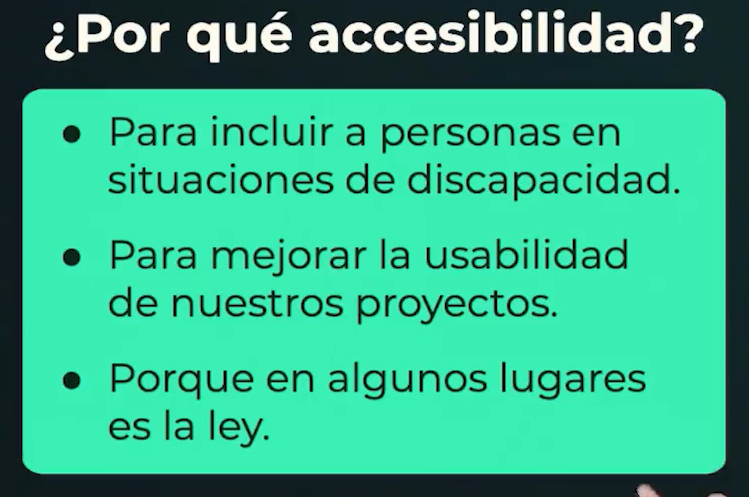
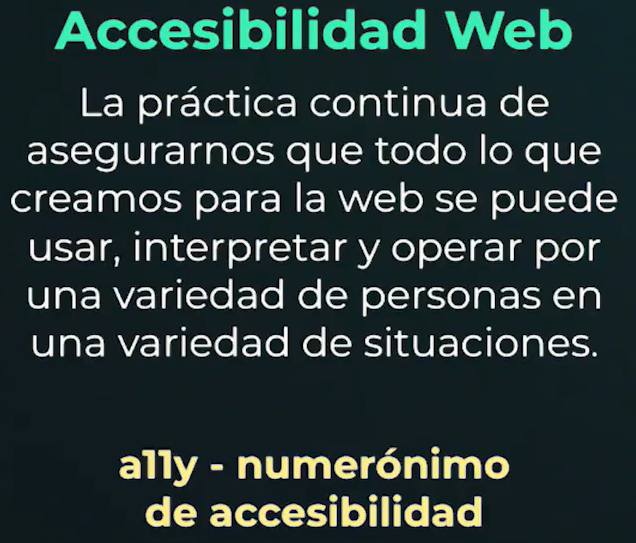
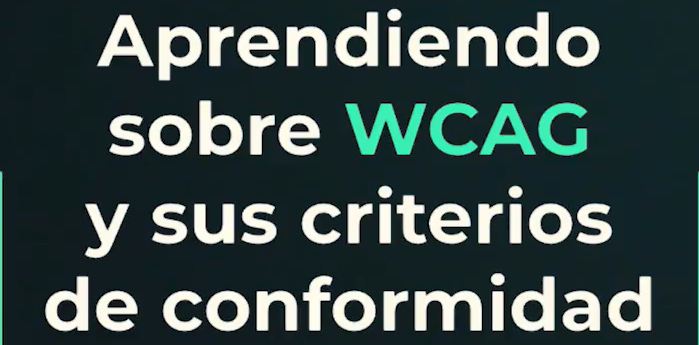
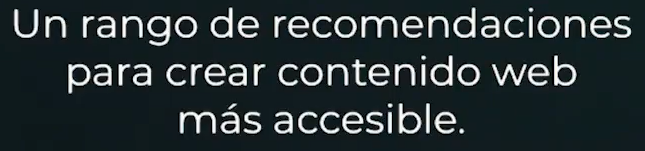
Curso de Accesibilidad Web

# Conociendo la importancia de la accesibilidad web







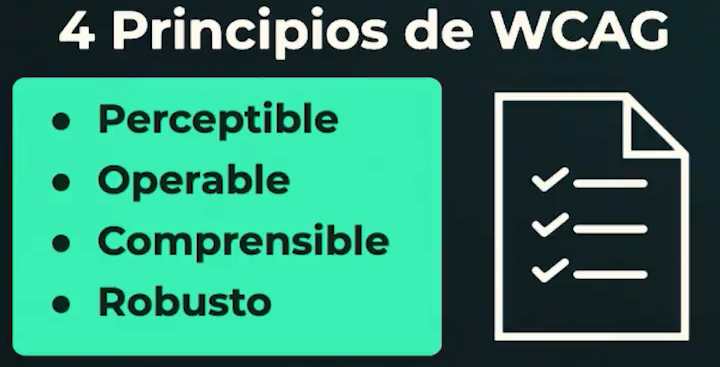
# Profundizando los 12 criterios de conformidad

Las pautas de accesibilidad de contenido web nos guían al momento de implementar la accesibilidad web. Las podemos usar durante todo el proceso de desarrollo de producto, desde que nace la idea para lo que queremos construir, durante el diseño de los estilos y las interfaces, hasta cuando escribimos código y hacemos pruebas de calidad. Nos señalan los puntos importantes para tener en cuenta con eso podemos construir páginas, productos y aplicaciones que le funcionan a todos nuestros usuarios sin importar su capacidades, dispositivos o preferencias.

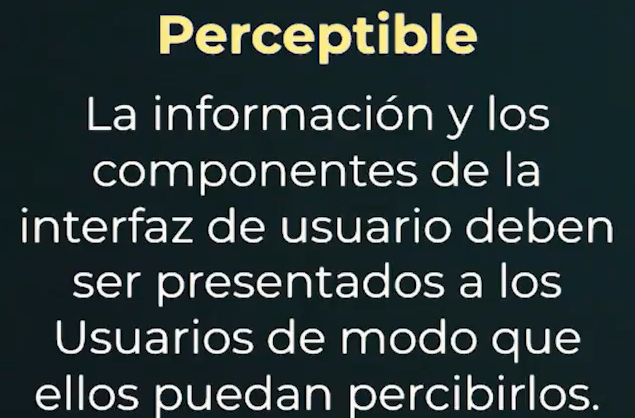
Aunque las pautas (referidas también por su nombre en inglés WCAG) son muy completas y siguen creciendo a medida que nuestras tecnologías crecen, es importante conocer los 4 principios que nos guían para poder tomar decisiones puntuales a nivel de nuestros productos. Dentro de cada principio encontramos los criterios de conformidad que se pueden interpretar acorde al nivel de WCAG que le quieres o debes aplicar a tu sitio web. Recuerda que los 3 niveles son:

A - nivel básico AA - nivel intermedio AAA - nivel avanzado al cual debemos apuntar si trabajamos en servicios del gobierno, universidades o servicios esenciales de la vida cotidiana.

En esta lectura vamos a profundizar sobre los criterios de conformidad para tener una idea más clara de que podemos encontrar dentro de WCAG. Podemos empezar por el primer principio:



**1. Perceptible** Los criterios de conformidad bajo este principio se enfocan en asegurar que todo el contenido que es importante se pueda interpretar de varias maneras. Aquí nos queremos

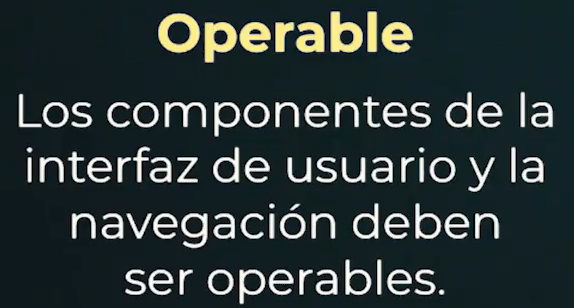


1.1 Asegurar que estamos proporcionando alternativas textuales para todo contenido no textual.  
1.2 Proporcionar alternativas para los medios tempodependientes (como videos o audios, queremos ofrecer opciones como subtítulos o guiones).

1.3 Crear contenido que se pueda presentar de diferentes formas sin perder información o estructura.

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

**2. Operable** Cuando hablamos de ser operable, queremos asegurarnos que nuestro sitio es fácil de usar y navegar, que se puede navegar de diferentes maneras y con mouse y/o teclado



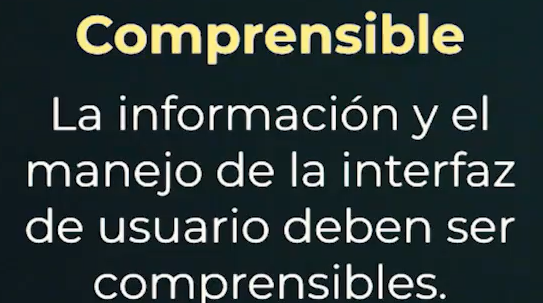
2.1 Proporcionar acceso a toda la funcionalidad mediante del teclado

2.2 Darle a las usuarias suficiente tiempo para leer y usar el contenido (tener esto en cuenta cuando hacemos time outs, comunicarlos y dar opciones para pedir más tiempo si es necesario)

2.3 Tener mucho cuidado de no usar elementos que brillan o se mueven muy rápido ya que pueden provocar ataques, espasmos o convulsiones

2.4 Asegurarnos que nuestros usuarios pueden navegar, encontrar contenido y determinar dónde se encuentran en nuestros sitios

**3. Comprensible** Hacer que nuestro sitio web sea comprensible nos asegura que un rango de personas lo pueden usar desde la persona que vive con una discapacidad mental que necesita patrones comunes para saber cómo usar la web hasta la persona que va de carrera y necesita consultar algo rápido



3.1 Tener en cuenta los tamaños de texto y contraste de colores para que los textos resulten legibles y comprensibles

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

3.3 Dar instrucciones para evitar errores y oportunidades para corregirlos cuando ocurren

**4. Robusto** Cuando hacemos productos que funcionan en muchos lados, abrimos las posibilidades que cualquier persona los use sin importar su ubicación, máquina, navegador y mucho más

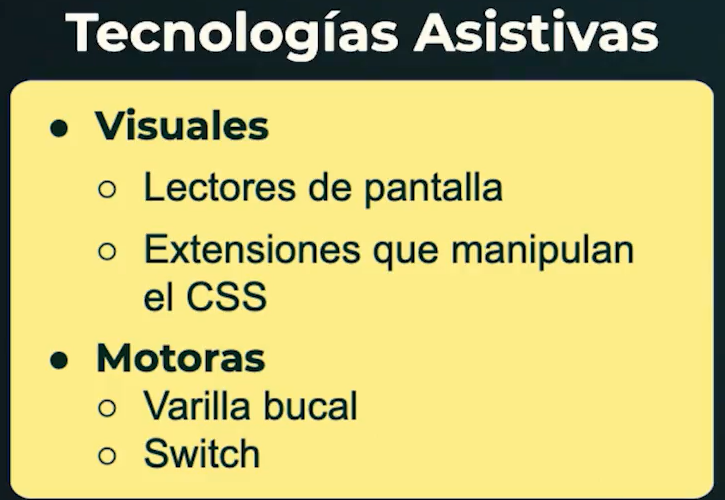


4.1 Maximizar la compatibilidad con las aplicaciones de usuario actuales y futuras, incluyendo las tecnologías asistivas

Así son los criterios de conformidad para la version 2.0 de WCAG. Te invito a entrar a la documentación y leer sobre que es necesario a cada nivel. ¿Hay alguno que te sorprende? ¡Cuéntame en los comentarios!

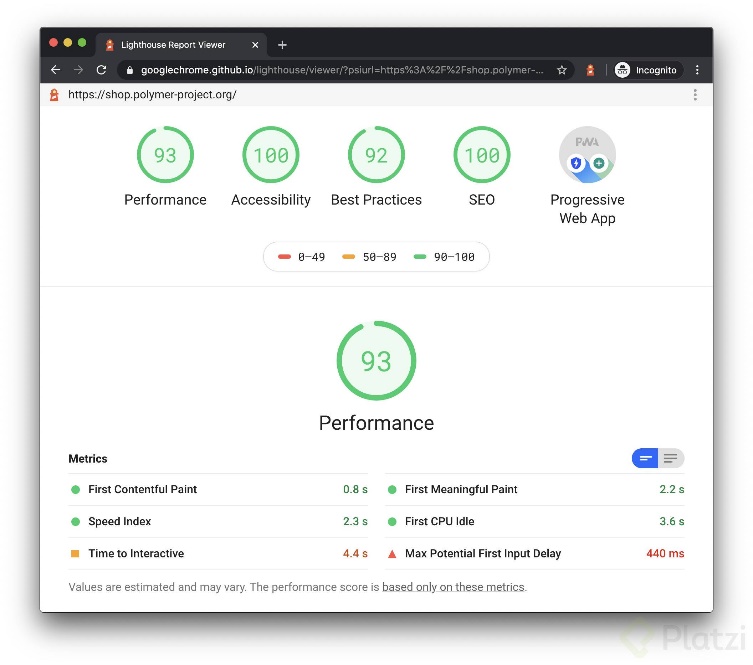
<http://www.sidar.org/traducciones/wcag20/es/>

# Tecnología Asistivas - cuáles son y cómo funcionan

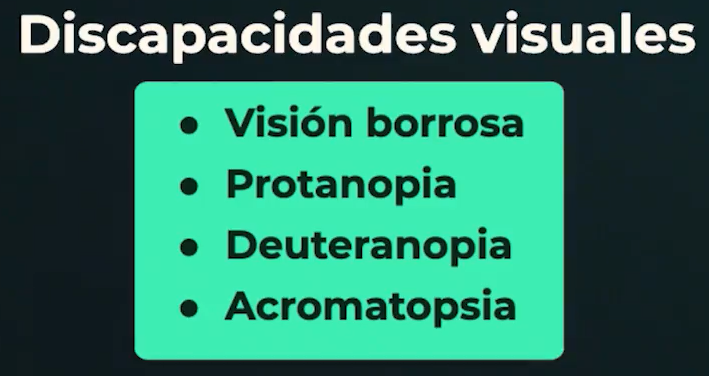


# Pruebas automáticas con Lighthouse

Lighthouse es una herramienta automatizada de código abierto diseñada para mejorar la calidad de tus apps web. Puedes ejecutarla como una extensión de Chrome o desde la línea de comandos.



# Pruebas con simuladores de discapacidades visuales



<https://chrome.google.com/webstore/detail/nocoffee/jjeeggmbnhckmgdhmgdckeigabjfbddl>

# Pruebas manuales con tu teclado

El teclado es muy importante en la accesibilidad web porque las tecnologías asistivas tienden a conectarse con el teclado para darle a sus usuarios diferentes maneras de navegar cuando no tienen acceso a un mouse o un teclado como nosotros.

Los elementos que reciben foco de teclado son elementos interacctivos:

* Enlaces: nos llevan a otra página, son para la navegación.
* Botones: nos ayudan a interactuar con la página de alguna manera.
* Formularios: requieren nuestra interacción para llenar nuestros datos.

Los div no reciben foco porque son un elemento presentacional, no esperan ninguna interacción a menos de que tú lo programes.

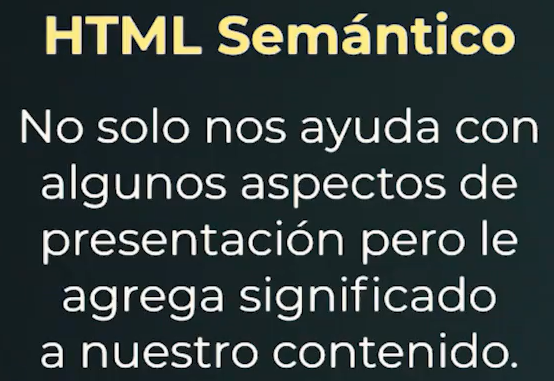
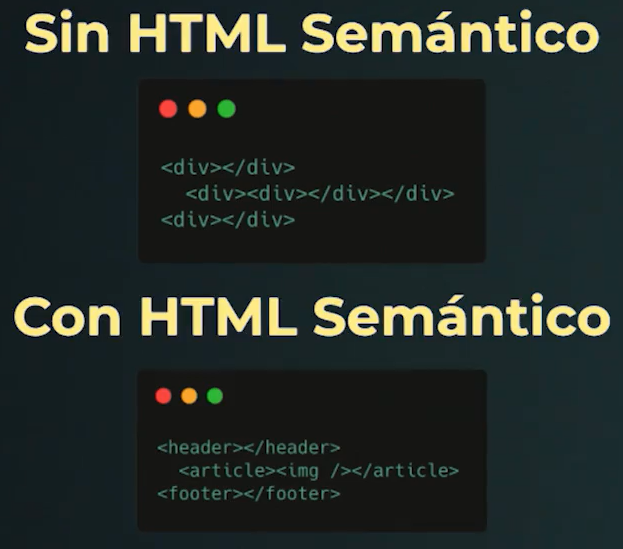
# Pruebas manuales con lectores de pantalla

* Los lectores de pantalla han abierto a la web a diversas personas que antes no podían usar los sitios web
* Son Software que mapea las páginas que usamos y las lee en voz alta. Usadas normalmente por personas que tienen incapacidades visuales.

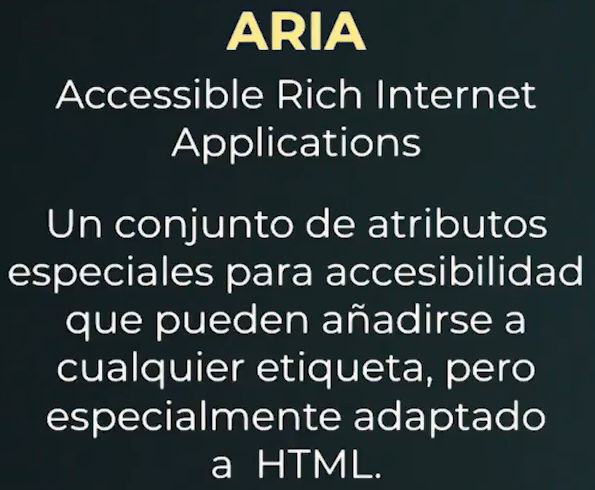
<h3>Combinaciones de lectores de pantalla y navegadores</h3>

* Es importante tener en cuenta cuales lectores podemos usar de manera compatible con los diferentes navegadores web
* **NVDA** → Mozilla Firefox
* **JAWS** → Microsoft Internet Explorer (EDGE)
* **VoiceOver** → Sfari
* **ChromeVox** → Google Chrome
* Apartado de Chorme DevTools → En **Elements** → **Accessibility** → **Accessibility Tree**

# ¿Qué es el HTML semántico y por qué es importante?

# ARIA - Accessible Rich Internet Applications

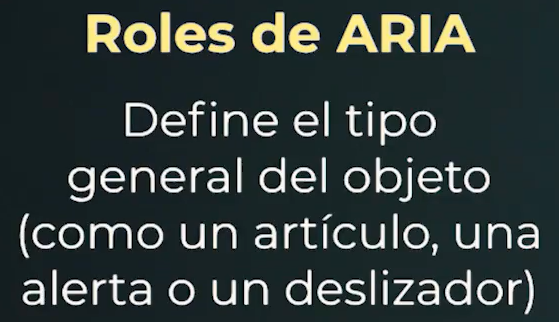
 

ARIA ( Accessible Rich Internet Applications ): Un conjunto de atributos especiales para accesibilidad que pueden añadirse a cualquier etiqueta, pero especialmente adaptado a HTML.

Fue creado por la W3C, lo crearon para que podamos comunicar cambios especiales en nuestras aplicaciones.

\*\*Atributos de ARIA: \*\*

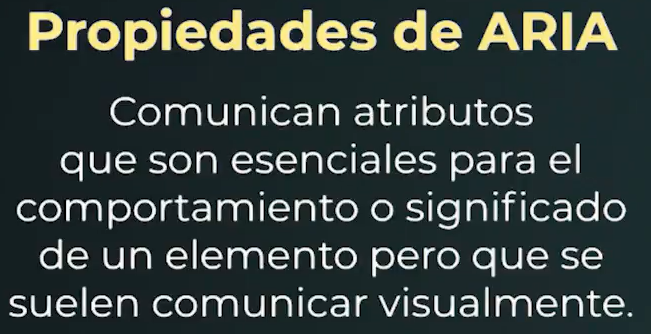
* Roles
* Propiedades
* Estados

ARIA roles

Roles de ARIA: Define el tipo general del objeto (como un artículo, una alerta o un deslizador)

Los roles le comunican al navegadores cuales son las interacciones que debería esperar y cómo se va a usar este objeto en nuestro proyecto. Se usan en situaciones muy específicas, es mejor depender del HTML semántico para comunicar los roles.

ARIA properties

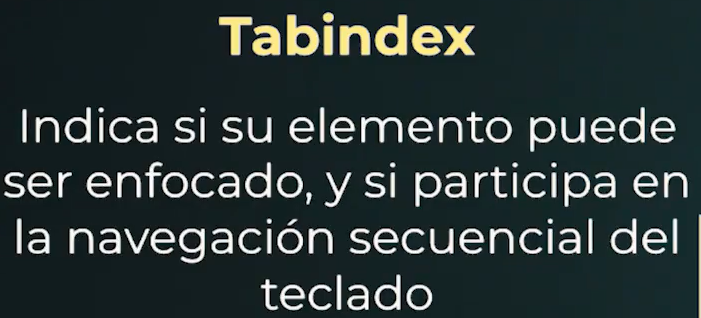
 Atributo: “aria-label”

ARIA states

Estados de ARIA: Comunican estados y cambios de estados en elementos que se suelen comunicar visualmente.

Añadimos a la etiqueta de <li> el atributo: aria-hidden=“true” para que el lector de pantalla solo lea los elementos visibles en el estado actual de la página.

# Manejando el foco del teclado - el carousel



Tabindex: Indica si su elemento puede ser enfocado, y si participa en la navegación secuencial del teclado.

Valores de Tabindex:

* Valor negativo(-1): El elemento debe ser enfocado, pero no debe de ser accesible a través de la navegación.
* Valor positivo (>0): Debe poder ser enfocado y su orden relativo es definido por el valor del atributo.
* Valor de 0: Debe ser enfocado y ser accesible a través de la navegación secuencial del teclado, pero su orden relativo es definido por convención de la plataforma.