/wp-content/uploads/2017/06/ES_Information_for_all.pdf

El lenguaje inclusivo

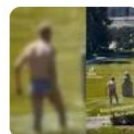
A través del lenguaje, las sociedades transmiten sus creencias, tradiciones y normas culturales de generación en generación, incluyendo el uso de palabras y expresiones cargadas de prejuicios o estereotipos que pueden contribuir a la marginalización de ciertos grupos.

Podemos distinguir lenguajes inclusivos hacia distintos colectivos, como el antirracista, el feminista, el LGTBIQ+ o el de las personas con discapacidad, en el cual nos centraremos.

Observa los titulares de estas noticias, donde se utiliza un lenguaje no inclusivo para llamar la atención. Todas ellas hacen mención a personas de una forma peyorativa.

 El Mundo

El vecino "loco" de Plaza de España que torea coches, se baña junto al Palacio Real y agrede e insulta a los transeúntes



Lv Levante-EMV

A prisión por apuñalar a un joven y dejarlo postrado en una silla de ruedas

El presunto agresor de la discoteca de Alfafar se entrega meses después del intento de homicidio.



 El Español

Juanjo Lara, el policía carnavalero muerto en Andújar por frenar a un veterinario esquizofrénico

Amante del carnaval, era integrante de la chirigota de su pueblo, Marmolejo, tenía 40 años y llevaba 15 en el Cuerpo Nacional de Policía.



EC El Comercio

Se queda embarazada de un 'Enano Boy' en su despedida de soltera

Ocurrió en Valencia, donde una mujer tuvo sexo con un enano stripper en su despedida de soltera y ahora espera un hijo del bailarín.

 El Diario

Carmen Lomana: "¿Las personas normales pagamos las subvenciones que el Gobierno presume dar a los desfavorecidos?"



Captura de pantalla 33 Titulares que utilizan términos no inclusivos

Afortunadamente, el lenguaje es una entidad en continua transformación, permeable a las diferentes sensibilidades y realidades sociales. Conscientes del poder que el lenguaje ejerce sobre la sociedad, los movimientos sociales intentan avanzar en un lenguaje que no perpetúe imágenes negativas o les estigmatice.

Sin embargo, como todo desafío hacia la norma establecida, se encuentran posiciones enfrentadas y resistencia al cambio, por lo que la evolución del lenguaje es, en ocasiones, lenta. Por ejemplo, en la Constitución Española de 1978 se introdujo la protección a los “disminuidos físicos, sensoriales y psíquicos”, un término que en la actualidad se considera [inapropiado, discriminatorio y ofensivo](#)¹⁰². Finalmente, se ha conseguido modificar [el artículo 49 de la Constitución Española en 2024](#)¹⁰³ para que se refiera a “personas con discapacidad”, adaptándose así a la Convención Internacional sobre los Derechos de las Personas con Discapacidad de 2006.

Debería evitarse usar el lenguaje de forma peyorativa, pero para ello, es necesario, por un lado, ser consciente del mal uso del lenguaje, y por otro, saber cómo usarlo de forma más apropiada. En la [Guía de Lenguaje Inclusivo](#)¹⁰⁴ publicada por la Confederación Española de Personas con Discapacidad Física y Orgánica (COCEMFE) y el Parlamento de Navarra, se exponen unas pautas para el uso de un lenguaje correcto, respetuoso y consensuado a la hora de referirse a las personas con discapacidad, dada la temática de este libro.

Se puede concluir que, las **personas con discapacidad**:

- **Ante todo, son personas:** la discapacidad es solo una característica más.
- **Son parte de la sociedad:** las personas con discapacidad son miembros de un grupo.
- **Son normales:** las personas sin discapacidad no son las normales, sólo son “personas sin discapacidad” o “el resto de la población”.
- **No son superhombres ni supermujeres,** simplemente hacen sus tareas cuando disponen de los medios adecuados y se respetan sus derechos.

Por eso, sus **recomendaciones de lenguaje** principales son:

- No uses palabras con **connotaciones negativas:** inválidos, minusválidos, retrasados...
- **Evita las descripciones negativas o sensacionalistas,** ya que generan compasión, en lugar de la “aceptación social fundada en el respeto”.
- **Evita los eufemismos** cuando hables de discapacidad, ya que están cargados de condescendencia, generan confusión, inseguridad jurídica y rebajan la protección de sus derechos y libertades.

La siguiente tabla recoge términos inclusivos y su equivalente a evitar. Incluye ejemplos de la [Guía para la comunicación inclusiva](#)¹⁰⁵ de Médicos Sin Fronteras y de la [Guía de estilo sobre discapacidad para profesionales de los medios](#)¹⁰⁶ del Gobierno de España.

¹⁰² <https://sid-inico.usal.es/noticias/una-lucha-de-20-anos-para-reformar-el-articulo-49-de-la-constitucion-y-eliminar-la-palabra-disminuidos/>

¹⁰³ Nuevo texto del artículo 49 de la Constitución Española en su versión en el BOE y publicado por el CERMI en Lectura Fácil, en pictogramas y en Lengua de Signos Española: <https://cermi.es/novedad/nuevo-texto-del-articulo-49-de-la-constitucion-espanola>

¹⁰⁴ https://www.cocemfe.es/wp-content/uploads/2019/02/20181010_COCEMFE_Lenguaje_inclusivo.pdf

¹⁰⁵ https://manual.msf.mx/wp-content/uploads/2021/07/Guia-comunicacion-inclusiva_MXCA.pdf

¹⁰⁶ <https://www.consaludmental.org/publicaciones/Guia-estilo-discapacidad-medios-comunicacion.pdf>

Tabla 10 Ejemplos de lenguaje inclusivo y no inclusivo

Lenguaje inclusivo	Lenguaje no inclusivo
Persona con discapacidad	Discapacitado, inválido, minusválido, tullido, mutilado, disminuido, retrasado, cojo, loco, oligofrénico, trastornado
Tener una discapacidad / enfermedad	Padecer, sufrir, víctima de una discapacidad o enfermedad, afectado, enfermo con diversidad funcional, con otras capacidades o capacidades diferentes
Personas en situación de dependencia	Dependientes
Persona con discapacidad física	Discapacitado físico
Persona con discapacidad orgánica	Discapacitado orgánico
Persona usuaria de silla de ruedas, que utiliza silla de ruedas	Persona en /postrada en / condenada a una silla de ruedas
Persona con tetraplejía / paraplejía	Tetraplético / parapléjico
Personas con enfermedad mental / con un trastorno de salud mental / que experimentan una enfermedad mental / que utilizan los servicios de salud mental	Enfermos mentales
Ha sido diagnosticado con esquizofrenia	Es esquizofrénico
Tiene episodios psicóticos	Es psicótica / tienes brotes psicóticos
Tiene depresión	Es depresivo
Hospital para personas con enfermedad mental / unidad de salud mental	Manicomio / psiquiátrico
Ingresado / hospitalizada en	Internado / recluida en
Ingreso hospitalario	Ingreso psiquiátrico
Descompensación	Brote
Atención en salud mental y psicosocial / atención a personas con enfermedad mental	Atención psiquiátrica
Persona de talla baja o con acondroplasia	Enano
Persona ciega / Persona con discapacidad visual	Ciego, Invidente
Sistema braille	Lenguaje, lengua o idioma braille
Personas sordas, con sordera o con discapacidad auditiva	Sordo, Sordomudo
Audífonos, implante coclear	Sonotone
Lengua de signos	Lenguaje de señas, lenguaje de signos
Intérprete de lengua de signos	Traductor de lengua de signos
Personas sin discapacidad, resto de la población	Personas normales

Guías de interés de lenguaje inclusivo

En la [Guía de estilo sobre discapacidad para profesionales de los medios de comunicación](#)¹⁰⁶ y en la [Guía de estilo sobre salud mental para medios de comunicación: las palabras sí importan](#)¹⁰⁷ se muestran decenas de ejemplos de contenidos sesgados, estereotipados, sensacionalistas y despersonalizados en el tratamiento de la discapacidad, así como las alternativas inclusivas y el lenguaje correcto para tipo de discapacidad. Además, incluye un extenso e interesante capítulo sobre la evolución de la discapacidad en la sociedad.

Si quieres ampliar conocimientos sobre lenguaje inclusivo hacia otros colectivos, recomendamos la guía [Uso inclusivo del lenguaje](#)¹⁰⁸ publicada por la Universidad del País Vasco, donde se habla del sexismo, racismo, homofobia y transfobia a partir del lenguaje y de las imágenes.

¹⁰⁶ <https://www.consaludmental.org/publicaciones/Guia-estilo-discapacidad-medios-comunicacion.pdf>

¹⁰⁷ <https://consaludmental.org/publicaciones/Guia-estilo-salud-mental.pdf>

¹⁰⁸ <https://www.ehu.eus/documents/2007376/10507176/Uso-inclusivo-del-castellano.pdf/7dce2de6-4ad3-7353-dd5c-68312586a3cc>



Herramientas de validación

No existen herramientas que detecten todos los errores de accesibilidad de un sitio web, pero existen aplicaciones que ayudan a comprobar criterios específicos y registrar la evaluación de los criterios.

En este capítulo te presentamos las más relevantes.

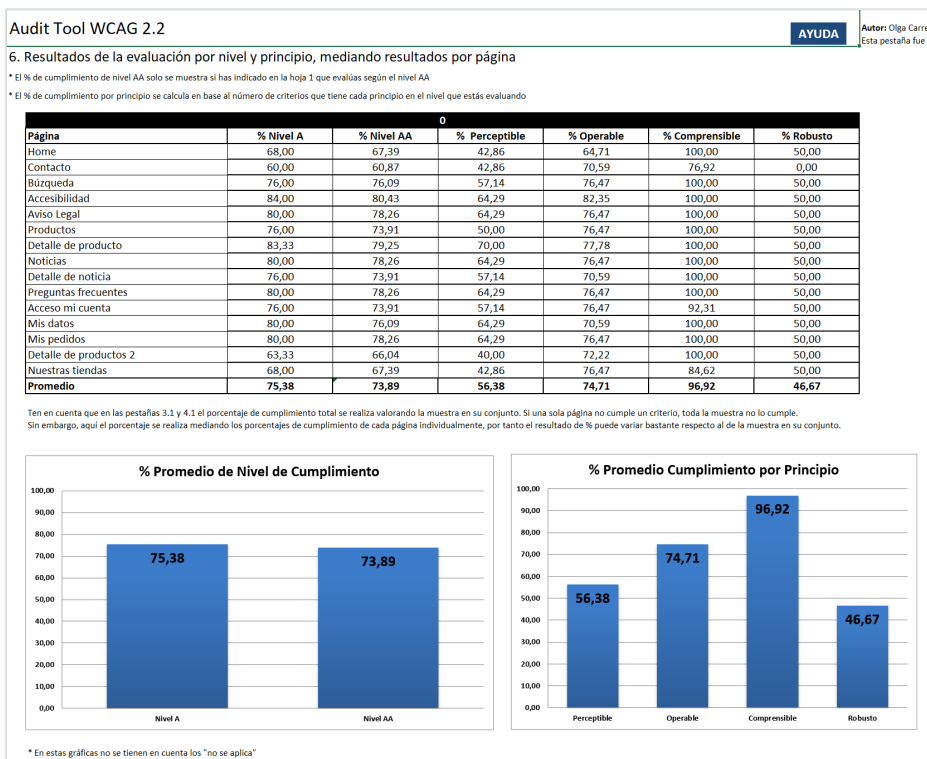
Audit Tool WCAG 2.2

Es una herramienta en formato Excel que permite ir recogiendo los datos obtenidos durante la revisión automática y manual de un sitio web de acuerdo con las WCAG 2.2. La herramienta 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 y las WCAG 2.1.

Esta información es útil para realizar el informe ejecutivo y presentar los resultados; comparar sitios web 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.

Además, sigue disponible la descarga de la versión anterior para evaluar de acuerdo con las WCAG 2.1, tanto en español como en inglés.

Enlace: [Audit Tool WCAG 2.2 \(español\)](#)¹⁰⁹



Captura de pantalla 34 Audit Tool WCAG 2.2

¹⁰⁹ <https://olgacarreras.blogspot.com/2023/12/audit-tool-wcag-22-herramienta-para.html>

Herramientas de validación global automática

Hay muchas aplicaciones, extensiones y marcadores de navegador que realizan validaciones automáticas de una página web. Reseñamos a continuación varias de ellas.

Accessibility Insights

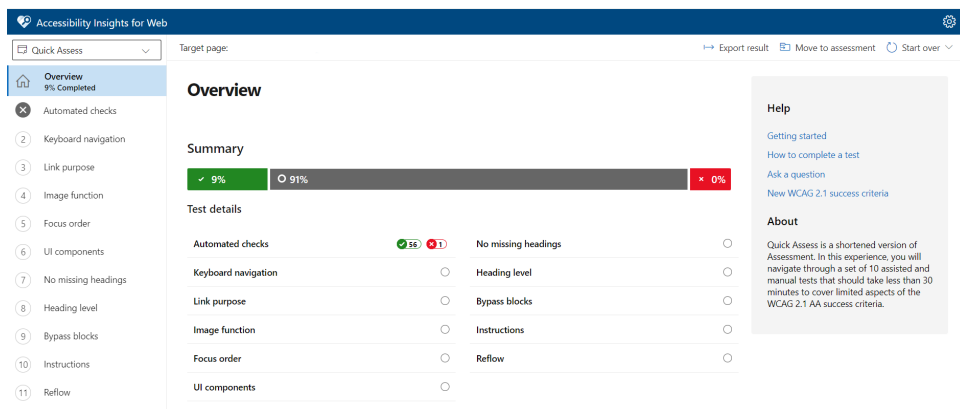
Accessibility Insights de Microsoft está disponible de forma gratuita como extensión de Chrome y de Edge, y como aplicación de Windows.

La herramienta admite tres escenarios principales:

- **FastPass:** es un proceso rápido que ayuda a los desarrolladores a identificar problemas de accesibilidad comunes. La herramienta realiza una evaluación automática, da instrucciones para validar el acceso por teclado y un listado de posibles errores que deben revisarse manualmente.
- **Quick Assess:** además de la evaluación automática, proporciona instrucciones paso a paso para realizar de forma asistida diez tipos de comprobaciones manuales: acceso por teclado, propósito de los enlaces, función de las imágenes, etc., lo cual puede ser útil para evaluadores con menos experiencia.
- **Assessment:** en este caso, son 24 el número de tipos de validaciones guiadas.

El informe se puede exportar como HTML y JSON.

Enlace: [Accessibility Insights](https://accessibilityinsights.io/) ¹¹⁰



The screenshot displays the 'Accessibility Insights for Web' interface. The top navigation bar includes 'Quick Assess' and 'Target page:'. Below the navigation bar, the 'Overview' section is active, showing a progress bar with '9%' completed (green) and '0%' remaining (red). The 'Test details' section lists various checks: 'Automated checks' (36 passed, 1 failed), 'No missing headings', 'Keyboard navigation', 'Heading level', 'Link purpose', 'Bypass blocks', 'Image function', 'Instructions', 'Focus order', 'Reflow', and 'UI components'. A 'Help' sidebar on the right provides links for 'Getting started', 'How to complete a test', 'Ask a question', and 'New WCAG 2.1 success criteria', along with an 'About' section explaining the tool's purpose.

Captura de pantalla 35. *Accessibility Insights*

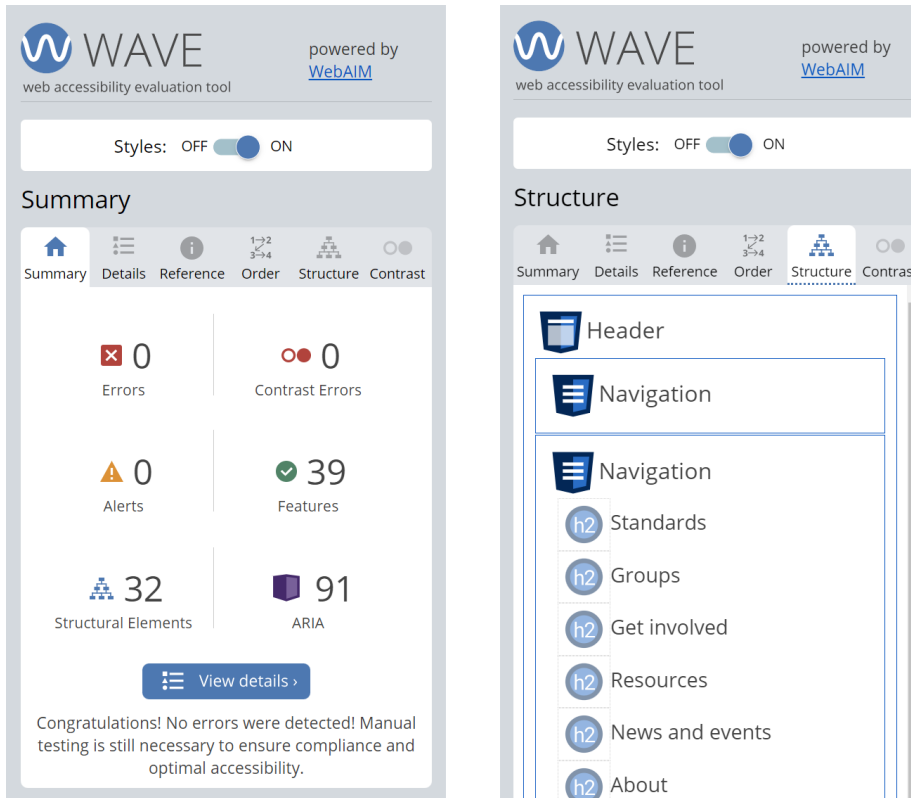
¹¹⁰ <https://accessibilityinsights.io/>

WAVE

WAVE es un conjunto de herramienta desarrollado por WebAIM, como la extensión del navegador para Chrome, Firefox y Edge; o la suscripción a WAVE API e instalación local.

La extensión de navegador, además de hacer validaciones automáticas, permite revisar el contraste, deshabilitar los estilos, consultar el orden de tabulación y la estructura de las páginas a través de sus regiones y encabezados.

Enlace: [WAVE](https://wave.webaim.org/) ¹¹¹



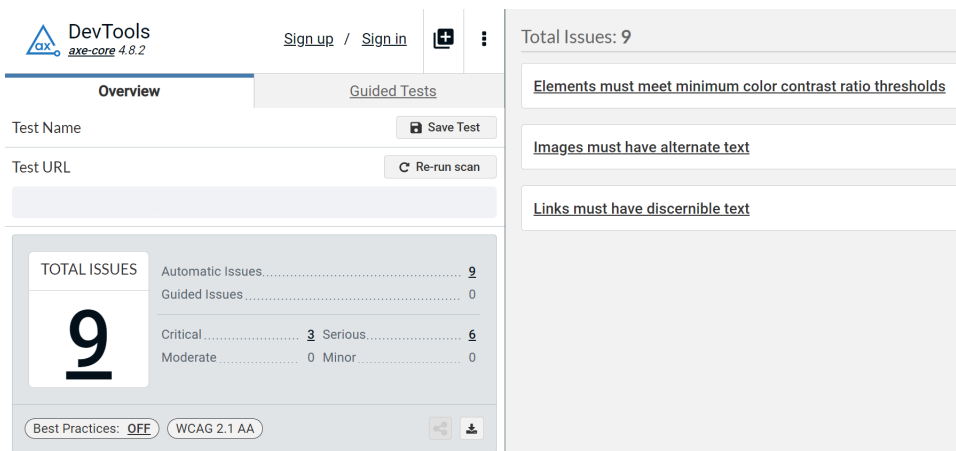
Captura de pantalla 36. Extensión WAVE.

¹¹¹ <https://wave.webaim.org/>

Axe DevTools

Axe DevTools es una herramienta desarrollada por la empresa Deque. Es una extensión de navegador con funcionalidades gratuitas y otras de pago. En la versión de pago ya se incluye la validación de acuerdo con las WCAG 2.2.

Enlace: [Axe DevTools](#) ¹¹²



Captura de pantalla 37. Axe DevTools

Deque también tiene otras herramientas como [axe Auditor](#)¹¹³ que ayuda a la revisión manual mediante heurísticas, [axe Monitor](#)¹¹⁴ para auditar y monitorizar sitios web completos, y [axe-core](#)¹¹⁵, un motor de pruebas de accesibilidad para sitios web y otras interfaces de usuario basadas en HTML.

¹¹² <https://www.deque.com/axe/browser-extensions/>

¹¹³ <https://www.deque.com/axe/auditor/>

¹¹⁴ <https://www.deque.com/axe/monitor/>

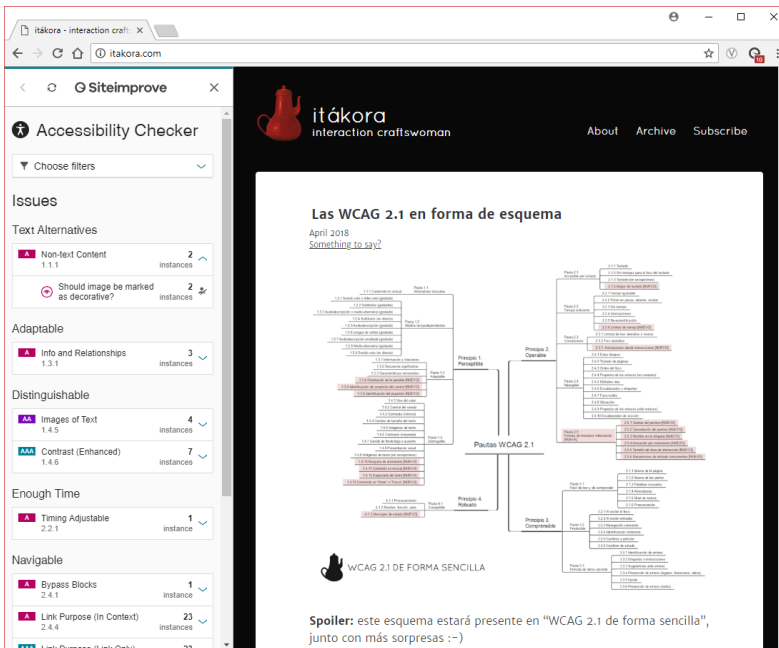
¹¹⁵ <https://github.com/dequelabs/axe-core>

SiteImprove

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 UX/UI, *webmaster* y desarrollador). Es muy fiable en los resultados, sin falsos positivos. Solo disponible en inglés.

Enlaces:

- [extensión SiteImprove del navegador Chrome \(versión 2021\)](#) ¹¹⁶
- [extensión SiteImprove del navegador Chrome \(versión 2023\)](#) ¹¹⁷



Captura de pantalla 38 Extensión SiteImprove (versión 2021)

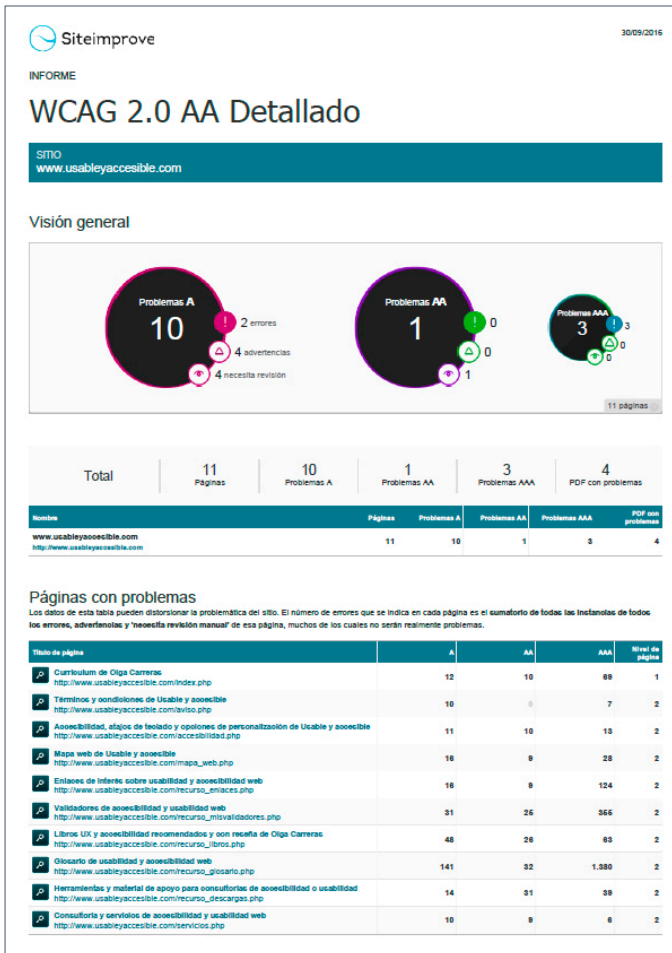
SiteImprove también existe como aplicación online profesional de pago que permite programar evaluaciones periódicas de sitios completos (incluidos documentos PDF), integrarse con diversos gestores de contenidos, generar informes detallados, definir análisis personalizados, además de tener gestión de usuarios, histórico de acciones y otros tipos de análisis (ortografía, posicionamiento en buscadores, inventarios, etc.)

Enlace: [página web de SiteImprove](#) ¹¹⁸

¹¹⁶ <https://chrome.google.com/webstore/detail/siteimprove-accessibility/efcfolpjhicnikpmhnmphjhpicljj>

¹¹⁷ <https://chromewebstore.google.com/detail/siteimprove-accessibility/djcglbmbegflehmbleechkjmedcopn>

¹¹⁸ <https://siteimprove.com/en/content-accessibility/>



Captura de pantalla 39 [Ejemplo de informe](http://www.usableyaccessible.com/archivos/informe_ejemplo_siteimprove.pdf)¹¹⁹ generado por la aplicación de pago Siteimprove

¹¹⁹ http://www.usableyaccessible.com/archivos/informe_ejemplo_siteimprove.pdf

Validador del Observatorio de Accesibilidad Web (OAW)

Las herramientas que te permiten evaluar sitios web completos suelen ser herramientas de pago, sin embargo, el **Observatorio de Accesibilidad Web (OAW)** dispone de una herramienta gratuita.

El rastreador OAW puede ser utilizado bajo demanda por el personal del sector público en España desde el sitio web del PAE (Portal de la Administración Electrónica) pero también puede instalarse de manera local.

Es uno de los validadores más completos y fiables, permite evaluar hasta 51 páginas y genera un informe en PDF muy detallado.

Enlace: [Validador de accesibilidad del Observatorio de Accesibilidad Web](#)¹²¹

Captura de pantalla 40 Validador de accesibilidad del OAW

¹²¹ <https://github.com/ctt-gob-es/oaw>

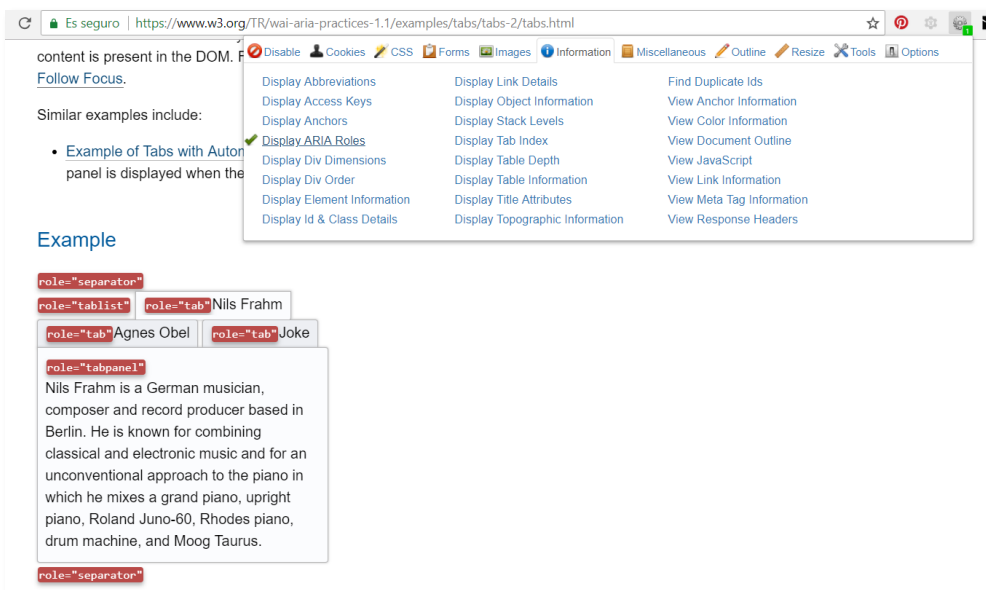
Herramientas de ayuda en la validación manual

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**.

Enlace: [Web Developer Toolbar](https://www.w3.org/TR/wai-aria-practices-1.1/examples/tabs/tabs-2/tabs.html)¹²¹



Captura de pantalla 41 Barra de navegador Web Developer Toolbar

¹²¹ <https://chrispederick.com/work/web-developer>

ANDI

ANDI es una herramienta de la SSA (*Social Security Administration*) de los EEUU que se incluye como un marcador del navegador.

Tiene seis funcionalidades:

- **Elementos que reciben el foco:** puedes ir avanzando entre los diferentes elementos que toman el foco, los cuales quedan resaltados en pantalla. Puedes visualizar el orden de tabulación y la etiqueta que tienen aplicada.
- **Imágenes:** identifica las imágenes, incluidos los SVG, y te permite validar fácilmente su código y texto alternativo, indicándote cómo lo anuncia el lector de pantalla.
- **Enlaces y botones:** puedes avanzar visualizando su código y cómo lo leerá el lector de pantalla; también te muestra un listado de enlaces internos y externos con su nombre accesible y su URL.
- **Estructura:** puedes navegar entre los encabezados, las listas, los *landmarks* y las *live regions* visualizando el código y cómo será anunciado por el lector de pantalla. También puedes consultar el título de la página, el idioma, y los atributos “role” y “lang”.
- **Contraste de color:** además de la ratio de contraste y los colores usados, te indica el tamaño de fuente. Permite visualizar la página en blanco y negro.
- **Contenido oculto:** puedes revelar el contenido oculto y encontrar rápidamente el contenido oculto con diferentes técnicas.

Enlace: [ANDI](#) ¹²²



Captura de pantalla 42 Herramienta ANDI

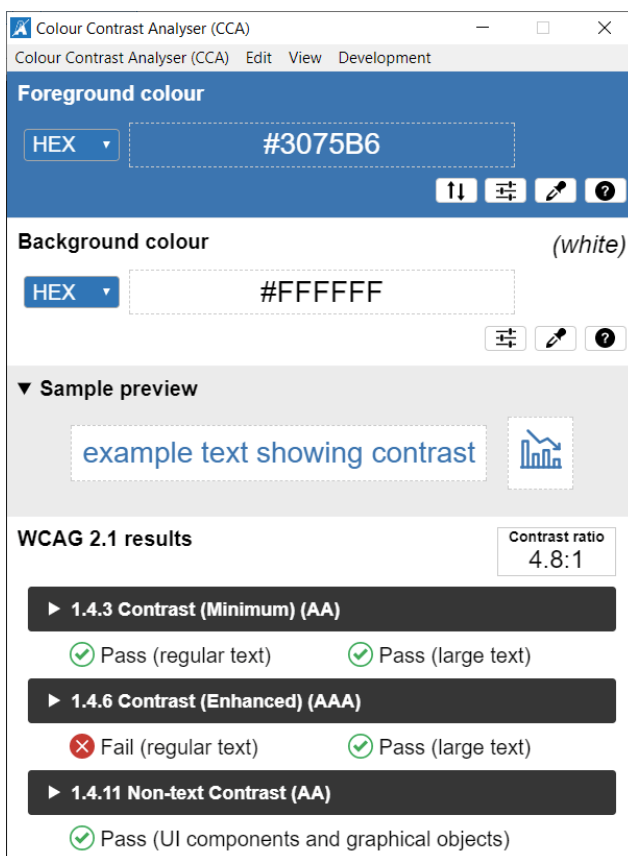
¹²² <https://www.ssa.gov/accessibility/andi/help/howtouse.html>

Herramientas de validación de criterios

Validación del 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*. Con la herramienta de cuentagotas seleccionas los colores de cualquier parte de la pantalla y compruebas los resultados del contraste. Si lo prefieres, puedes indicar directamente los códigos de color que quieres analizar.

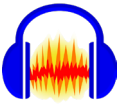
Enlace: [Colour Contrast Analyser](https://developer.paciellogroup.com/resources/contrastanalyser/)¹²³ para Windows y Mac



Captura de pantalla 43 Color Contrast Analyser

¹²³ <https://developer.paciellogroup.com/resources/contrastanalyser/>

Validación del contraste de sonidos

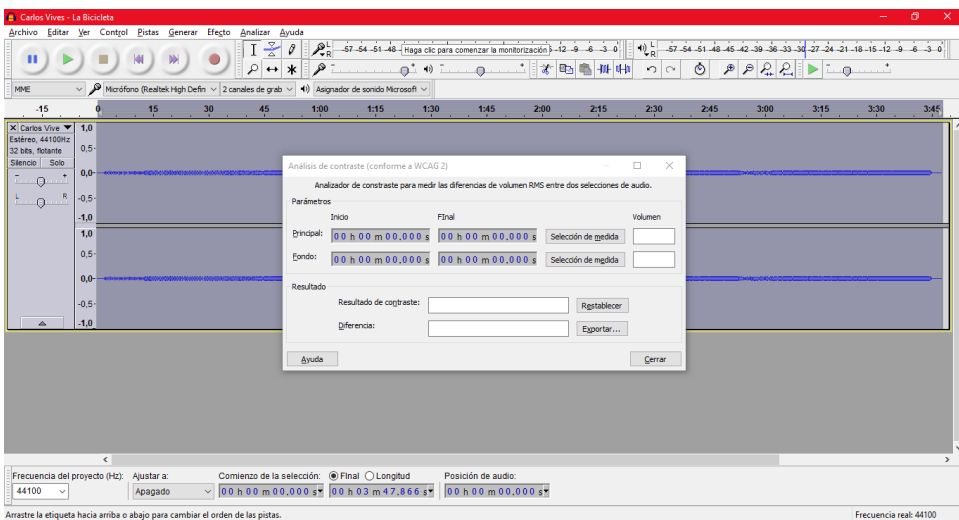


Para comprobar si los sonidos de fondo y de primer plano se diferencian 20 decibelios se puede usar Audacity®, una herramienta de escritorio gratuita (disponible para Windows, Max y Linux) que incluye una opción para verificar el contraste de acuerdo con el criterio 1.4.7 (AAA)

Dentro del menú “Analizar” tienes la opción “Contraste”. Selecciona las partes de la pista de audio que quieras verificar y obtendrás el resultado.

Enlaces:

- [Descarga de Audacity](#) ¹²⁴
- [Tutorial: Cómo comprobar el contraste con Audacity](#) ¹²⁵



Captura de pantalla 44 Análisis de contraste de sonidos con Audacity

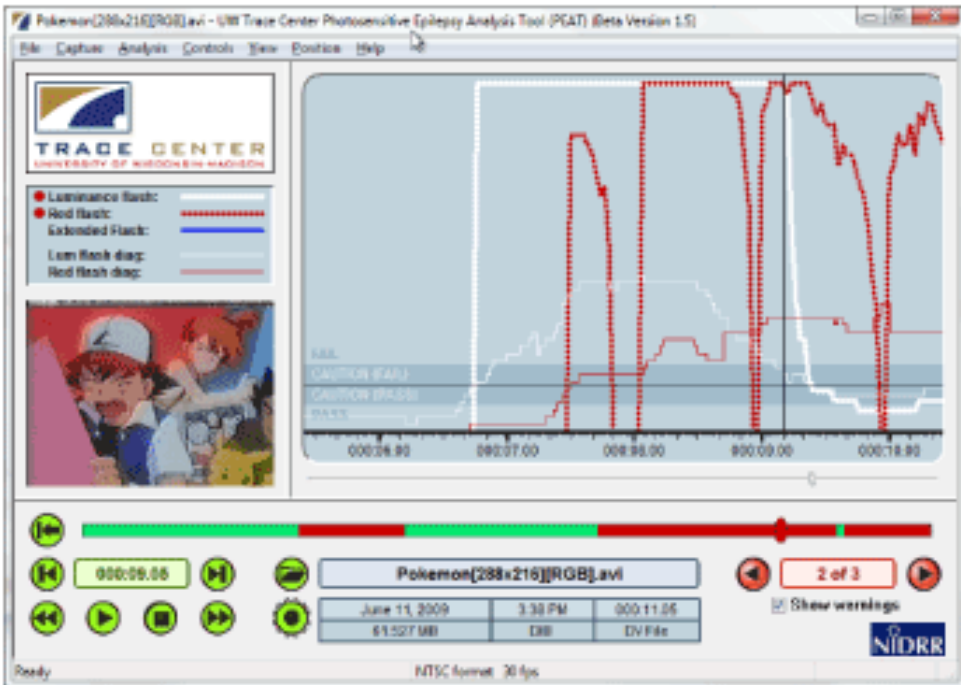
¹²⁴ <https://www.audacityteam.org/>

¹²⁵ <https://manual.audacityteam.org/man/contrast.html>

Validación de los destellos

De acuerdo con los criterios 2.3.1 (A) y 2.3.2 (AAA), para probar si un vídeo contiene destellos que pueden causar convulsiones, recomendamos la herramienta **PEAT** – *Photosensitive Epilepsy Analysis Tool*, que muestra visualmente los momentos conflictivos.

Enlace: [PEAT](https://trace.umd.edu/peat/)¹²⁶



Captura de pantalla 45 Análisis de destellos con PEAT

¹²⁶ <https://trace.umd.edu/peat/>

Validación de la legibilidad de los textos

El programa gratuito 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).

Enlace: [INFLESZ](#) ¹²⁷



Captura de pantalla 46 Análisis de legibilidad con Análisis con INFLESZ

Otra herramienta similar, también gratuita pero online en este caso, es **Legible**:

Enlace: [Legible](#) ¹²⁸

Legible

[Contacto](#) [Blog](#)

Analizador de legibilidad de texto

Averigua si un texto castellano es fácil de leer con esta herramienta. Pega o teclea tu texto o la URL y pulsa el botón «Analizar»:

Texto o dirección web (URL):

Introduce la dirección web que quieras analizar o pega o teclea el texto de hasta dos millones de caracteres y pulsa «Analizar».

[Analizar](#)

Captura de pantalla 47 Análisis de legibilidad con Legible

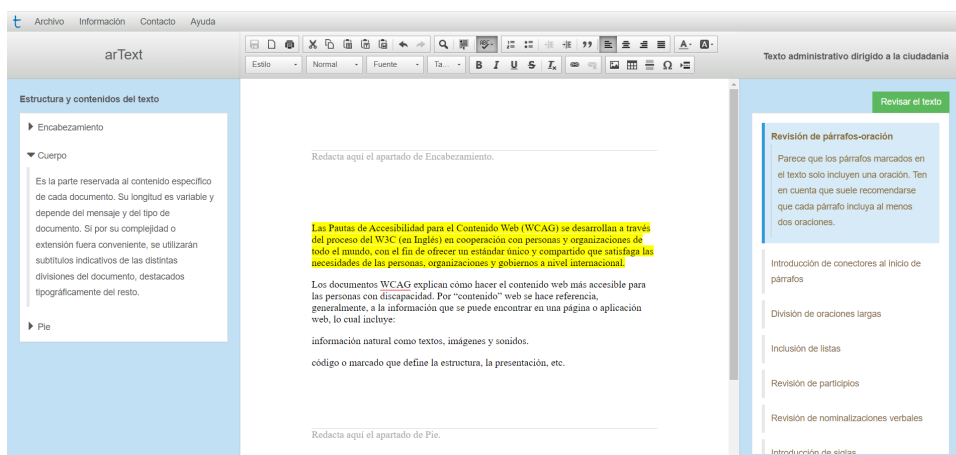
¹²⁷ <https://legibilidad.blogspot.com/2015/01/el-programa-inflesz.html>

¹²⁸ <https://legible.es/>

Una herramienta que te puede resultar muy útil para escribir textos con un lenguaje más claro es **arText**, una aplicación online y gratuita que incorpora recursos y estrategias del Procesamiento del Lenguaje Natural (PLN), una rama de la inteligencia artificial (IA).

arText aporta recomendaciones específicas sobre la estructura y la redacción, así como sugerencias lingüísticas para lograr que el texto sea más claro y comprensible.

Enlace: [arText](#) ¹²⁹



Captura de pantalla 48 Redacción de un texto con **arText**, a la derecha las recomendaciones de mejora

Hay **otras herramientas** que puedes probar:

- [Clara](#) ¹³⁰, herramienta gratuita y online de Prodigioso Volcán. Es un experimento de inteligencia artificial que analiza, mediante aprendizaje automático y diversos algoritmos, la claridad de textos en español.
- [Easier](#) ¹³¹, herramienta gratuita y online desarrollada por la Universidad Carlos III de Madrid (UC3M). Detecta las palabras difíciles de un texto, ofreciendo un pictograma de [ARASAAC](#) ¹³², sinónimos y una definición, si es posible en lectura fácil gracias al ["Diccionario fácil"](#) ¹³³ de Plena Inclusión Madrid.
- [ChatGPT](#) ¹³⁴, herramienta online con una versión gratuita. Puedes darle instrucciones claras y precisas para que corrija la ortografía y la gramática de un texto, o para que adapte el tono, la claridad o la sencillez de este.

¹²⁹ <http://sistema-artext.com>

¹³⁰ <https://clara.comunicacionclara.com>

¹³¹ <http://easier.hulat.uc3m.es>

¹³² <https://arasaac.org>

¹³³ <https://www.diccionariofacil.org>

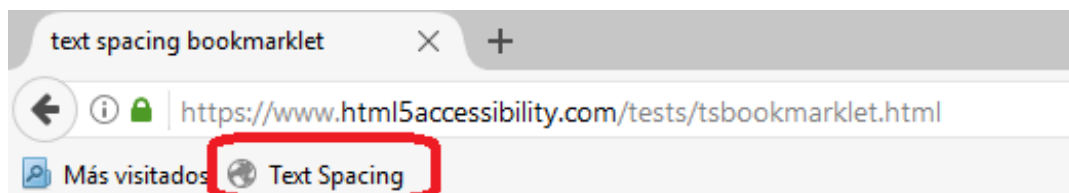
¹³⁴ <https://chat.openai.com>

Validación del espaciado de texto

El 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 podemos usar el *bookmarklet* “*Text spacing*” de **Steve Faulkner**, que se incluye en el navegador como un marcador.

Enlace: [bookmarklet “Text spacing” de Steve Faulkner](#)¹³⁵



Captura de pantalla 49 Bookmarklet “Text spacing” de Steve Faulkner

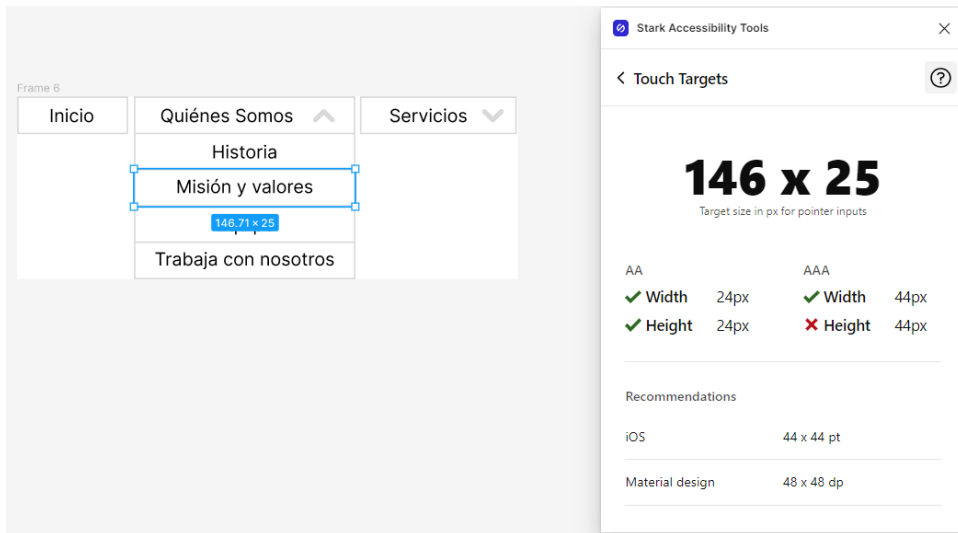
135

https://espaciocompartir.inap.es/v3/pluginfile.php/5071/mod_resource/content/20/bookmarklet_text_spacing_de_steve_faulkner.html

Validación del tamaño del área de interacción

Los criterios 2.5.8 (AA) y 2.5.5 (AAA) nos indican que los elementos interactivos deben contar con un área mínima de interacción para que puedan ser activados sin problemas. Con el plugin de Figma **Stark Accessibility Tools** se puede comprobar si el elemento interactivo cumple o no con las dimensiones mínimas.

Enlace: [plugin de Figma Stark Accessibility Tools](#) ¹³⁶



Captura de pantalla 50 Comprobación de tamaños de área de interacción con *Stark Accessibility Tools*

¹³⁶ <https://www.figma.com/community/plugin/732603254453395948>

Validación de código

El Servicio de Validación del W3C ofrece dos herramientas para comprobar el criterio 4.1.1, una para validar el código HTML, XHTML, SMIL, MathML o SVG de tu web y otra para las hojas de estilo CSS.

Enlaces:

- [Servicio de Validación del W3C](https://validator.w3.org/)¹³⁷
- [Validador de CSS del W3C](https://jigsaw.w3.org/css-validator/)¹³⁸

The screenshot shows the W3C Markup Validation Service interface. At the top, there's a blue header with the W3C logo and the text "Markup Validation Service". Below this, there are three tabs: "Validate by URI", "Validate by File Upload", and "Validate by Direct Input". The "Validate by URI" tab is selected. Underneath, there's a form with a "Validate a document online" label and an "Address:" input field. Below the input field, there's a "More Options" section with several checkboxes and dropdown menus. The "Check" button is at the bottom of the form. Below the form, there's a paragraph of text explaining the service and a link to "Other validators and tools". At the bottom, there's a footer with the HP logo, a "Donate" button, and a copyright notice.

Captura de pantalla 51 Servicio de validación de código del W3C

The screenshot shows the W3C CSS Validation Service interface. At the top, there's a blue header with the W3C logo and the text "CSS Validation Service". Below this, there are three tabs: "mediante URI", "mediante Carga de Archivo", and "mediante Entrada directa". The "mediante URI" tab is selected. Underneath, there's a form with a "Validar mediante URI" label and an "Introduce la URI de un documento (HTML con CSS o sólo CSS) que deseas validar:" label. Below this, there's a "Dirección:" input field. Below the input field, there's a "Más opciones" section with several dropdown menus and checkboxes. The "Check" button is at the bottom of the form. Below the form, there's a paragraph of text explaining the service and a link to "comprobar que el (X)HTML utilizado es válido". At the bottom, there's a footer with the Mozilla logo, a "Donate" button, and a copyright notice.

Captura de pantalla 52 Servicio de validación de CSS del W3C

¹³⁷ <https://validator.w3.org/>

¹³⁸ <https://jigsaw.w3.org/css-validator/>



Herramientas de trabajo

Aparte de la evaluación y validación, existen otras herramientas que puedes usar para garantizar la accesibilidad de tus proyectos en diferentes momentos del ciclo del producto.

Te mostramos los principales lectores de pantalla, reproductores de audio y vídeo; y simuladores de discapacidades, para que puedas experimentar en primera persona cómo se manejan las personas con discapacidad en la web.

Por último, los diseñadores deben transferir a los desarrolladores y otros miembros del equipo sus creaciones, y la documentación de traspaso es una herramienta fundamental para verificar y comunicar la accesibilidad de los componentes creados.

Lectores de pantalla

Los lectores de pantalla de escritorio se manejan con atajos de teclado, y los lectores de pantalla de dispositivos móviles mediante gestos táctiles.

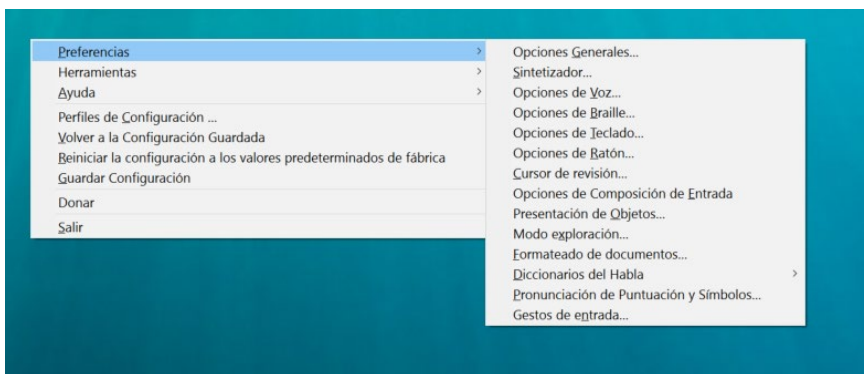
Las opciones son parecidas entre los diferentes lectores de pantalla, pero cada uno tiene sus peculiaridades y no siempre enuncian los contenidos de la misma manera. Por ello, es importante ajustarse lo máximo posible a los estándares y a la norma, para que puedan interpretar las páginas y aplicaciones con la mayor fidelidad posible.

Lectores de pantalla de escritorio

Existen muchos lectores de pantalla, pero nuestra recomendación para *Windows* es **NVDA**, que es gratuito y ofrece muchas opciones de configuración.

Enlaces:

- [NVDA](#) ¹³⁹
- [Guía de uso de NVDA](#) ¹⁴⁰
- [Principales atajos de teclado de NVDA](#) ¹⁴¹



Captura de pantalla 53 Opciones de NVDA en desktop

Windows incluye un navegador por defecto en el sistema: **Narrador**, que se puede activar pulsando las teclas Windows + Control + Enter.

El lector de pantalla por defecto de macOS es **VoiceOver**, que puedes activar con Comando + F5.

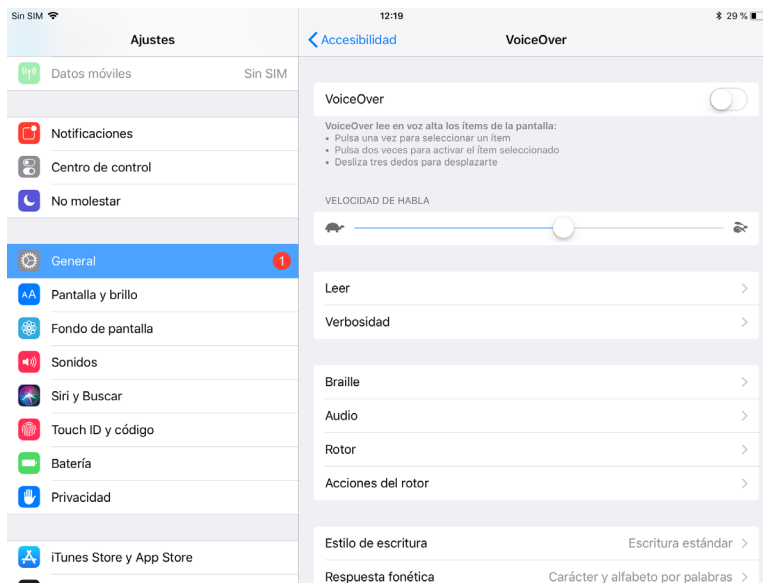
¹³⁹ <https://www.nvaccess.org>

¹⁴⁰ <https://www.nvaccess.org/files/nvda/documentation/userGuide.html>

¹⁴¹ <https://dequeuniversity.com/screenreaders/nvda-keyboard-shortcuts>

Lectores de dispositivos móviles

Las personas que utilizan el sistema operativo iOS (Apple) disponen del lector de pantalla **VoiceOver** integrado en el sistema.



Captura de pantalla 54 VoiceOver en iOS para tablet

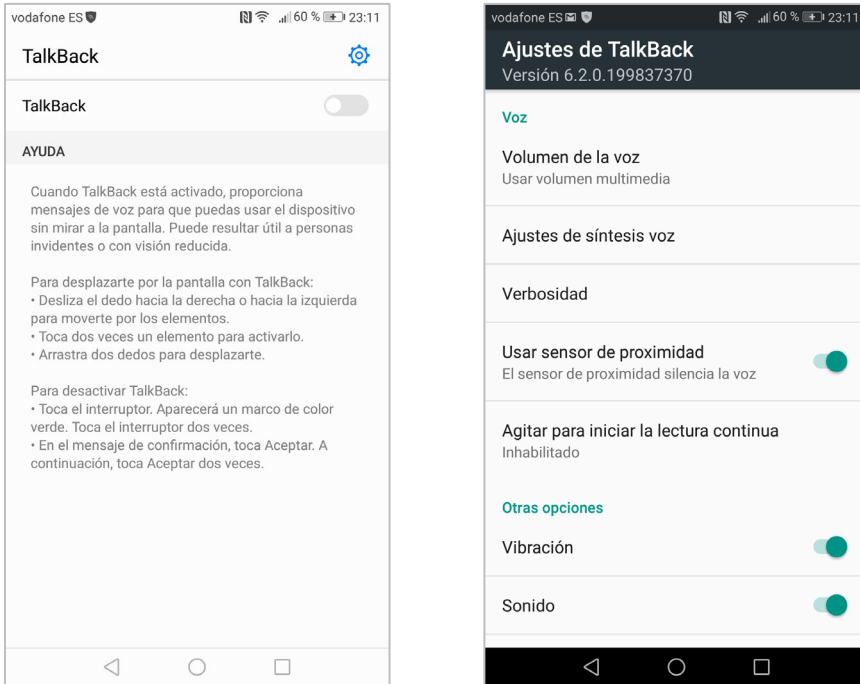
Además de los gestos más sencillos, como deslizar el dedo hacia la derecha o hacia la izquierda para recorrer los elementos de pantalla, o dos toques para seleccionar el elemento que tiene el foco, uno de los gestos más útiles es el rotor. Para activarlo tienes que girar dos dedos sobre la pantalla del dispositivo iOS o iPadOS como si estuvieras girando un dial.

Te permite seleccionar diferentes opciones, por ejemplo, moverte por una página web a través de sus encabezados o a través de sus regiones:



Captura de pantalla 55 VoiceOver en iOS con el rotor activado

En el sistema operativo Android se puede activar **TalkBack** dentro de Ajustes.



Captura de pantalla 56 Funcionalidad de Talkback y sus ajustes en un móvil Android

Además de los gestos más sencillos, como deslizar el dedo hacia la derecha o hacia la izquierda para recorrer los elementos de pantalla, o dos toques para seleccionar el elemento que tiene el foco, con Talkback puedes sacar un menú contextual deslizando un dedo hacia abajo y hacia la derecha en un movimiento continuo.

Si estás en una página web puedes probar a deslizar, de forma rápida y seguida, el dedo hacia arriba y luego hacia abajo para seleccionar cómo navegar por la página (por ejemplo, a través de sus encabezados):



Captura de pantalla 57 Menú de Talkback en una página web



Guía de interés sobre accesibilidad de aplicaciones móviles

Si te interesa cómo aplicar las WCAG 2.2 y la EN 301 549 en la revisión de accesibilidad de aplicaciones móviles nativas, puedes leer el libro gratuito "[Guía: Aplicación móviles accesibles](#)"¹⁴², de Olga Carreras (2023).

¹⁴² <https://www.cedid.es/es/documentacion/ver-seleccion-novedad/586759/>

Reproductores de audio y vídeo 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.

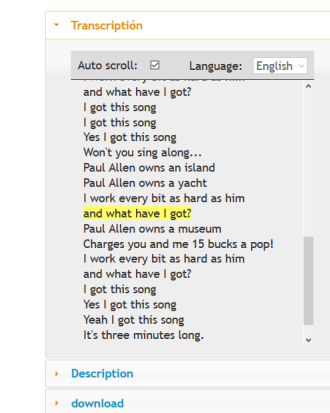
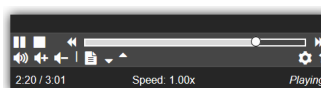
Por ello es muy recomendable utilizar otros reproductores HTML5, basados en los elementos `<audio>` o `<video>`, que sí son completamente accesibles y admiten todas las alternativas necesarias para cumplir con los criterios de la Pauta 1.2. Nuestras recomendaciones son **Able Player** y **OzPlayer**. Ambos admiten vídeos de YouTube, siendo más recomendable utilizar estos reproductores por su apoyo ampliado a la accesibilidad que el propio reproductor de YouTube. Puedes consultar una tabla comparativa de diferentes reproductores y sus características en [HTML5 Video Player Comparison](#)¹⁴³

Enlaces:

- [Able Player](#)¹⁴⁴
- [OzPlayer](#)¹⁴⁵



Captura de pantalla 58 Vídeo con transcripción en Oz Player



Captura de pantalla 59 Audio con transcripción en Able Player

¹⁴³ <http://videosws.praegnanz.de/#sws>

¹⁴⁴ <https://ableplayer.github.io/ableplayer/>

¹⁴⁵ <https://www.accessibilityoz.com/ozplayer>

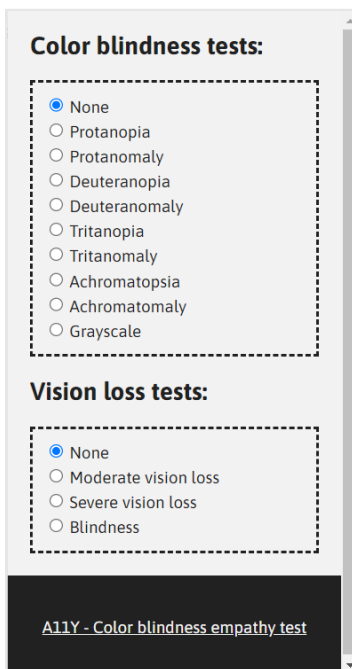
Simulación del acceso con discapacidad visual

En el navegador Chrome y en los programas Figma o Sketch podemos usar extensiones que modifican las páginas web de tal manera que permiten ponernos en los zapatos de personas con daltonismo, baja visión y ceguera y comprobar cómo perciben, manejan y operan la web.

También podemos comprobar diferentes versiones de imágenes que estemos creando con la página Pilestone.

Enlaces:

- [Extensión A11Y](https://chrome.google.com/webstore/detail/a11y-color-blindness-empa/idphhflanmeibmigaciaadkmjebjhcc)¹⁴⁷ para Chrome.
- [Stark](https://www.getstark.co/support/getting-started/using-the-vision-simulator) para Figma y Sketch¹⁴⁸.
- [Simulador de daltonismo Pilestone](https://pilestone.com/pages/color-blindness-simulator-1#)¹⁴⁹



Captura de pantalla 60 Extensión A11Y para Chrome



Captura de pantalla 61 Extensión Stark para Figma y Sketch

¹⁴⁷ <https://chrome.google.com/webstore/detail/a11y-color-blindness-empa/idphhflanmeibmigaciaadkmjebjhcc>

¹⁴⁸ <https://www.getstark.co/support/getting-started/using-the-vision-simulator>

¹⁴⁹ <https://pilestone.com/pages/color-blindness-simulator-1#>

Documentación de traspaso de diseño a desarrollo

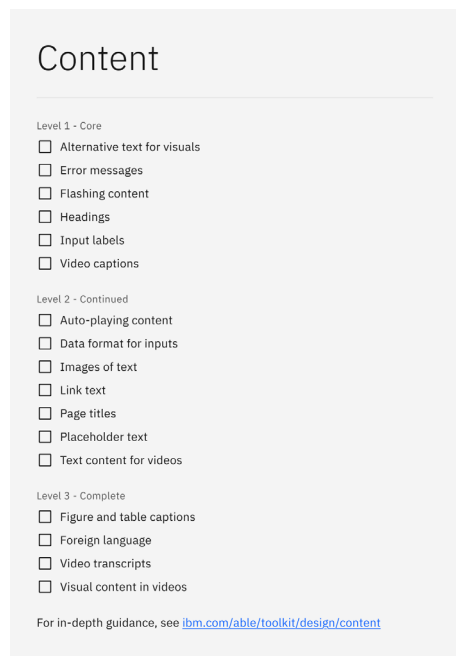
Un momento esencial en la dinámica de un equipo digital se produce cuando los diseñadores UX/UI transfieren su trabajo a los desarrolladores. En inglés, este paso se denomina “*handoff*”, que se traduce como “manos libres”, indicando la transferencia de responsabilidad de un equipo a otro. Para ejecutar este traspaso, los diseñadores UX/UI elaboran cierta documentación que facilita a los desarrolladores entender los diseños, incluidas las características de accesibilidad.

IBM ha elaborado un kit que facilita esta documentación, con una lista de comprobación y guías para que no se olvide documentar los encabezados, las etiquetas de formularios, el orden de los encabezados, el orden de la tabulación, la interacción por teclado...

El kit está disponible tanto en Figma como en Sketch.

Enlaces:

- [Figma: IBM Accessibility Design Kit](#)¹⁴⁹
- [Sketch: IBM Accessibility Design Kit](#)¹⁵⁰



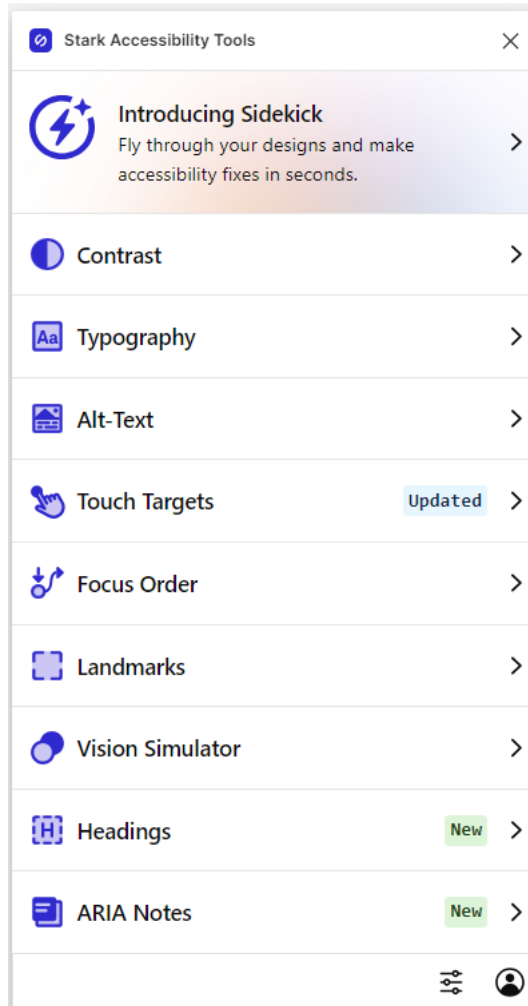
Captura de pantalla 62 Kit de accesibilidad de IBM

¹⁴⁹ <https://www.figma.com/community/file/1118184491812988116>

¹⁵⁰ <https://www.sketch.com/s/f0a04c0d-fb62-4d71-92c6-07c402f8cae7>

Además, otros *plugins*, como el ya mencionado **Stark Accessibility Tools**, se puede utilizar para integrar comentarios adicionales al diseño que permitan a los desarrolladores implementar los diseños de forma accesible, como la indicación del orden del foco, de los encabezados, de los textos alternativos de las imágenes, anotaciones de ARIA...

Enlace: [Stark Accessibility](#)¹⁵¹



Captura de pantalla 63 Stark Accessibility

¹⁵¹ <https://www.figma.com/community/plugin/732603254453395948>

Resúmenes y esquemas

En este apartado presentamos algunos resúmenes y esquemas que pueden ayudarte a entender mejor las WCAG 2.2.

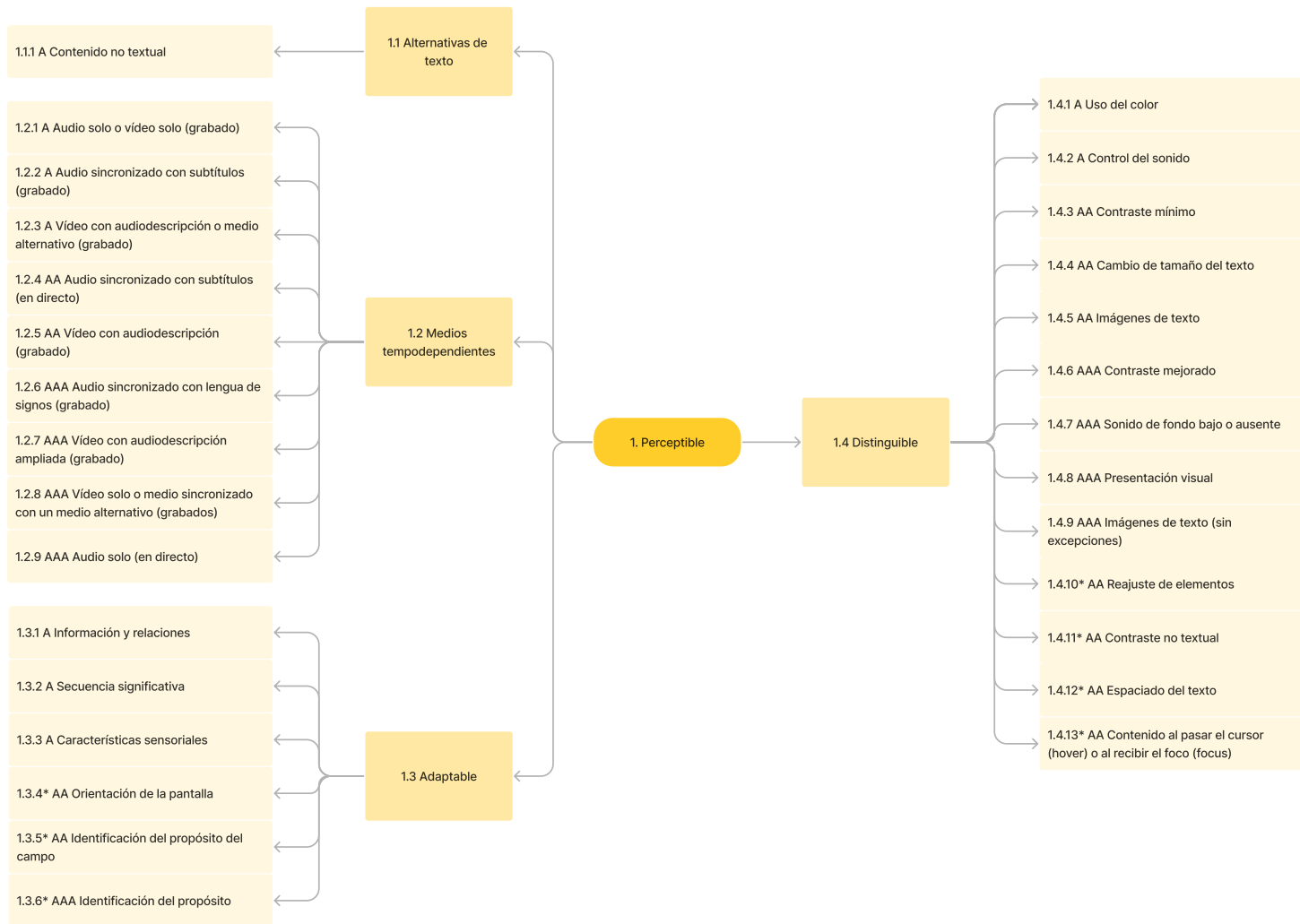
En primer lugar mostramos los diagramas conceptuales de las pautas, en forma esquemática.

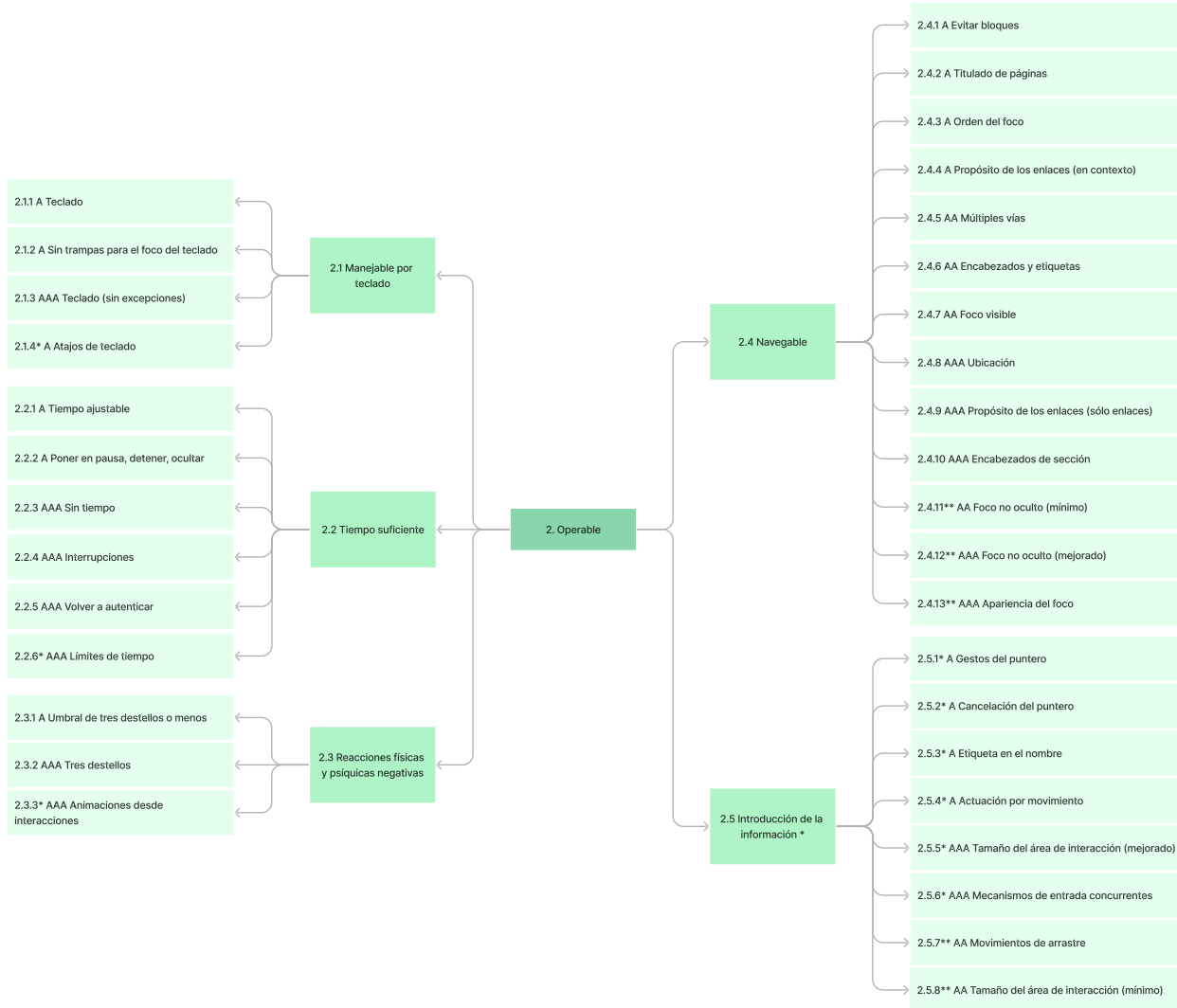
A continuación listamos las pautas y criterios agrupados de diferentes maneras, para que los puedas tomar como referencia de la forma en la que los necesites: por principios, por niveles, o por profesiones.

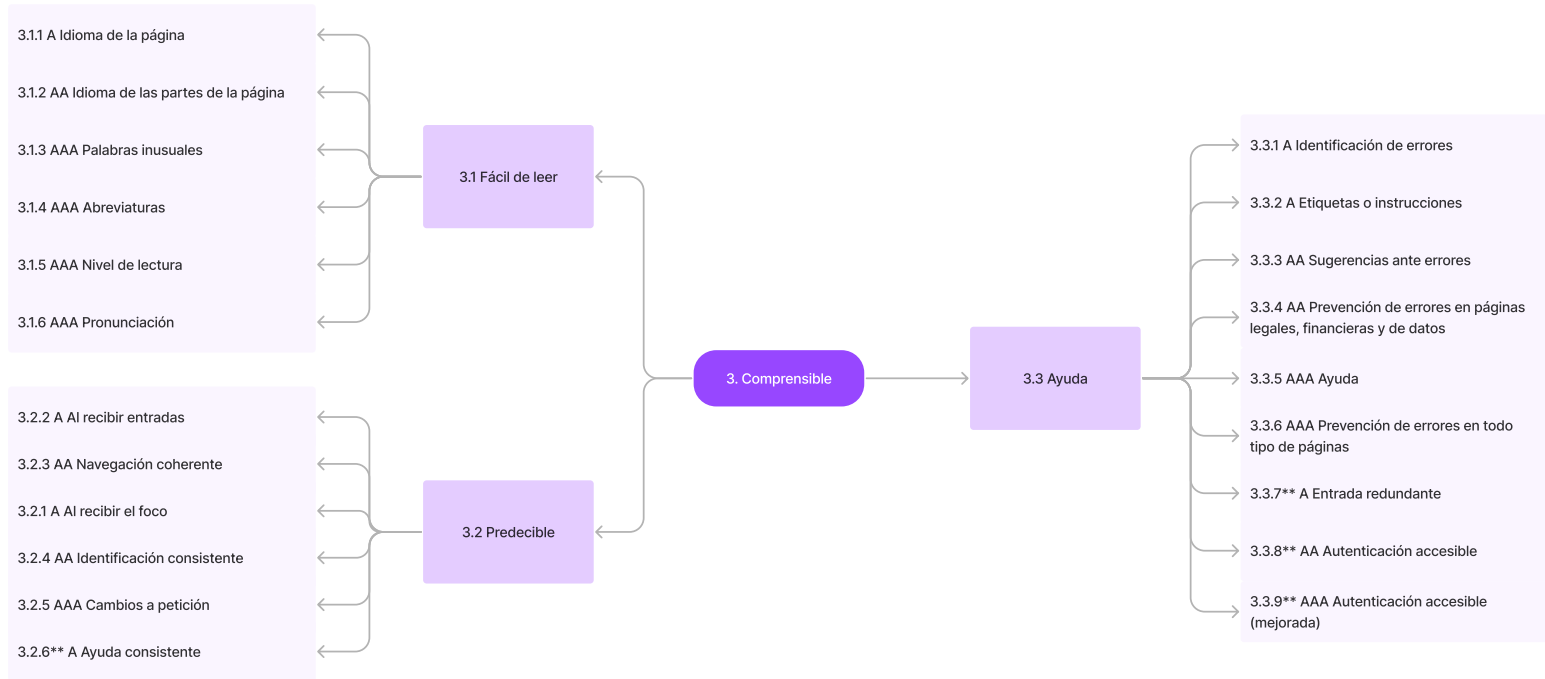
Diagramas conceptuales de las WCAG 2.2

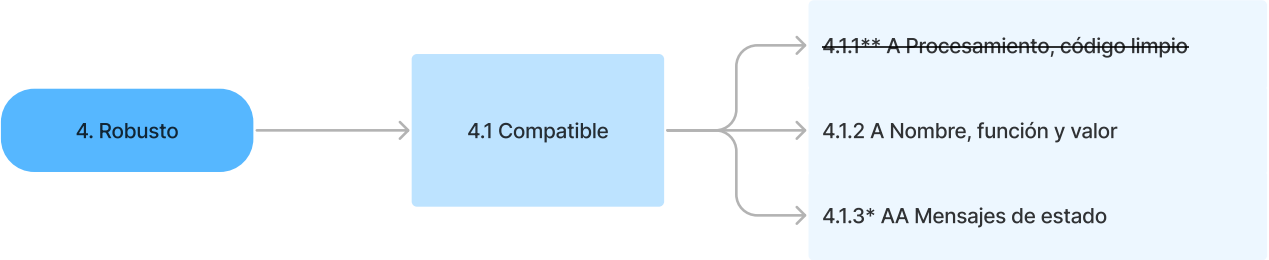
* Nuevos WCAG 2.1 ** Nuevos WCAG 2.2











Principios, Pautas y Criterios de Conformidad

* Nuevos WCAG 2.1 ** Nuevos WCAG 2.2

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	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)

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 al pasar el cursor (<i>hover</i>) o al recibir el foco (<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	Volver a autenticar
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
2.4.11**	AA	Foco no oculto (mínimo)
2.4.12**	AAA	Foco no oculto (mejorado)
2.4.13**	AAA	Apariencia del foco

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 (mejorado)
2.5.6*	AAA	Mecanismos de entrada concurrentes
2.5.7**	AA	Movimientos de arrastre
2.5.8**	AA	Tamaño del área de interacción (mínimo)

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
3.2.6**	A	Ayuda consistente

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
3.3.7**	A	Entrada redundante
3.3.8**	AA	Autenticación accesible
3.3.9**	AAA	Autenticación accesible (mejorada)

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 WCAG 2.1 ** Nuevos WCAG 2.2

Nivel A

1.1.1	Contenido no textual
1.2.1	Audio 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
3.2.6**	Ayuda consistente
3.3.1	Identificación de errores
3.3.2	Etiquetas o instrucciones
3.3.7**	Entrada redundante
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 al pasar el cursor (<i>hover</i>) o al recibir el foco (<i>focus</i>)
2.4.5	Múltiples vías
2.4.6	Encabezados y etiquetas
2.4.7	Foco visible
2.4.11**	Foco no oculto (mínimo)
2.5.7**	Movimientos de arrastre
2.5.8**	Tamaño del área de interacción (mínimo)
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)
3.3.8**	Autenticación accesible
4.1.3*	Mensajes de estado

Nivel AAA

1.2.6	Lengua de signos (grabado)
1.2.7	Audiodescripción ampliada (grabada)
1.2.8	Medio alternativo (grabado)
1.2.9	Sólo audio (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	Volver a autenticar
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
2.4.12**	Foco no oculto (mejorado)
2.4.13**	Apariencia del foco
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)
3.3.9**	Autenticación accesible (mejorada)

Crerios para creadores de contenido

* Nuevos WCAG 2.1 ** Nuevos WCAG 2.2

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)
	1.4.2	A	Control del audio
Redacción sencilla	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
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

Concepto	Criterio	Nivel	Descripción
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.

Crterios para diseadores UX/UI

* Nuevos WCAG 2.1 ** Nuevos WCAG 2.2

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
	2.4.13**	AAA	Apariencia del foco
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
	3.2.6**	A	Ayuda consistente
	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
	3.3.7**	A	Entrada redundante
	3.3.8**	AA	Autenticación accesible (mínima)
	3.3.9**	AAA	Autenticación accesible (mejorada)
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

Concepto	Criterio	Nivel	Descripción
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 (mejorada)
	2.5.8**	AA	Tamaño del área de interacción (mínimo)
Imágenes, vídeos 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
Manejo sencillo	1.4.13*	AA	Contenido al pasar el cursor (<i>hover</i>) o al recibir el foco (<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
	2.5.7**	AA	Movimientos de arrastre
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	2.2.4	AAA	Interrupciones
	2.2.5	AAA	Volver a autenticar
No tomes decisiones	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 desarrolladores

* Nuevos WCAG 2.2 ** Nuevos WCAG 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 al pasar el cursor (<i>hover</i>) o al recibir el foco (<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
	2.4.11**	AA	Foco no oculto (mínimo)
	2.4.12**	AAA	Foco no oculto (mejorado)
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 rellenar los 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
	3.3.8**	AA	Autenticación accesible (mínima)
	3.3.9**	AAA	Autenticación accesible (mejorada)

Concepto	Criterio	Nivel	Descripción
No incomodar	2.2.4	AAA	Interrupciones
	2.2.5	AAA	Volver a autenticar
Sin agobios de tiempo	2.2.1	A	Tiempo ajustable
	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
	1.4.8	AAA	Presentación visual
	1.4.10*	AA	Reajuste de elementos
	1.4.12*	AA	Espaciado del texto

Glosario

ACR (Accessibility Conformance Report). Es un documento que describe el grado de conformidad de un producto o servicio de tecnología de la información y la comunicación respecto a un conjunto acordado de pautas y estándares internacionales de accesibilidad.

<https://www.section508.gov/sell/how-to-create-acr-with-vpat/>

AENOR - Asociación Española de Normalización y Certificación. Es la entidad reconocida en España como organismo nacional de normalización. Se dedica 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. Por ejemplo, MSA, IAccessible2 o UI Automation son API de accesibilidad del sistema operativo Windows. https://www.usableyaccessible.com/recurso_glosario.php#api_accesibilidad

ARIA - Accessible Rich Internet Applications. Es una especificación técnica que proporciona formas de marcar el código HTML u otros lenguajes de marcado 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>

COGA - Grupo de trabajo sobre accesibilidad a las discapacidades cognitivas y de aprendizaje.

<https://www.w3.org/WAI/GL/task-forces/coga/>

CSS - Cascading Style Sheets. Es un lenguaje de diseño para definir la presentación de las páginas HTML, u otros lenguajes de marcado, de forma independiente a su estructura.

<https://www.w3.org/Style/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 desarrollador de tal modo que las aplicaciones de usuario, incluyendo los productos de apoyo, pueden extraer y presentar esta información a las personas 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.

<https://www.w3.org/2005/03/DOM3Core-es/introduccion.html>

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>

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>

Flexbox. Es una propiedad de CSS para adaptar el tamaño y posición de los elementos a diferentes tamaños de pantalla y diferentes dispositivos.

https://www.w3schools.com/css/css3_flexbox.asp

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/HTTP>

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/#keybrd-interfacedef>

ISO - International Organization for Standardization. Es una organización internacional que publica estándares para todo tipo de productos y servicios.

<http://www.iso.org>

ITI - Information Technology Industry Council. Es una asociación que representa a un grupo de importantes empresas tecnológicas.

<https://www.itic.org/about/>

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 "HOL4".

<http://es.wikipedia.org/wiki/Leet>

Microformatos. Son pequeños patrones de HTML usados para representar semánticamente contactos de personas, eventos de calendario, lugares, etc.

<http://microformats.org>

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>

Producto de apoyo o ayudas técnicas. Cualquier tecnología (*hardware* o *software*) fabricada especialmente para prevenir, compensar, controlar, mitigar o neutralizar deficiencias, limitaciones en la actividad y restricciones en la participación de las personas con discapacidad.

<https://www.discapnet.es/vida-independiente/productos-de-apoyo>

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/>

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/Gráficos_vectoriales_escalables

Tests A/B: son un método para comparar dos versiones de una página web o aplicación para determinar cuál funciona mejor.

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. Mediante su referencia en disposiciones legales las autoridades pueden decidir que sea de obligado cumplimiento.

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 organización internacional sin ánimo de lucro de interés público, donde organizaciones miembro, un personal a tiempo completo 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 para las 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>

VPAT - Voluntary Product Accessibility Template. Es una plantilla creada por el ITI (*Information Technology Industry*) para elaborar un ACR (*Accessibility Conformance Report*). Está disponible para los principales estándares de accesibilidad: Sección 508 (Estados Unidos), EN 301 549 (Unión Europea) y W3C/WAI WCAG (versiones 2.0, 2.1 y 2.2).

<https://www.itic.org/policy/accessibility/vpat>

XHTML - eXtensible HyperText Markup Language. es un documento HTML expresado como XML válido, más estricto a nivel técnico. <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.....	9
Qué es la Accesibilidad Web	10
La interacción accesible.....	13
La accesibilidad y la usabilidad	16
La ética en el diseño de la interacción	17
Qué profesionales deben saber de Accesibilidad Web.....	19
Qué debes saber según tu profesión	21
Implantar y gestionar la política de Accesibilidad Web de una organización	27
Ejemplos de “Personas” para una universidad	30
Las Pautas WCAG	32
Qué son las Pautas WCAG	33
Novedades introducidas por las Pautas WCAG 2.2	34
Documentos que componen las Pautas WCAG 2.....	35
Cómo se organizan las Pautas WCAG 2.2	36
Los cuatro principios y sus pautas	37
Las técnicas y los errores.....	40
¿A qué tecnologías se aplican las WCAG?	41
Por qué seguir las WCAG, y actualizarse a las 2.2	42
Los requisitos de conformidad.....	43
Los 5 requisitos	44
Niveles de conformidad de las páginas web.....	45
Declaraciones de conformidad.....	46
Evaluar la accesibilidad con WCAG-EM.....	53
La metodología WCAG-EM	54
Proceso de revisión	55
Paso 1. Definir el alcance de la evaluación.....	56
Paso 2. Explorar el sitio web.....	57
Paso 3. Seleccionar una muestra representativa de páginas	58
Paso 4. Auditar la muestra seleccionada.....	59
Paso 5. Registrar los resultados y elaborar un informe	60
Declarar la conformidad tras seguir la WCAG-EM	61

Herramienta de ayuda WCAG-EM Report Tool.....	62
Principio 1. Perceptible.....	63
Pauta 1.1.....	64
Pauta 1.2.....	68
Pauta 1.3.....	81
Pauta 1.4.....	93
Principio 2. Operable.....	111
Pauta 2.1.....	112
Pauta 2.2.....	117
Pauta 2.3.....	126
Pauta 2.4.....	130
Pauta 2.5.....	145
Principio 3. Comprensible.....	156
Pauta 3.1.....	157
Pauta 3.2.....	164
Pauta 3.3.....	172
Principio 4. Robusto	183
Pauta 4.1.....	184
Documentos PDF accesibles.....	190
La accesibilidad en los documentos PDF	191
Las técnicas específicas de PDF.....	192
Implementar las técnicas PDF	193
Validar la accesibilidad del PDF	208
Exportar a PDF	210
ARIA, el aliado del HTML accesible.....	213
Qué es ARIA.....	214
Roles	219
Estados y propiedades	221
Buenas prácticas en ARIA	232
Cómo revisar ARIA.....	235
Las técnicas específicas de ARIA	241
Single Page Applications (SPA) accesibles.....	243
Qué es una SPA.....	244

Mensajes de estado	245
Título de página dinámico	246
Control del foco de teclado	247
Recomendaciones finales	250
Sistemas de diseño accesibles	251
Qué es un Sistema de diseño	252
Cómo diseñar y desarrollar un sistema de diseño accesible	253
Documentar el sistema de diseño.....	256
Cuidar la escritura	262
La importancia de las palabras	263
La comunicación clara.....	264
Redactar en lenguaje claro.....	266
Lectura fácil o lenguaje fácil.....	270
El lenguaje inclusivo	277
Herramientas de validación.....	281
Audit Tool WCAG 2.2.....	282
Herramientas de validación global automática	283
Herramientas de ayuda en la validación manual	289
Herramientas de validación de criterios.....	291
Herramientas de trabajo	299
Lectores de pantalla	300
Reproductores de audio y vídeo accesibles.....	304
Simulación del acceso con discapacidad visual.....	305
Documentación de traspaso de diseño a desarrollo.....	306
Resúmenes y esquemas	308
Diagramas conceptuales de las WCAG 2.2	309
Principios, Pautas y Criterios de Conformidad	314
Criterios de conformidad por niveles.....	320
Criterios para creadores de contenido	323
Criterios para diseñadores UX/UI	325
Criterios para desarrolladores	327
Glosario.....	329
Índice global.....	332

Revilla Muñoz, Olga; Carreras Montoto, Olga

Accesibilidad Web. WCAG 2.2 de forma sencilla. / Revilla Muñoz, Olga; Carreras Montoto, Olga – Madrid: Itákora Press, 2024 – 366 p.: 46 il.; 25.4 cm.

Copyright © 2024 Itákora Press. Todos los derechos reservados.

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



Olga Revilla Muñoz
Itákora

Durante más de dos décadas he trabajado como consultora y profesora en Diseño de Interacción, Experiencia de Usuario y Accesibilidad.

He dirigido y participado en una amplia gama de proyectos, abarcando distintos sectores, dimensiones y niveles de complejidad, tanto en grandes empresas consolidadas como con startups emergentes.

Me interesa la aplicación de la innovación y el diseño para la mejora de la sociedad en todas sus vertientes.

[LinkedIn](#) · [Twitter](#) · [Sitio web](#)



Olga Carreras Montoto
Usable y accesible

Soy consultora independiente de accesibilidad digital con 20 años de experiencia, autora del blog Usable y accesible.

Realizo auditorías de accesibilidad en el sector público y privado. Imparto formación en empresas, organismos públicos y en diversos postgrados universitarios.

He publicado otras guías, como "Aplicaciones móviles accesibles", "PDF accesibles con Adobe Acrobat Profesional" o "Documentos PowerPoint accesibles".

Soy miembro del IAAP y miembro de ASEPAU.

[LinkedIn](#) · [Twitter](#) · [Sitio web](#)

