Engineer

axe

el motor de accesibilidad con el que se audita este sitio

Qué es

axe, publicado como biblioteca JavaScript de código abierto axe-core, es un conjunto de reglas automáticas de accesibilidad que se ejecuta dentro de una página web, inspecciona lo renderizado por el navegador e informa de cada elemento que incumple una regla. Lo mantiene Deque Systems, empresa de accesibilidad, bajo Mozilla Public License 2.0. Otras herramientas se apoyan en él: la categoría de accesibilidad de Lighthouse de Chrome pondera las auditorías según las clasificaciones de impacto de axe, y la propia guía de Playwright ejecuta axe mediante un pequeño paquete complementario.

El motor es un único script. Un ejecutor de pruebas o una extensión lo carga en la página y llama a `axe.run` ; entonces recorre el documento, incluidos marcos anidados y shadow DOM, aplicando todas las reglas cuyas condiciones se cumplan. Como lee después de que el navegador haya dispuesto la página, comprueba cosas que el código fuente no revela: si el texto es visible, sobre qué color aparece y si un elemento está oculto. Por eso algunas reglas requieren un navegador real. El proyecto afirma que el contraste no funciona en JSDOM, el documento simulado de muchos entornos de pruebas unitarias.

Cada regla lleva etiquetas. Una señala la versión y nivel WCAG, como `wcag2a` , `wcag2aa` , `wcag21aa` o `wcag22aa` , o indica `best-practice` cuando ningún criterio la exige. Otras relacionan reglas con EN 301 549, Section 508 o las reglas W3C de pruebas de conformidad. Mediante esas etiquetas decide el equipo qué comprueba cada ejecución.

El resultado no es un único aprobado o suspenso, sino cuatro listas:

- `violations` : elementos que incumplieron una regla, cada uno con impacto menor, moderado, grave o crítico, un resumen breve de la solución y un enlace a una explicación más extensa.
- `passes` : elementos que la superaron, lo que permite ver que una regla llegó a ejecutarse.
- `incomplete` , también llamada «needs review» en la documentación: elementos que el motor no pudo aprobar ni rechazar con certeza y que debe examinar una persona.
- `inapplicable` : reglas sin nada que probar, como las de vídeo en una página sin vídeo.

Utilice el concepto en una decisión

Aplique «¿Qué es axe?» a una decisión real. Podemos ayudarle a relacionar el concepto con el sistema que está construyendo o comprando. Interfaces que cargan rápido, funcionan en el teléfono y se comportan igual en los navegadores que utilizan sus clientes.

Explore nuestros servicios Póngase en contacto →

Compruebe los supuestos detrás del término

Utilice la explicación y los trabajos enlazados para formular preguntas sobre su proyecto. Podemos examinar los supuestos con usted antes de elegir una herramienta. Dirección del desarrollo integral de mapas GIS — cartografía integral, de la base de datos al navegador

Dirección del desarrollo integral de mapas GIS Póngase en contacto →

Por qué importa

axe ayuda a decidir qué partes de la accesibilidad delegar en una máquina y si un fallo debe impedir una versión. Dos propiedades lo hacen adecuado. Su ejecución es barata y permite comprobar cada build en vez de auditar una vez al año. Además, sus autores declaran el objetivo de cero falsos positivos, salvo errores del propio motor: cualquier duda va a la lista de revisión, no a la de infracciones. Una comprobación que da falsas alarmas acaba desactivada; una que solo informa de lo demostrable puede detener un build sin que nadie lo considere injustificado.

El límite es la cobertura. Deque dice que axe-core encuentra automáticamente una media del 57 por ciento de los problemas WCAG y devuelve resultados inciertos para su revisión manual. La cifra describe los problemas detectados, no la proporción de criterios de conformidad que puede evaluar una máquina.

Lo que queda fuera del alcance es sobre todo el juicio. axe puede confirmar que una imagen tiene texto alternativo, que un botón tiene nombre y que un campo de formulario tiene etiqueta. No puede determinar si el texto describe la imagen, si el nombre dice qué hace el botón o si la página tiene sentido al leerse en voz

alta y en orden. Esas preguntas necesitan una persona con lector de pantalla y teclado, y el tiempo que una máquina ahorra en las comprobaciones mecánicas es tiempo que esa persona puede dedicarles.

Elegir etiquetas tiene su propia contrapartida. Ejecutar solo las WCAG da un resultado directamente vinculable a una declaración de conformidad, y nada más. Añadir buenas prácticas detecta problemas que WCAG no nombra pero el usuario encuentra: contenido fuera de todas las regiones o una página sin encabezado principal. El coste son algunos fallos que una auditoría de conformidad no citaría.

Cómo se aplica en la práctica

Dos páginas del sitio muestran qué se confía al motor.

En la batería de una plataforma acompaña a cuatro capas de pruebas. Un control axe-core exige cero infracciones WCAG 2.0 y 2.1 A y AA, convirtiendo la accesibilidad en fallo de build, no en hallazgo meses después. La clave es el momento: corrige el defecto quien acaba de escribirlo, mientras el cambio sigue abierto.

En este sitio, los créditos describen auditorías de páginas representativas, una por plantilla en cada idioma, contra WCAG 2.2 A y AA en ambos esquemas de color, como parte del control obligatorio y no como informe aparte. Probar ambos permite que un fallo de contraste solo visible en oscuro bloquee el build. El muestreo supone que las páginas de una plantilla comparten estructura; así 38 páginas representaron un sitio de 2.268 en octubre de 2026. Las reglas automatizadas no cubren todos los juicios que exige WCAG; por eso el registro se mantiene por criterio, no como una nota de aprobado.

Dónde surgen los problemas

La mayoría son fallos de confianza: una ejecución comprobó menos de lo aparente o se interpreta el resultado más allá de lo que dice.

La lista de etiquetas omite reglas. Las etiquetas no son acumulativas. `wcag2aa` solo contiene las reglas para los criterios de nivel AA de WCAG 2.0, por lo que limitar una ejecución a ella omite todas las reglas de nivel A; la documentación de la API dice expresamente que hacen falta tanto `wcag2a` como `wcag2aa` . Del

mismo modo, `wcag21aa` solo contiene reglas de los criterios que añadió 2.1, y una ejecución limitada a etiquetas WCAG nunca ejecuta una regla `best-practice`. Las descripciones de las reglas enumeran las de WCAG 2.2 como desactivadas por defecto, por lo que una ejecución sin etiquetas las omite; nombrar su etiqueta las activa. Enumere todas las etiquetas que deba cubrir la ejecución y conserve la lista junto a la declaración que respalda:

```
const results = await axe.run(document, {
    runOnly: {
        type: "tag",
        values: ["wcag2a", "wcag2aa", "wcag21a", "wcag21aa", "wcag22aa"],
    },
});
```

Una regla ausente de las cuatro listas no se ejecutó. Por eso conviene leerlas cada vez que cambie la configuración.

Una ejecución limpia se interpreta como página accesible. Cero infracciones significa que las reglas ejecutadas no pudieron demostrar un fallo. No dice si el texto alternativo es correcto, el orden de foco sensato o una persona con lector puede terminar la tarea. Tómela como base y programe las comprobaciones que no puede hacer.

Se descarta la lista de revisión. Un control que solo falla por infracciones deja pasar `incomplete` sin que nadie lo vea. Son elementos examinados y dejados sin decidir, precisamente los que más necesitan atención humana. Envíelos a alguien que los lea o haga que también detengan el build.

La página se prueba en un solo estado. El motor comprueba lo que está renderizado en el momento en que se ejecuta. Un menú cerrado, un diálogo sin abrir, un mensaje de error que aparece solo tras un envío incorrecto y el otro esquema de color quedan fuera de una ejecución contra la página tal como se carga por primera vez. Lleve la página a cada estado importante y ejecute de nuevo el motor en cada uno.

El motor corre donde no se dibuja nada. Un documento simulado carece de disposición y color calculado. El proyecto enumera el contraste entre las reglas que no funcionan allí. Las pruebas unitarias con axe aún detectan problemas estructurales; el contraste y lo visual pertenecen a un navegador real.

Se desactiva una regla para publicar. Apagarla para toda la ejecución porque falla un componente oculta sus infracciones en los demás, incluidas páginas futuras. Excluya solo ese elemento, con una nota que explique por qué y cuándo debe retirarse la excepción.

Lecturas adicionales

- axe-core — el código fuente del motor, sus principios declarados y la lista de navegadores compatibles.
- Descripciones de reglas — cada regla con su impacto y etiquetas, agrupadas por versión WCAG, buenas prácticas, AAA, experimentales y obsoletas.
- API de axe-core — las etiquetas, la opción `runOnly`, cómo incluir o excluir partes de una página y la estructura de las cuatro listas de resultados.
- Pruebas de accesibilidad — la guía de Playwright para ejecutar axe dentro de las pruebas de navegador, incluido el análisis de una página después de interactuar con ella.



Glosario

Elija según sus restricciones reales

Describa el flujo de trabajo, las interfaces existentes y los usuarios que necesitan el cambio. Podemos definir una entrega comprobable y un traspaso que permita a su equipo mantener lo entregado.

Vea cuánto cuesta Póngase en contacto →

Haga práctica la terminología

Pregunte cómo se aplica «¿Qué es axe?» a su organización. Basta un ejemplo concreto de su sistema actual para iniciar la conversación.

Explore nuestros servicios

welcome@engineer.company +45 50 10 61 84 Copenhague, Dinamarca

Llévese esta página

Tarjeta de contacto

Creado con esmero y cariño por D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Derechos de autor © 2025 – actualidad · Engineer ApS · Todos los derechos reservados.

Impreso desde https://engineer.company/es/glossary/axe/