

---

# Accesibilidad

---

PID\_00253481

Benjamin Hawkes-Lewis  
Tom Hughes-Croucher

---

Tiempo mínimo previsto de lectura y comprensión: 3 horas

---



---

Universitat  
Oberta  
de Catalunya

---



# Índice

<b>1. Conceptos básicos de accesibilidad</b>	5
1.1. ¿Qué es la accesibilidad?	5
1.2. ¿Por qué es importante la accesibilidad?	6
1.2.1. Los aspectos legales del accesibilidad	7
1.2.2. Mercados potenciales	8
1.2.3. Motores de búsqueda	8
1.2.4. Ética y marcas	9
1.3. Diseñar pensando en la accesibilidad	9
1.4. Requisitos de interoperabilidad	10
1.5. Características de una página web accesible	11
1.5.1. Estructura semántica	11
1.5.2. Contenido alternativo	12
1.5.3. Definir la interacción	14
1.6. Estándares de accesibilidad	18
1.6.1. Directrices de accesibilidad del contenido web 1.0	18
1.6.2. Directrices de accesibilidad del contenido web 2.0	19
1.6.3. Otros estándares	20
Resumen	21
Preguntas de repaso	21
<b>2. Pruebas de accesibilidad</b>	22
2.1. ¿Cuándo se deben realizar las pruebas?	23
2.2. Entender los requisitos	24
2.2.1. Requisitos externos	24
2.2.2. Detalles de conformidad	24
2.2.3. Superar las expectativas	25
2.2.4. La importancia de la interfaz de usuario	27
2.2.5. Personajes con discapacidades	27
2.2.6. Elegir un estándar de accesibilidad	28
2.2.7. El espíritu de la ley	29
2.3. ¿Quién debe hacer las pruebas?	30
2.4. Pruebas avanzadas	31
2.4.1. Comprobadores de accesibilidad semiautomatizados	32
2.4.2. Inspectores estructurales	32
2.4.3. Simulación y uso de tecnologías de asistencia de usuario final	34
2.4.4. Inspección detallada	35
2.5. Pruebas de usuario	40
2.5.1. Encontrar verificadores	40
2.5.2. Consideraciones prácticas	41
2.5.3. Elegir las tareas	42

2.5.4. Interpretar los resultados .....	43
2.6. Comunicar los resultados de las pruebas de accesibilidad .....	44
Resumen .....	44
Preguntas de repaso .....	45

# 1. Conceptos básicos de accesibilidad

Tom Hughes-Croucher

sitio web, la accesibilidad (conseguir que todo el mundo lo pueda utilizar, sean cuales sean sus capacidades o discapacidades) debería ser una de las preocupaciones más importantes. En esta asignatura, hasta ahora la accesibilidad ha estado siempre implícita en todos los ejemplos que hemos visto, incluso aunque no os dierais cuenta. En este apartado, sin embargo, hablaremos de ello explícitamente para que podáis entender bien qué es, por qué es importante, cómo se garantiza que los sitios sean accesibles y qué directrices existen para definir los sitios accesibles.

Antes de hablar concretamente de la accesibilidad en la web, empezaremos hablando de la accesibilidad en general; al fin y al cabo, la accesibilidad no es sólo una cuestión asociada a los sitios web; es un aspecto que, potencialmente, debe tener en cuenta todos los servicios, objetos o tecnologías con los que os encontréis.

## Nota

Tened en cuenta que un tema asociado sobre el que os deberíais informar es WAI ARIA\*: la iniciativa *Web Accessibility Initiative's Accessible Rich Internet Applications*, que es básicamente una metodología que permite la creación de aplicaciones con Ajax/JavaScript más accesibles. Podéis encontrar un artículo muy bueno de introducción al ARIA en dev.opera.com\*\*.

\* <http://www.w3.org/WAI/intro/aria>

\*\* <http://dev.opera.com/articles/view/introduction-to-wai-aria/>

## 1.1. ¿Qué es la accesibilidad?

Mirad a vuestro alrededor. Es muy posible que veáis a otras personas; si no es así, ¿por qué no vais a dar un paseo rápido? Probablemente lo disfrutaréis y os sentará bien. Todas las personas que veréis serán diferentes: algunas tienen el pelo castaño, otras no. Algunas tienen los ojos azules, otras no. Otras llevan gafas, otras no. No hay dos personas exactamente iguales. Algunas diferencias, como el color del pelo o de los ojos, son puramente estéticas; es decir, no afectan de ninguna manera importante a nuestra vida. Algunas diferencias, como el hecho de llevar gafas, sí que la afectan. La accesibilidad es algo muy sencillo, una filosofía, aunque en algunos países también forma parte de la ley.

La accesibilidad es tratar a todo el mundo igual, sean cuales sean sus capacidades.

Somos muy conscientes de que esta afirmación está abierta a interpretación. La mayoría de las discusiones sobre accesibilidad empiezan hablando sobre la discapacidad. Esto implica que las personas con alguna discapacidad merecen un tratamiento especial. Pero la accesibilidad no es eso; en realidad es un síntoma del modo como la gente ha construido tradicionalmente edificios, sitios web y muchas otras cosas.

Cuando se construyen cosas presuponiendo que todo el mundo es igual que nosotros, podéis estar seguros de que siempre funcionarán mal para algunas personas. La gente presupone que la accesibilidad es ayudar a las personas con alguna discapacidad, ya que la accesibilidad añadida es algo muy presente en nuestra sociedad. Por ejemplo, muchos edificios que se construyeron sólo con escaleras han incorporado de repente unas rampas baratas y muy feas. Pero la accesibilidad ha sido durante mucho tiempo una cuestión importante en el diseño militar. ¿Por qué? Pues porque a menudo es vital para la supervivencia; con unas fuerzas g muy elevadas, los pilotos de los aviones de combate no pueden hacer lo mismo que hacen cuando están en el suelo. Si los diseñadores no tuvieran en cuenta las necesidades de los pilotos en entornos tanto de gravedad alta como baja, habría muchos más accidentes de avión.

Pero ¿qué significa esto para los desarrolladores de sitios web? La respuesta rápida es que debéis intentar ser más conscientes de las necesidades de todo el público potencial que puede visitar vuestro sitio web. Una respuesta más larga implicaría, por vuestra parte, ser mínimamente conscientes de los diferentes niveles de capacidades que puede tener la gente y del uso que hacen de los ordenadores. Aplicando las técnicas que se describen en este apartado y en otros apartados relacionados podréis crear unos sitios web que funcionen bien con muchas formas de interacción. Vuestros sitios web deben poder ser utilizados por personas:

- Ciegas o con deficiencias visuales graves que escuchen los sitios web utilizando un lector de pantalla o que los lean en una pantalla braille.
- Miopes que aumenten el tipo de letra hasta el 200% de su tamaño.
- Con discapacidades motrices que no puedan utilizar las manos para manipular un ratón y que, por lo tanto, utilicen un dispositivo de señalización para manipular el teclado o incluso un puntero visual para manipular el sitio web.
- Utilicen ratones de bola u otros tipos de sistemas de control del ordenador menos habituales.

De momento, no es necesario que os preocupéis por los detalles concretos de estas interacciones; ya los iremos viendo paso a paso.

## 1.2. ¿Por qué es importante la accesibilidad?

La accesibilidad es importante por una razón principal y por muchas otras razones pequeñas. La principal es que todos somos diferentes pero todos tenemos el

mismo derecho a utilizar los sitios web, pero hay muchas otras razones por las que deberíais tener en cuenta la accesibilidad a la hora de crear sitios web:

- En algunos países es obligatorio por ley.
- No se deben excluir clientes/visitantes potenciales de vuestro sitio web.
- Los sitios accesibles suelen estar mejor situados en los motores de búsqueda.
- Demostraréis una buena ética, algo que los clientes sabrán valorar.
- Si creáis los sitios web siguiendo los estándares web, no necesitaréis hacer prácticamente nada más para hacerlos accesibles, lo cual aporta muchas ventajas; además, también hay muchas similitudes entre los sitios más accesibles y los sitios más compatibles con los navegadores de teléfonos móviles, que es otra circunstancia que dificulta el uso de los sitios web, aunque por unas circunstancias diferentes. De hecho, se han realizado algunos estudios de análisis de la relación entre la accesibilidad de la web y las mejores prácticas del desarrollo de la web móvil; podéis ver la página del WAI “Web Content Accessibility and Mobile Web”\*, donde encontraréis más información sobre esto.
- Las técnicas que ayudan a la gente con discapacidades benefician a todos los usuarios.

\* <http://www.w3.org/WAI/mobile/>

Ahora pasaremos a ver algunos de estos puntos más detalladamente.

### 1.2.1. Los aspectos legales del accesibilidad

#### Nota

Es muy importante entender los fundamentos de los aspectos legales, pero si no es que sois abogados y sabéis bien de qué estáis hablando, sería necesario que fuerais con mucho cuidado a la hora de dar una opinión sobre cuestiones legales.

En el Reino Unido, según la DDA\*, es ilegal discriminar a las personas discapacitadas a la hora de contratar trabajadores y ofrecer servicios o educación. La discriminación se define como no hacer “ajustes razonables” para dar apoyo a todo el mundo, sean cuales sean sus capacidades o discapacidades. Esto se aplica, evidentemente, a los servicios y la educación disponibles a través de sitios web.

\* <http://www.direct.gov.uk/en/DisabledPeople/index.htm>

En Estados Unidos y en la Unión Europea también hay exigencias para los sitios web gubernamentales. En Estados Unidos, los sitios web del gobierno federal (y de algunos gobiernos estatales) deben cumplir la Sección 508\*. La Sección 508 es un documento que intenta definir los requisitos mínimos para

\* <http://www.section508.gov/>

conseguir la accesibilidad. La Sección 508 no afecta sólo a los sitios web, sino que también hace referencia a cualquier otra tecnología que puedan utilizar los empleados federales. En Europa, la Comisión Europea ha reconocido la web Accessibility Initiative (WAI) de W3C y recomienda su uso a todos los Estados miembros. La WAI\* publica directrices para sitios web, fabricantes de herramientas de creación de webs y navegadores web (por ejemplo, el WCAG, del que hablaremos más adelante).

\* <http://w3.org/WAI>

### 1.2.2. Mercados potenciales

Si creáis sitios web (o cualquier otra cosa) para un tipo de personas concretas, estaréis excluyendo otros tipos de personas incluso aunque no os deis cuenta de ello, y toda esta gente excluida puede representar fácilmente una cuota de mercado importante (o incluso mayoritaria). En el año 2000, la cadena de supermercados Tesco del Reino Unido puso en marcha un proyecto para crear una versión diferente de su sitio de comestibles en línea pensado específicamente para las personas con discapacidades visuales. Julie Howell, de RNIB, comentó que “El trabajo realizado por Tesco.com para conseguir que su servicio de comestibles fuera más accesible para las personas ciegas ha dado como resultado un aumento de los ingresos de 13 millones de libras anuales, unos ingresos que no estaban al alcance de la compañía cuando el sitio web no era accesible para los clientes ciegos”. Así pues, si Tesco no hubiera pensado en la gente con discapacidades visuales, no habría llegado nunca a un mercado de clientes con un valor de 13 millones de libras como mínimo.

La lección que debemos extraer es que todo el mundo, sean cuales sean sus capacidades, necesita los mismos servicios: comestibles, taxis, electricidad, etc. y deben poder disfrutar de las mismas cosas: películas, música, bares, etc. Presuponer que la situación vital personal de alguien hace que cambien sus capacidades o sus deseos de participar en todos los aspectos de la sociedad se ha demostrado una y una otra vez como un gran error.

### 1.2.3. Motores de búsqueda

Los motores de búsqueda no son personas. A menudo, cuando alguien crea un sitio web lo hace sin tener en cuenta cómo se podrá encontrar en Google, Yahoo!, etc. Los motores de búsqueda son sólo programas informáticos y para indexar la página sólo pueden utilizar la información que pueden entender. Eso les hace ser muy similares a los lectores de pantalla que pueden utilizar las personas con alguna discapacidad visual.

El ejemplo más obvio de los efectos que esto tiene sobre el diseño de webs son las imágenes. Los ordenadores muestran imágenes a partir de una lista con los colores que deben tener todos los píxeles y envían esta información al monitor. Si po-



néis en una página web una imagen que contiene texto, como por ejemplo un logotipo, el ordenador no tiene ni idea de qué dice el texto y ni siquiera sabe que la imagen contiene texto. En HTML, el elemento de imagen contiene una manera de describir con texto el contenido de una imagen, que es el atributo `alt`. Debéis ofrecer un texto para describir todas las imágenes no decorativas de vuestro sitio, y nunca deberíais representar párrafos completos como imágenes (o Flash); las personas ciegas y los motores de búsqueda no tendrán ni idea de qué dice el texto. Como resultado, la posición en un motor de búsqueda (es decir, la facilidad de encontrar vuestro sitio web con motores de búsqueda como Google) se verá afectada y quedaréis innecesariamente fuera de un mercado muy valioso.

#### **1.2.4. Ética y marcas**

Todo el mundo debería tener en cuenta la accesibilidad, pero esto no quiere decir que todo el mundo lo haga. Si tenéis en cuenta la accesibilidad, estaréis actuando en beneficio de la comunidad. Eso es algo de lo que os podréis enorgullecer; si demostráis que tenéis en cuenta a todos los que integran nuestra sociedad, conseguiréis mejorar mucho vuestra imagen de marca. Como profesionales, nuestra obligación es intentar conseguir los resultados de la mejor calidad posible. En una sociedad que nos valora como personas individuales, es importante no excluir a alguien por el hecho de que tenga unas necesidades diferentes.

Si seguís unos principios responsables y demostráis de una manera genuina que los aplicáis, podréis crear una imagen de marca extraordinariamente positiva. Las compañías que demuestran que se preocupan por sus clientes los fidelizarán mucho más que las que no lo hacen.

### **1.3. Diseñar pensando en la accesibilidad**

La clave para la accesibilidad es pensar en un problema y decidir solucionarlo para más de un tipo de usuarios.

Si pretendéis tratar la accesibilidad como algo que ya añadiréis una vez que hayáis acabado todos los demás trabajos, sólo conseguiréis tener algo mal hecho y mal integrado. Necesitaréis más tiempo, no funcionará bien y tendrá un aspecto horrible.

La mejor manera de conseguir una solución bien hecha es diseñar la web teniendo en cuenta todas las necesidades desde el primer momento. Eso no quiere decir que no podáis cambiar vuestros planes o añadir algo que os hubiera pasado por alto, pero deberíais intentar ser conscientes de cuál es el problema general que pretendéis solucionar con vuestro diseño. En el caso de los sitios web, eso implica crear una solución usable por todos los usuarios, incluyendo a quienes quizá no pueden utilizar un ratón, un teclado, un monitor, etc.

## 1.4. Requisitos de interoperabilidad

Los requisitos de interoperabilidad pueden variar mucho de una situación a otra:

- A menudo se introducen nuevas tecnologías que no incorporan funciones de accesibilidad.

### Ejemplo

El nuevo conector Silverlight de Microsoft no presenta la información a través de los API de accesibilidad utilizados por los lectores de pantalla y otras tecnologías de asistencia, aunque sí que está previsto incorporar funciones de accesibilidad en el futuro.

- En los casos en los que estas funciones de accesibilidad están teóricamente disponibles, puede ser necesario un cierto tiempo para que las tecnologías de asistencia las utilicen.

### Ejemplo

Los lectores de pantalla más nuevos funcionan mucho mejor con las actualizaciones con JavaScript de las estructuras HTML que los lectores de pantalla más antiguos.

- Incluso cuando ya están establecidas desde hace mucho tiempo, las características de accesibilidad pueden continuar difiriendo entre las distintas plataformas.

### Ejemplo

El conector Adobe Flash Player expone desde hace mucho tiempo la información en el API de accesibilidad de Windows, pero no en el equivalente de Apple o GNOME.

- También suele haber un cierto retraso entre la llegada de la tecnología y su distribución generalizada. Actualmente, los navegadores y los conectores suelen ser gratuitos, mientras que las tecnologías de asistencia mayoritarias pueden llegar a ser muy caras.

### Ejemplo

Uno de los lectores de pantalla más populares es JAWS, de Freedom Scientific, para Windows. Prácticamente cada año aparece una versión nueva. El precio de JAWS Profesional para el consumidor final es de 1.095 dólares, y aunque os gastéis 200 dólares adicionales para disfrutar de un contrato de mantenimiento de software para las dos versiones siguientes, las actualizaciones seguirán costando 500 dólares o más. Por lo tanto, a pesar de que la última versión es la 9, todavía encontraréis a muchos usuarios de JAWS que utilicen versiones anteriores.

Cuando queráis construir sitios para la web pública, por lo tanto, deberéis incorporar una cierta interoperabilidad con una combinación muy variada de usuarios y tecnologías. Hay cuatro enfoques posibles:

- Mejorar progresivamente vuestro sitio web e ir comprobando la compatibilidad a medida que vayáis avanzando.
- Permitir que todos los usuarios puedan desactivar las mejoras problemáticas.
- Ofrecer versiones alternativas con el mismo contenido o las mismas funciones.
- Avisar a los clientes de las tecnologías que deben incorporar y dar ejemplos de compañías que sean compatibles con estas tecnologías.

En las intranets, la compatibilidad con versiones anteriores y la variedad son un problema mucho más pequeño. Una organización concreta puede garantizar, por ejemplo, que todos los empleados con discapacidades tengan acceso a una tecnología de asistencia que acepte bien el DHTML. En estas circunstancias, y con las pruebas pertinentes con la tecnología de asistencia ofrecida, sería razonable tomar el JavaScript como mínimo.

Sin embargo, la compatibilidad con versiones posteriores y la compatibilidad entre plataformas sí que siguen siendo un problema, por lo que se deberían preferir las tecnologías abiertas y estándar sobre las tecnologías propietarias y no estándares.

Podrías desarrollar, por ejemplo, una aplicación de formación de la intranet para una gran corporación. Os han pedido que garantizéis que la aplicación sea accesible, pero no os han especificado ningún estándar al que os debáis adaptar. Habláis con el departamento de informática y descubrís que todo el mundo tiene la última versión de la Internet Explorer con el JavaScript habilitado y Flash instalado y activado, y también que todo el mundo dispondrá de la tecnología de asistencia moderna necesaria para poder utilizar todos estos programas. Aunque la compañía pase a una plataforma Unix, seguirá habiendo la tecnología de asistencia necesaria para el JavaScript, pero el texto y los controles en Flash sólo son accesibles desde Windows. Podrías hacer que el *script* y el Flash fueran una referencia obligatoria para vuestra aplicación. Pero decidís utilizar el Flash sólo para reproducir vídeo y construir los grupos de controles para el vídeo Flash a partir de los estándares web, ya que la tecnología de asistencia sólo puede acceder a los controles del Flash desde la plataforma Windows. Así, la aplicación continuaría siendo accesible aunque la compañía migrara a Unix.

Las políticas de TI de la organización pueden cambiar, y todos los esfuerzos mejor intencionados para hacer que las funciones JavaScript sean operativas y para explotar los grupos de funciones de accesibilidad de los conectores pueden fallar; por lo tanto, continúa siendo una buena idea realizar una mejora progresiva de la referencia tecnológica a partir de una capa HTML básica.

## **1.5. Características de una página web accesible**

En este subapartado explicaremos las funciones de accesibilidad diferentes de un sitio web; es decir, qué debe contener un sitio web accesible. Explicaremos cada una de ellas con detalle.

### **1.5.1. Estructura semántica**

Una de las bases de los estándares de la web es el uso de la estructura semántica en el HTML. La estructura semántica también es sumamente importante para la accesibilidad. Es así porque ofrece un marco de referencia para la información de la página. Cuando la gente no puede ver el estilo visual de la página,

la estructura semántica ayuda a explicarles varias cosas. Les puede indicar su posición en la jerarquía del documento y los modos como pueden interactuar con los diferentes elementos de la página; también ofrece énfasis en el contenido textual en los lugares adecuados.

Un buen ejemplo de la importancia de la estructura semántica de un documento para la accesibilidad es la navegación. Un menú de navegación bien estructurado es una lista de puntos. Podéis etiquetarlo como una lista HTML:

```
<ul>
  <li>Entrada del menú 1</li>
  <li>Entrada del menú 2</li>
  <li>Entrada del menú 3</li>
</ul>
```

Con unos menús de navegación estructurados como listas, los usuarios de lectores de pantalla podrán saber muy fácilmente que es una lista aunque no la puedan ver. Eso es así porque los lectores de pantalla les dicen que es una lista. Si no utilizáis el etiquetado de lista, entonces el lector de pantalla no tiene ninguna manera de saber que es una lista y no lo puede decir al usuario.

! Podéis encontrar más información sobre el uso de la semántica correcta en vuestro HTML en muchos de los apartados anteriores de esta asignatura, básicamente en los que hablan del HTML.

### 1.5.2. Contenido alternativo

Tal como se menciona en el subapartado sobre los motores de búsqueda, es esencial que haya una alternativa accesible al contenido y la navegación. El texto se considera la moneda universal del contenido, pero al respecto hay que hacer una advertencia, tal como veréis más adelante. Un lector de pantalla puede leer el texto sin ninguna dificultad, se puede hacer más pequeño o más grande, se puede modificar el contraste muy fácilmente y se pueden hacer muchas más transformaciones. Como el texto es tan fácil de manipular, todas las formas más exóticas de contenido deberían tener siempre una alternativa basada en texto. Algunos formatos, como las versiones más modernas de Flash, ya integran el acceso en texto con el fin de poder acceder directamente a su contenido textual directamente sin necesidad de tener que ofrecer una alternativa para todo el soporte.

! Podéis ver los motores de búsqueda en el subapartado 1.2.3 de este módulo.

El único grupo con discapacidad para el que una alternativa de texto puede no tener ninguna utilidad es el grupo de personas con discapacidades cognitivas.

La dificultad que presentan las personas con discapacidades cognitivas es que normalmente necesitan un contenido diferente y no el mismo contenido en un so-

\* <http://www.clearest.co.uk/>

porte diferente. Con eso no queremos decir que no haya que hacerlo. La simplificación del lenguaje y de la terminología utilizada en vuestro sitio web re-

resentará un beneficio para todo el mundo. Algunos grupos, como la Plain Language Commission\*, recomiendan un enfoque de “lenguaje sencillo” para el material que utilizan las compañías para dar a sus clientes la información importante, como los requisitos legales y los términos y las condiciones. Estos grupos ofrecen un léxico muy sencillo\* que contiene los términos que se pueden utilizar para una comunicación efectiva utilizando el lenguaje más simple posible.

\* <http://www.clearest.co.uk/?id=46>

¿Cómo podéis incluir alternativas de texto en vuestro sitio web? El primer paso es identificar los elementos que todavía no son texto. En el HTML hay muchos elementos que todavía no son texto. Las imágenes son el ejemplo más evidente de esto.

Aquí tenéis un ejemplo de uso accesible de una imagen:

```
1 <p>An interesting piece of art is Michelangelo's "God creates Adam"
2 .</p>
```

La imagen de este ejemplo forma parte integral del contenido. El atributo `alt` contiene una breve descripción de la imagen para las personas (o para los motores de búsqueda) que quizá no pueden verla correctamente. El atributo `longdesc` permite enlazar con una página HTML que contiene una descripción completa de la imagen. Esto se utiliza normalmente sólo para describir imágenes complejas que se utilizan como parte importante del contenido. Y también tiene el inconveniente de que los navegadores no lo aceptan completamente. La mayoría de las veces sólo utilizaréis el atributo `alt`.

Al usar las imágenes para cosas diferentes que no son contenidos, como la navegación o simplemente como decoración visual, las deberíais tratar diferente de las imágenes de contenido. Las imágenes utilizadas para hacer más atractivos los botones o la navegación de la página deben tener un atributo `alt` que coincida con el texto de la imagen. El atributo `alt` funciona sencillamente como una manera simple para permitir que el ordenador lea el texto contenido en la imagen (y lo lea a un usuario de un lector de pantalla).

En el caso de las imágenes puramente decorativas, las imágenes utilizadas para el seguimiento de anuncios o cualquier otra imagen que no tenga en principio ningún interés para el usuario o con el que no tenga que interactuar, el atributo `alt` debería quedar vacío. Esto no quiere decir que se deba omitir este atributo, sino que se debe definir como `alt=""`. Esto es así a causa de una táctica que utilizan los lectores de pantalla para ayudar a sus usuarios a hacer frente a páginas muy inaccesibles. Cuando una imagen no tiene ningún atributo `alt`, especialmente cuando forma parte de un enlace, el lector de pantalla lee la URL de la imagen al

usuario. Lo hace así para que los usuarios puedan saber que es la imagen a partir de la URL, por ejemplo si la imagen tiene un nombre similar a `anadir_al_carrito.gif`. Así pues, deberíais definir `alt=""` para las imágenes que sepáis que no tendrán ningún interés para el usuario, de manera que los lectores de pantalla no lean todas y cada una de las URL de las imágenes, lo cual podría llegar a ser muy frustrante para el usuario del lector de pantalla.

No todas las formas de contenido son tan simples como una imagen. Los soportes más complejos, como Flash (los archivos Flash pueden ser sitios web enteros en sí mismos) o las películas exigen unas descripciones más complejas. Las versiones más recientes de Flash permiten ofrecer alternativas de texto para los elementos de una película Flash, igual que en el HTML.

### 1.5.3. Definir la interacción

Muchas webs actuales implican el uso de tecnologías más allá del HTML. Incluso algo tan básico como el CSS se puede utilizar de maneras que hagan que una página o una interacción sean mucho menos accesibles.

La clave para la accesibilidad de la interacción se encuentra en el hecho de empezar con las interacciones más sencillas y utilizarlas como componentes básicos para las interacciones más complejas.

#### Nota

Tened en cuenta que el objetivo de este ejemplo es haceros ver la función que tienen los diferentes elementos de las páginas web. Para garantizar que sean accesibles, deben ser semánticamente solventes en términos tanto de los elementos HTML como de las metáforas visuales que se utilicen. Si lo encontráis todo un poco confuso, podéis volver a leer el ejemplo unas cuantas veces y mirad también unos cuantos menús y otros componentes de páginas web mientras vais comprobando no sólo si se utiliza el HTML correcto, sino también si el aspecto del componente tiene sentido en relación con su función. No podéis esperar nunca que un visitante de una página web realice una búsqueda con un cuadro de texto que se llama “Introducid vuestra dirección electrónica para suscribiros a nuestro boletín informativo”, ni tampoco podéis esperar que un visitante vidente pueda localizar contenido de interés si todos los títulos tienen el mismo estilo que el texto normal (de manera similar, tampoco podéis esperar que un usuario ciego localice contenido de interés si todos los “títulos” son en realidad sólo otros párrafos con un tamaño de texto más grande aplicado con el CSS o elementos `font`).

Un buen ejemplo de ello es la metáfora visual tan utilizada de las pestañas. La metáfora de las pestañas se ha extraído de las carpetas de anillas con clasificadores de temas. Eso se ha trasladado a los ordenadores para permitir que una única área de la pantalla pueda mostrar información de varios temas representados por pestañas conectadas al área. Hasta aquí todo es razonablemente sencillo. El problema son las tecnologías utilizadas para crear las pestañas, que normalmente se aplican con JavaScript.

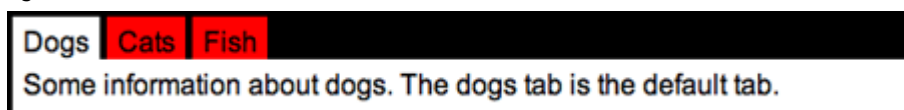
En cuanto las pestañas se utilizan como parte de una interacción más compleja que el simple hecho de permitir que el usuario seleccione la información, la metáfora original queda rota, pero a menudo se sigue utilizando el mismo código para representar las pestañas. En el siguiente ejemplo, el HTML muestra el aspecto de un control de pestaña que muestra información:

```
1 <div class="tabcontrol">
2   <div class="hd">
3     <ul>
4       <li><a href="#dogs" class="selected">Dogs</a></li>
5       <li><a href="#cats">Cats</a></li>
6       <li><a href="#fish">Fish</a></li>
7     </ul>
8   </div>
9   <div class="bd">
10    <p id="dogs" class="selected">Some information about dogs. The dogs tab is
      the default tab.</p>
11    <p id="cats">Some information about cats.</p>
12    <p id="fish">Some information about fish.</p>
13  </div>
14 </div>
```

En este ejemplo, la clase `selected` se utilizaría para especificar qué pestaña debe mostrar el gráfico de “pestaña frontal”; por ejemplo, podéis consultar la pestaña “Artículos” de la parte superior de esta página que utiliza este método.

Esta estructura es adecuada para el contenido que transmite información. En este ejemplo, la `class` de `selected` se utilizaría para indicar qué pestaña es la activa, es decir, la que está abierta y muestra su información; las demás estarían cerradas (es decir, sus párrafos estarían ocultos) hasta el momento de hacer clic en sus enlaces correspondientes. La pestaña “Dogs” es la pestaña activa por defecto, tal como muestra la figura 1.

Figura 1



Un control para pestañas muy sencillo que muestra la pestaña “Dogs” como la pestaña activa por defecto.

Cuando se hace clic en otro enlace (tal como muestra la figura 2), se utilizará el JavaScript para mover dinámicamente `class="selected"` hasta ese enlace; en este momento se aplicará el estilo a esta pestaña para mostrarla, y la que se mostraba previamente se ocultará.

Figura 2



Ahora se ha hecho clic en un enlace diferente y su pestaña correspondiente pasa a ser la pestaña activa.

También es muy habitual utilizar pestañas para permitir que los usuarios seleccionen tipos diferentes de búsquedas. En este caso, si intentáis reutilizar el estilo de código del ejemplo previo, el concepto se empieza a descomponer:

```

1  <div class="tabcontrol">
2    <div class="hd">
3      <ul>
4        <li><a href="#dogs" class="selected">Dogs</a></li>
5        <li><a href="#cats">Cats</a></li>
6        <li><a href="#fish">Fish</a></li>
7      </ul>
8    </div>
9    <div class="bd">
10     <form id="dogs" class="selected" action="search.html"
11       method="GET"><div><label for="dogsearch"><input type="text"
12         name="dogsearch" id="dogsearch"><input type="submit" value="Search for
13         Dogs"></div></form>
14     <form id="cats" action="search.html" method="GET"><div>
15       <label for="catsearch"><input type="text" name="catsearch"
16         id="catsearch"><input type="submit" value="Search for cats"></div></form>
17     <form id="fish" action="search.html" method="GET"><div>
18       <label for="fishsearch"><input type="text" name="fishsearch"
19         id="fishsearch"><input type="submit" value="Search for fish"></div></form>
20     </div>
21   </div>

```

El uso de la misma estructura de código ya no tiene ningún sentido; en este caso, están los mismos elementos `form` repetidos una y otra vez para que todo encaje con el concepto de sustituir el contenido, lo que es un despilfarro de etiquetado. En lugar de pensar visualmente, lo importante es pensar en la interacción en sí. En este ejemplo, más que seleccionar información nueva para



ver las pestañas, deberíais cambiar la interacción del usuario con el formulario de búsqueda. De hecho, lo único que debe hacer una pestaña es seleccionar el tipo de animal que busca el usuario. Si lo ponéis en práctica, podréis crear una interacción mucho mejor para todos los usuarios del sitio con un etiquetado más limpio y más sencillo de mantener:

```
1 <form action="search.html" method="GET">
2   <fieldset>
3     <legend>Search within:</legend>
4     <ul>
5       <li><label for="dogs">Dogs</label><input id="dogs" type="radio"
6         name="animal" value="dog" checked></li>
7       <li><label for="cats">Cats</label><input id="cats" type="radio"
8         name="animal" value="cat"></li>
9       <li><label for="fish">Fish</label><input id="fish" type="radio"
10        name="animal" value="fish"></li>
11     </ul>
12   </fieldset>
13   <input type="text" id="searchfield" name="search">
14   <input type="submit" value="Search">
15 </form>
```

Si se crea primero la interacción, el etiquetado es más limpio y todos los usuarios del sitio obtendrán la mejor experiencia posible. Cuando hemos empezado ampliando una metáfora visual hemos roto rápidamente la interacción y hemos creado un etiquetado horrible basado en los supuestos del ejemplo previo. Si hubiéramos utilizado AJAX para insertar el contenido en lugar de tenerlo todo en la página, el resultado habría sido aún peor. Los usuarios sin JavaScript habrían tenido que cargar una página totalmente nueva para pasar al formulario de búsqueda para gatos o peces. Si pensamos primero en la interacción básica (más que en el aspecto visual), el problema será mucho más sencillo. Ahora podemos seguir manteniendo la metáfora de las pestañas (aunque con unos cuantos estilos y *scripts*) y utilizar un único formulario para todas las búsquedas.

Esto es básico para entender cómo se debe crear una interacción accesible. Uno de los puntos positivos de HTML es que el trabajo más pesado de conseguir que las interacciones en HTML sean accesibles ya está hecho. Si no utilizáis tecnologías con HTML para romper la metáfora, podréis conseguir que la mayoría de las cosas funcionen sin muchos esfuerzos para la mayor parte de las personas.

## 1.6. Estándares de accesibilidad

En este subapartado revisaremos algunos de los estándares y directrices disponibles que pretenden definir la accesibilidad de la web y ayudar a los desarrolladores web a crear sitios accesibles. La mayoría de estos sistemas incluyen algún tipo de sistema de lista de control para que los desarrolladores puedan comprobar si sus sitios cumplen los diferentes criterios de accesibilidad.

### 1.6.1. Directrices de accesibilidad del contenido web 1.0

El W3C es uno de los principales organismos de estándares de Internet. Su Web Accessibility Initiative (WAI) publicó la primera versión de sus directrices para unos sitios web accesibles en el mes de mayo de 1999. Las directrices de accesibilidad del contenido web (WCAG) son el estándar más utilizado para la accesibilidad de la web. Varios organismos gubernamentales, que incluyen a la UE y al gobierno italiano, recomiendan o exigen el uso de las WCAG 1.0.

Las WCAG 1.0 son un grupo de 14 directrices que intentan resumir los objetivos necesarios para hacer que una página sea accesible.

Cada una de las directrices incluye varios puntos de control, que son la esencia real del documento. Las directrices explican los conceptos que los autores querían transmitir, pero la validación se hace con los puntos de control. Cada uno de los puntos de control está clasificado con una prioridad del 1 al 3 para identificar su importancia. Con el fin de ajustarse a las WCAG 1.0 hay que cumplir todos los puntos de control de prioridad 1. El cumplimiento de todos los puntos de control de prioridad 1 da una clasificación de conformidad “A”. Si, además, se cumplen todos los puntos de control de prioridad 2, entonces el nivel de conformidad es “AA”. Si se cumplen todos los puntos de control de prioridad 1, 2 y 3, entonces el nivel de conformidad será “AAA”, que es la clasificación más elevada.

Sin embargo, en realidad, las WCAG 1.0 ya se han quedado un poco anticuadas. Muchas compañías empiezan con una clasificación de conformidad “A” o “AA” y entonces se dedican a añadir otras directrices, como por ejemplo la See it Right de RNIB. Las WCAG 1.0 son un buen punto de partida, pero también deberíais tener en cuenta otros estándares más actuales, especialmente si utilizáis mucho JavaScript u otras tecnologías surgidas después del año 1999, que fue cuando se publicaron las WCAG 1.0.

Otro dato importante que hay que tener en cuenta sobre el estándar WCAG 1.0 es que fue diseñado como parte de un conjunto de 3 documentos. Otro hacía referencia a los “agentes de usuario”, que describe a los navegadores (como Opera) y otras tecnologías adicionales que la gente puede necesitar para utilizar la web

(como lectores de pantalla). El tercero cubría las herramientas de autoría, como Dreamweaver o los sistemas de gestión de contenido; su objetivo es conseguir que estas herramientas hagan una buena parte del trabajo necesario para acabar teniendo unas páginas web accesibles. Desafortunadamente, esta visión no ha cuajado y el único estándar de estos tres que se ha adoptado de una manera generalizada es el WCAG 1.0. Esto significa que a menudo no se cumplen las expectativas del WCAG 1.0 con respecto a los agentes de usuario, y que las herramientas de autoría hacen una parte muy pequeña del trabajo para conseguir unos sitios web accesibles. Esto no quiere decir que no podáis utilizar el WCAG 1.0; sencillamente significa que sólo hace referencia a una parte de los criterios de accesibilidad y que no es la solución completa.

### 1.6.2. Directrices de accesibilidad del contenido web 2.0

Una vez publicadas las directrices WCAG 1.0, el W3C ya empezó a trabajar con las WCAG 2.0.

#### Nota

En el momento de escribir este apartado, esta versión actualizada del estándar se encuentra todavía en la fase de borrador. Teniendo en cuenta el proceso que sigue el W3C, es muy probable que este estándar se publique a principios del 2009.

Las WCAG 2.0 son un poco diferentes, ya que intenta no centrarse tanto en una tecnología concreta como las WCAG 1.0; es decir, que se puede aplicar a HTML, CSS, Flash, etc. las WCAG 2.0 se basan en 4 principios de accesibilidad. Éstos son:

#### Perceptibilidad

Las personas deben poder acceder al contenido por medio de un soporte que tengan a su disposición. Por ejemplo, las personas con problemas deben poder escuchar el contenido.

#### Operatividad

Las personas deben poder interactuar con la aplicación o el contenido de la web.

#### Comprensibilidad

Las personas que utilicen el contenido y la interfaz de usuario los deben entender.

#### Robustez

Cualquier solución ofrecida debería estar disponible en diferentes plataformas o sistemas. Con esto se pretende evitar que se inventen soluciones que

la mayoría de las personas no podrán utilizar porque el hardware/software está restringido o es muy caro.

Aquí es importante decir que no se espera que los sitios web cumplan todos estos requisitos. La tecnología de que dispone el usuario también debería hacer parte del trabajo. Por ejemplo, se supone que las personas que lo necesiten ya tendrán un lector de pantalla, y por lo tanto no es necesario que todos los sitios web ofrezcan una versión de audio del contenido. Pero sí que se supone que el sitio web debe ofrecer unas páginas que se puedan leer utilizando la tecnología de lectura de pantalla más habitual con el fin de hacerlo posible. Esta diferencia es importante, igual que lo es la diferencia entre un sitio web con “*widgets* de accesibilidad” (como por ejemplo un botón para aumentar el tamaño de la letra) y una página web que funcionará en muchas situaciones diferentes (por ejemplo, varios navegadores y dispositivos que serían imposibles de prever).

Las WCAG 2.0 también son diferentes de las WCAG 1.0 por su enfoque hacia la tecnología. Como el estándar es más agnóstico con respecto a la tecnología y hace referencia a conceptos sobre accesibilidad en lugar de concretar detalles técnicos, es importante prestar atención a los documentos que acompañan el estándar. El documento “estándar”<sup>\*</sup> de WCAG 2.0 ofrecerá “la información básica”, pero el documento “técnicas”<sup>\*\*</sup> ofrece datos sólidos y aplicables al desarrollador. Éste se divide en técnicas “generales” (ambiguas con respecto a la tecnología) y en detalles concretos para las tecnologías individuales del W3C. El W3C no escribe documentos para tecnologías propietarias, por lo que deberéis encontrar técnicas para tecnologías como Flash y Silverlight en otros sitios.

<sup>\*</sup> <http://www.w3.org/TR/WCAG20/>

<sup>\*\*</sup> <http://www.w3.org/TR/WCAG20-TECHS/>

### 1.6.3. Otros estándares

Otro estándar importante desarrollado por el W3C es el estándar WAI-ARIA<sup>\*</sup>. Estas siglas son el acrónimo de Web Accessibility Initiative-Accessible Rich Internet Applications. Se trata de una serie de documentos que definen la manera de hacer accesibles las aplicaciones web complejas que utilizan tecnologías como HTML, JavaScript y AJAX.

<sup>\*</sup> <http://www.w3.org/WAI/intro/aria>

Hay muchos otros estándares para la accesibilidad de la web; demasiado numerosos para hablar de ellos con detalle. El W3C mantiene una lista excelente de políticas internacionales relativas a la accesibilidad de la web<sup>\*</sup>; éste es un recurso muy importante para poder encontrar los documentos de políticas de todos los gobiernos locales.

<sup>\*</sup> <http://www.w3.org/WAI/Policy/>

## Resumen

La accesibilidad es una cuestión importante por razones tanto económicas como sociales. No es una característica de un sitio web, sino una medida de la calidad con la que se creó. Si tenéis en cuenta el público de vuestro sitio mientras lo vais creando (y antes de hacerlo), construiréis unas páginas más accesibles con todas las ventajas que eso supone. Hay varias directrices muy conocidas que os pueden ayudar y si os ajustáis a ellas, podréis garantizar que lo que hayáis construido satisfará a los criterios de los expertos para hacer que vuestras páginas sean accesibles.

## Preguntas de repaso

1. Dad 3 razones por las que es importante crear sitios web accesibles.
2. Utilizad Internet para ver cuáles son las leyes sobre accesibilidad de vuestro país y elaborad una lista de todas las leyes que creéis que serían aplicables a vuestros sitios web. Vuestra lista también debería incluir si estas leyes os exigen que utilicéis algunos estándares de la web, como WCAG o la Sección 508.
3. Explicad de qué manera la accesibilidad es importante para la optimización de los motores de búsqueda.
4. Cread un ejemplo de un uso accesible de contenido alternativo utilizando parte de vuestro contenido propio, como por ejemplo una fotografía.
5. Utilizad Internet para ver cómo podríais conseguir que una tecnología como Flash o Silverlight sea accesible y escribid una comparación entre hacerlas accesibles y cómo se hace accesible el HTML.
6. Explicad cómo diseñaríais una interacción de una página web para que sea accesible. Escribid unas instrucciones paso a paso para crear un control de árbol (no es necesario que lo acabéis creando realmente).

## 2. Pruebas de accesibilidad

Benjamin Hawkes Lewis

Las pruebas de accesibilidad web son un subgrupo de las pruebas de usabilidad en las que los usuarios que se tienen en cuenta tienen discapacidades que afectan a su manera de utilizarlo. El objetivo final, tanto con respecto a la usabilidad como a la accesibilidad, es descubrir la facilidad con la que se puede utilizar un sitio web y utilizar esta información para mejorar futuros diseños e implementaciones.

La evaluación de la accesibilidad está más formalizada, en general, que las pruebas de usabilidad. Las leyes y la opinión pública desapruueban totalmente la discriminación de las personas con discapacidades. Para ser justos con todo el mundo, los gobiernos y otras organizaciones intentan cumplir varios estándares de accesibilidad de la web, como por ejemplo la legislación de la Sección 508 del Gobierno Federal de Estados Unidos y las directrices de accesibilidad del contenido web (WCAG) del W3C.

No obstante, es importante distinguir entre cumplir un estándar y maximizar la accesibilidad de un sitio web. En una situación ideal, ambas cosas serían la misma, pero cualquier estándar puede fallar a la hora de cumplir con los objetivos siguientes:

- Satisfacer las necesidades de todas las personas con discapacidades.
- Encontrar el equilibrio entre las necesidades de las personas con diferentes discapacidades.
- Hacer corresponder estas necesidades con las técnicas óptimas.
- Utilizar un lenguaje claro para expresar las necesidades o las técnicas.

Estos puntos débiles pueden hacer que las personas con las mejores intenciones vayan por el mal camino y pueden ser aprovechados por los que pretenden la aprobación oficial de productos inaccesibles.

Además, la accesibilidad web es un objetivo y no una cuestión de sí o no. Es un nexo entre las necesidades humanas y la tecnología. A medida que nuestro conocimiento de las necesidades humanas vaya evolucionando y a medida que la tecnología se vaya adaptando a estas necesidades, los requisitos de accesibilidad cambiarán y los estándares actuales quedarán desfasados. Los diferentes sitios web, y las distintas webs, satisfarán necesidades diferentes con tecnologías diferentes. Los servicios de chat por voz, como Skype, son ideales para las personas ciegas, mientras que los servicios de chat por vídeo son una gran ayuda para los usuarios del lenguaje de signos.

! Podés ver la legislación de la Sección 508 y las directrices de accesibilidad del contenido web en el apartado anterior de este módulo didáctico.

A la hora de determinar la facilidad de uso de un producto, las discapacidades plantean unos retos especiales, ya que pueden introducir unos huecos adicionales entre usuarios y evaluadores. La evaluación de la accesibilidad debe tener en cuenta cómo se puede experimentar la web con los diferentes sentidos y con diferentes capacidades cognitivas, y también las distintas opciones de configuración inusuales y el software especializado que permitirá el acceso a la web a las personas con discapacidades concretas.

Si intentáis evaluar la usabilidad o la accesibilidad de vuestro sitio web, es muy difícil que os podáis poner en el lugar de un adolescente amante del cine o de un director de banco de 59 años que utilizan vuestro sitio, incluso antes de tener en cuenta las discapacidades. Pero ¿qué sucede si el adolescente amante del cine es sordo y necesita que las películas estén subtituladas? ¿Y si el director de banco es ciego y utiliza una tecnología especial (como un lector de pantalla) para interactuar con su entorno de escritorio y el navegador web con la que el evaluador no está nada familiarizado?

Las directrices y las herramientas de accesibilidad ayudan a salvar estas faltas de experiencia. A pesar de ello, no dejan de ser un complemento, y no una sustitución, para la imaginación empática, la ingenuidad técnica y las conversaciones con los usuarios.

En este apartado hablaremos de varios enfoques para evaluar la accesibilidad de la web, tanto desde la perspectiva del establecimiento de la conformidad formal como desde la perspectiva de la maximización de la accesibilidad.

### **2.1. ¿Cuándo se deben realizar las pruebas?**

Uno de los dichos de siempre del mundo de la ingeniería de software es “probad pronto y probad a menudo”. El hecho de realizar las pruebas al final de todo el proceso de desarrollo implica dos riesgos:

- 1) Los proyectos suelen durar más de lo previsto y tener un precio más alto del presupuestado. A causa de estas presiones, las pruebas se hacen normalmente con prisas, no se hacen o se ignoran.
- 2) Normalmente, supone más trabajo solucionar los problemas que se descubren una vez que el proceso ya está muy adelantado que hacerlo ya desde el principio.

Así pues, para garantizar la calidad y ahorrar tiempo y dinero, las evaluaciones de accesibilidad deberían empezar ya desde el primer momento del diseño del producto y se deberían incluir en todas las versiones de desarrollo siguientes hasta la entrega del producto acabado.

## 2.2. Entender los requisitos

Antes de empezar a evaluar la accesibilidad de un proyecto, debéis determinar cuáles son los requisitos clave teniendo en cuenta su entorno, el público al que va destinado y los recursos. Algunos requisitos vendrán impuestos por terceros como gobiernos y clientes; otros los podréis elegir por vuestra cuenta.

### 2.2.1. Requisitos externos

Normalmente los requisitos vienen impuestos desde fuera, como por ejemplo:

- **Gobiernos.** Los requisitos adoptan normalmente la forma de una legislación general contra la discriminación de las personas con discapacidades, y muy pocas veces exigen el uso de un estándar concreto o enumeran unos requisitos precisos de conformidad. Una excepción importante es cuando la legislación obliga al sector público a utilizar un estándar concreto. Por ejemplo, la Sección 508\* forma parte de la legislación de Estados Unidos y obliga a los sitios web producidos por agencias federales a satisfacer como mínimo un grupo concreto de requisitos definidos. La página de las políticas de la WAI relativas a la accesibilidad de la web\*\* ofrece una lista parcial de legislaciones similares. Pero si queréis saber exactamente qué obligaciones tenéis en vuestra jurisdicción, consultad a un abogado.
- **Políticas del cliente.** Por ejemplo, Shell está intentando garantizar que sus sitios web cumplan el nivel de conformidad “Doble A” del WCAG 1.0\*, por lo que, si estáis desarrollando un sitio web para Shell, deberéis satisfacer (como mínimo) este estándar.
- **Utilidad como marketing.** La conformidad con un estándar concreto, como por ejemplo la Sección 508, puede ayudar a vender un proyecto a clientes que se preocupan por la accesibilidad.
- **Política de accesibilidad interna en vuestra organización.** Por ejemplo, los productos producidos para la BBC deben satisfacer las directrices de accesibilidad de la BBC v1.3\*.

\* <http://www.section508.gov/>  
\*\* <http://www.w3.org/WAI/Policy/>

\* <http://www.shell.com/home/content/footer/accessibility/statement/>

\* <http://www.bbc.co.uk/guidelines/futuremedia/accessibility/>

### 2.2.2. Detalles de conformidad

Es muy importante tener tan claro como sea posible cuáles son los requisitos externos. Algunos estándares de accesibilidad incorporan más de un nivel o tipo de conformidad, por lo que hay que concretar muy bien cuál es el que se exige. Por ejemplo, el WCAG 1.0 tiene tres niveles de conformidad:

- 1) Las personas con alguna discapacidad “no podrán acceder de ninguna manera a la información” de un documento que no supere el “nivel A”.



2) Las personas con alguna discapacidad “lo tendrán difícil para acceder a la información” de un documento que no supere el nivel “Doble A”.

3) Las personas con alguna discapacidad “encontrarán algunas dificultades para acceder a la información” de un documento que no supere el nivel “Triple A”.

El borrador de WCAG 2.0 también incorpora tres niveles, pero las posibilidades de conformidad son más complicadas. Cuando un recurso forma parte de una serie de recursos que forman todo un proceso (por ejemplo la búsqueda, selección, compra y pago de un producto en una tienda en línea), el nivel de conformidad para todos los recursos de la serie es el del recurso con el nivel más bajo.

Las declaraciones de conformidad se deben basar en la tecnología de contenido “compatible con la accesibilidad”. Para poder ser una tecnología de contenido compatible con la accesibilidad, una tecnología debe hacer lo siguiente:

- Haber demostrado que funciona con la tecnología de asistencia de los usuarios.
- Tener unos agentes de usuario (navegadores, conectores, etc.) que funcionen con la tecnología de asistencia de los usuarios y estar disponible para los usuarios con discapacidades sin ningún coste superior al que tiene un usuario sin ninguna discapacidad.

Tened en cuenta que en un entorno de intranet quizá sí podréis garantizar que estos agentes de usuario estarán disponibles para los usuarios, pero en la web será imposible. Por ejemplo, es posible que una aplicación se pueda utilizar sin ninguna tecnología comercial, pero que sea accesible sólo con lectores de pantalla con un conector comercial para el que la organización ha comprado una licencia. Esta aplicación podría ser conforme a las WCAG 2.0 cuando se utiliza en la intranet de la organización, pero no cuando se utiliza en la web pública.

El WCAG 2.0 también permite unas declaraciones de conformidad más limitadas. Un recurso no accesible puede ser conforme si se ofrece una alternativa accesible. Los editores pueden hacer una declaración de conformidad parcial cuando se añade contenido procedente de otras fuentes.

### **2.2.3. Superar las expectativas**

La determinación de los requisitos externos debería ser sólo el principio de todo el proceso; estos requisitos se deberían considerar como los requisitos mínimos a los que hay que añadir unos objetivos más altos con el fin de maximizar la accesibilidad. Como evaluadores de accesibilidad, vuestra tarea es estimular la concienciación sobre la accesibilidad, ya que sois los expertos en el tema.

Cuando entreguéis un informe final, es posible que debáis distinguir los dos aspectos. Por ejemplo, las instrucciones de un cliente para un supermercado

en línea pueden especificar que la tienda sea accesible para los usuarios ciegos. Teniendo en cuenta al público al que va dirigida, también deberíais evaluar si la tienda será accesible para usuarios con otras discapacidades.

Tened en cuenta que los requisitos externos de conformidad con un estándar concreto no impiden necesariamente la aplicación de directivas de buenas prácticas de otros estándares. Por ejemplo, podéis evaluar un sitio web de una agencia federal pensado para su uso por parte de personas de la tercera edad y que debe cumplir con la Sección 508.

La Sección 508 estipula lo siguiente:

“§ 1194.22 (c) Las páginas web deben diseñarse de manera que toda la información que se transmita mediante colores esté disponible también sin colores, por ejemplo a partir del contexto o del etiquetado”.

Esta disposición ayuda a los usuarios, que saben cómo adaptar la presentación del contenido de la web, pero no maximiza la accesibilidad de la presentación por defecto del contenido en la audiencia destino, ya que no garantiza que habrá el contraste suficiente entre los colores sugeridos. Por suerte, no hay nada que impida que un sitio web satisfaga este requisito y, además, satisfaga las disposiciones del siguiente nivel del borrador de las directrices de accesibilidad del contenido web 2.0.

#### **Borrador de las directrices de accesibilidad del contenido web 2.0.**

“1.4.3 Contraste (mínimo): El texto y las imágenes de texto tienen una relación de contraste de como mínimo 5:1, excepto en los casos siguientes (nivel AA):

- Texto grande: El texto y las imágenes de texto de tamaño grande tienen una relación de contraste de 3:1 como mínimo.
- Secundario: El texto o las imágenes de texto que forman parte de un componente inactivo de la interfaz de usuario, que son puramente decorativos, que son texto secundario en una imagen o que no son visibles para nadie, no tienen ningún requisito de contraste mínimo.

1.4.6 Contraste (mejorado): El texto y las imágenes de texto tienen una relación de contraste de como mínimo 7:1, excepto en los casos siguientes: (Nivel AAA)

- Texto grande: El texto y las imágenes de texto de tamaño grande tienen una relación de contraste de 5:1 como mínimo.
- Secundario: El texto o las imágenes de texto que forman parte de un componente inactivo de la interfaz de usuario, que son puramente decorativos, que son texto secundario dentro de una imagen o que no son visibles para nadie, no tienen ningún requisito de contraste mínimo.

Los criterios 1.4.3 y 1.4.6 se pueden satisfacer mediante un control de contraste disponible en la página o desde ésta”.

#### **Nota**

Cuando habla del control de contraste, este criterio se refiere al hecho de que deberíais ofrecer una manera de cambiar los colores o una variación de alto contraste. Para comprobar los esquemas de control de contraste, podéis utilizar el analizador de contraste de color de Juicystudio\*.

\* <http://juicystudio.com/services/luminositycontrastio.php>

Las WCAG 2.0 se están diseñando para que tengan una gran compatibilidad con versiones anteriores de otros estándares, especialmente WCAG 1.0 y la Sección 508.

#### 2.2.4. La importancia de la interfaz de usuario

Pensad en la importancia que tiene el hecho de conseguir que la interfaz de usuario de un sitio web sea accesible. Aunque el contenido no esté disponible de una manera adecuada, una interfaz de usuario accesible puede ayudar a los usuarios a identificar el contenido de interés y solicitar ayuda externa para convertirlo a un formato en el que lo pueda utilizar. Por ejemplo, una persona con una discapacidad auditiva se puede encontrar con un vídeo de una charla o en un sitio donde se comparten vídeos sin subtítulos. Como la URL identifica el vídeo de una manera exclusiva, y como la persona puede seguir utilizando el reproductor para ver el vídeo, lo podrá enviar a un tercero, como por ejemplo el servicio gratuito Amara\*, para añadir subtítulos.

\* <http://amara.org>

#### 2.2.5. Personajes con discapacidades

Un enfoque ideal es integrar las discapacidades clave para vuestro proyecto directamente en los personajes que actuarán como usuarios\*: unos usuarios ficticios que harán de arquetipos para ver el modo como algunos tipos de usuarios particulares utilizarán un sitio web. Imaginémonos que estáis evaluando prototipos para un sitio donde se compartirán vídeos y que vuestros personajes incluyen:

\* <https://www.usability.gov/how-to-and-tools/methods/personas.html>

- James Smith, de 23 años, que está loco por el fútbol y que lo que desea por encima de cualquier otra cosa es compartir imágenes interesantes de deportes con sus amigos.
- Sarah Maddison, de 34 años, una madre trabajadora que normalmente no tendría tiempo para dedicarse a visitar un sitio como éste. Pero resulta que su hija de tres años es muy aficionada a ver vídeos, y por eso Sarah quiere sentarse con ella y ayudarla a encontrar los vídeos adecuados que quiere ver.

Podéis seleccionar a estos personajes e incorporarles discapacidades, como por ejemplo:

- discapacidad visual
- daltonismo
- ceguera
- sordera
- discapacidad auditiva
- sordoceguera
- epilepsia
- dislexia

Por ejemplo, podéis decidir que James también es sordo y que quiere subtítulos de los comentarios que se escuchan en los vídeos, y que Sarah ve mal y tiene problemas para ver los tipos de letra de fantasía y el texto de tamaño muy pequeño. Estos personajes os sirven de guía para rechazar prototipos que no incluyan la utilidad de subtítulos en el reproductor de vídeo o que utilicen unos tipos de letra muy elaborados que exigirían imágenes.

*How People with Disabilities Use the Web*\* de la WAI y *Just Ask: Integrating Accessibility Throughout Design*\*\* de Shawn Lawton Henry contienen otros ejemplos de personajes afectados por discapacidades que podéis utilizar para empezar a trabajar en este aspecto.

\* <http://www.w3.org/WAI/EO/Drafts/PWD-Use-Web/>  
\*\* <http://www.uiaccess.com/accessucd/personas.html>

No debería ser necesario decirlo, pero no penséis que las personas con discapacidades son todas iguales. La discapacidad es un fenómeno increíblemente variado y, además, las personas con discapacidades son tan variadas como las personas sin discapacidades; sus diferencias (por ejemplo) pueden ser de género, edad, intereses, valores y habilidades (y quizá las más importantes son las relacionadas con su experiencia informática).

De nuevo, la comparación de productos en cuanto a las directrices de accesibilidad puede ayudar a llenar los huecos que vuestros personajes no cubren. Por ejemplo, quizá seguís las WCAG 2.0 con el sitio para compartir vídeos pero vuestros personajes no incluyen a ningún usuario con epilepsia. Aun así, podéis leer la Directriz 2.3 (“Ataques: no se debe diseñar el contenido de ninguna manera que pueda provocar ataques”) y decidir que el sistema debe poder filtrar los vídeos cargados antes de mostrarlos por si contienen centelleos.

### 2.2.6. Elegir un estándar de accesibilidad

Si debéis elegir un estándar de accesibilidad para gestionar las cuestiones de accesibilidad de la web en un equipo o sencillamente para utilizarlo como guía durante las pruebas, aconsejamos las WCAG 2.0 por estos motivos:

- Están diseñadas a partir de las necesidades humanas básicas, aplicables a tecnologías diferentes de HTML y CSS (como Flash).
- Documentan de manera muy esmerada las razones de cada uno de los criterios de conformidad.
- Sugieren técnicas prácticas para satisfacer los criterios de conformidad utilizando tecnologías actuales.
- Garantizan que todas las disposiciones se puedan someter a pruebas.
- Incorporan estudios más recientes que las alternativas actuales.

- Están diseñadas para ser claramente compatibles con los estándares de accesibilidad existentes.
- Serán un estándar internacional.

Podéis citar la conformidad con un borrador concreto de WCAG 2.0; por cuestiones de marketing, lo mejor es buscar también la conformidad con estándares acabados como la Sección 508 y WCAG 1.0, además de este borrador.

### 2.2.7. El espíritu de la ley

A la hora de hacer pruebas en comparación con unas directrices, es importante tener en cuenta la base fundamental de cualquier orientación técnica específica, que es cumplir el espíritu de la ley y no sólo su letra.

A continuación se explica una historia muy aleccionadora. La Sección 508 (§ 1194.22) incluye un requisito que dice: “Se ofrecerá un equivalente de texto para todos los elementos que no sean de texto (por ejemplo, con `alt`, `longdesc` o en el contenido del elemento)”.

De manera similar, el WCAG 1.0 incluye un punto de control que dice: “Ofreced un equivalente de texto para todos los elementos que no sean de texto (por ejemplo, con `alt`, `longdesc` o en el contenido del elemento). Esto incluye: imágenes, representaciones gráficas de texto (incluyendo símbolos), regiones de mapa de imagen, animaciones (GIF animados, por ejemplo), miniaplicaciones y objetos programáticos, arte ascii, marcos, *scripts*, imágenes utilizadas como picos para listas, separadores, botones gráficos, sonidos (reproducidos con interacción del usuario o sin), archivos de audio autónomos, pistas de audio de vídeo y vídeo”.

Desafortunadamente, mucha gente que lee esta directriz interpreta mal lo que debe ser un equivalente de texto genuino para un separador y los elementos decorativos y generan un etiquetado como el siguiente:

```

```

De hecho, como estas imágenes no transmiten ninguna información nueva y no tienen ninguna función, su equivalente de texto correcto sería una cadena vacía (`alt=""`), que hace que el lector de pantalla pase por alto el atributo `alt` y no lo lea. Para un usuario de un lector de pantalla es muy molesto tener que oír en voz alta un texto como “borde decorativo” una y otra vez cuando en realidad no les ofrece ninguna información útil.

Las WCAG 2.0 intentan ser más claras. La directriz equivalente\* dice: “Todo el contenido que no sea texto tiene una alternativa de texto que presenta una información equivalente, excepto para las situaciones enumeradas más abajo”. Una de estas situaciones es: “Decoración, formato, invisible: si es puramente decoración, o si se utiliza sólo para el formato visual, o si no se presenta a los usuarios, entonces se implementa de manera que la tecnología de asistencia la pueda ignorar”. Y también, muy importante, WCAG 2.0 intenta detallar el razonamiento que hay tras la directriz\*\*.

\* <http://www.w3.org/TR/WCAG20/#text-equiv>  
\*\* <http://www.w3.org/TR/2007/WD-UNDERSTANDING-WCAG20-20071211/text-equiv.html#text-equiv>

“El objetivo de esta directriz es garantizar que todo el contenido que no sea texto también esté disponible como texto. «Texto» se refiere al texto electrónico y no a una imagen de texto. El texto electrónico tiene la ventaja especial de que es neutro con respecto a la presentación. Es decir, que se puede reproducir visualmente, por audio, por tacto o en cualquier combinación. Así pues, la información enviada como texto electrónico se puede presentar en la forma que mejor satisfaga las necesidades del usuario. También se puede ampliar muy fácilmente, se puede enviar con una voz fácil de entender o se puede reproducir en el formato táctil que mejor se adecue a las necesidades de un usuario”.

### 2.3. ¿Quién debe hacer las pruebas?

Hay básicamente dos grupos que realizan pruebas: expertos y usuarios.

Las pruebas realizadas por expertos son importantes porque éstos entienden el modo como interactúan las tecnologías web subyacentes, pueden actuar como nexo de unión para el conocimiento sobre los diferentes grupos de usuarios y tienen una inclinación a aprender a utilizar herramientas de pruebas especializadas.

Las pruebas realizadas por usuarios son cruciales porque éstos son los verdaderos expertos sobre sus capacidades y su tecnología de asistencia. Las pruebas hechas por usuarios también pueden poner de manifiesto los problemas de usabilidad entre usuarios más y menos técnicos, y entre personas familiarizadas con el sitio web en cuestión (como por ejemplo los verificadores expertos) y personas que no lo están (usuarios nuevos).

Es muy improbable que un desarrollador web que sepa cómo utilizar un lector de pantalla explore un sitio de la misma manera que un usuario normal de un lector de pantalla; y también es muy improbable que los usuarios de lectores de pantalla que programan sus *scripts* exploren el sitio utilizando las mismas estrategias que los usuarios de lectores de pantalla que se limitan a hacer las tareas informáticas normales, como por ejemplo escribir mensajes electrónicos.

La próxima vez que se realicen pruebas (ya sean otras pruebas del mismo proyecto o pruebas de un proyecto totalmente diferente), el conocimiento adquirido con las pruebas de usuarios se incorpora al proceso de pruebas avanzadas. Las pruebas de usuarios también tienen una ventaja más sutil. Por el hecho de humanizar la accesibilidad y de reunir desarrolladores y usuarios finales, estas pruebas pueden hacer aumentar la motivación para construir sitios web accesibles.

## 2.4. Pruebas avanzadas

Las pruebas avanzadas incluyen cuatro componentes:

- **Evaluación guiada por herramientas:** en las que una herramienta localiza problemas de accesibilidad y los presenta al evaluador (esto incluiría los comprobadores de accesibilidad y los analizadores de código).
- **Simulación:** en las que el experto simula una experiencia de usuario final del sitio web. A menudo no será necesario que busquéis mucho para encontrar problemas de accesibilidad. Es posible que no tengáis que hacer nada más que cargar la página en vuestro navegador y ya veréis que el texto es muy difícil de leer.
- **Inspección basada en herramientas:** cuando el evaluador utiliza una herramienta para investigar el modo como funcionan juntas las diferentes partes de un sitio web.
- **Revisión del código:** el evaluador mira directamente el código y los activos de un sitio web para localizar problemas.

Es muy probable que los principiantes dependan especialmente de la evaluación guiada por herramientas, pero los evaluadores de todos los niveles de experiencia se pueden beneficiar de cada uno de los componentes. Incluso los principiantes pueden detectar elementos `img` sin equivalentes de texto en un etiquetado HTML y, a medida que vais ganando experiencia, os será cada vez más fácil detectar problemas antes de que paséis a pruebas más rigurosas. Para los expertos en grandes proyectos, la revisión manual de todo el código del cliente o la inspección de todas las partes de un sitio web puede no ser factible; en estos casos, una evaluación guiada por herramientas puede localizar áreas con un problema concreto que merezcan una atención especial. Además, los evaluadores humanos pueden pasar por alto cosas que la evaluación de una máquina habría detectado.

Desafortunadamente, aunque hay muchas herramientas de accesibilidad, la mayoría falla en algún aspecto u otro. Por ejemplo, una herramienta que enumera los títulos de los documentos HTML comete el error de no incluir el texto `alt` de los elementos `img`. De la misma manera que debéis tener en cuenta el espíritu de la ley con respecto a la conformidad con los estándares, también lo deberíais tener en cuenta a la hora de utilizar herramientas. Antes de quejaros a alguien sobre un problema de accesibilidad, comprobad que se trata de un problema real y no de un error de la herramienta.

### 2.4.1. Comprobadores de accesibilidad semiautomatizados

Una vez que se han solucionado los problemas que se pueden detectar a simple vista, una buena manera de proseguir es pasar la página por una herramienta de comprobación de accesibilidad semiautomatizada. Si evaluáis la conformidad con un estándar concreto, probablemente querréis utilizar una diseñada para su uso con este estándar. La Iniciativa de Accesibilidad Web del W3C mantiene una buena lista de herramientas\*.

\* <https://www.w3.org/WAI/ER/tools/>

Estas herramientas tienen unas limitaciones importantes. No existe nada que se pueda considerar una prueba de accesibilidad totalmente automatizada. Por ejemplo, dada la naturaleza primitiva de la inteligencia artificial actual, un programa informático no puede tener la última palabra sobre si un texto es un equivalente genuino para una fotografía en un contexto. Incluso en áreas que teóricamente se pueden automatizar totalmente, los programadores de comprobadores se pueden equivocar en sus interpretaciones de las directrices de accesibilidad y perderse el espíritu de la ley entre sus letras.

Las buenas herramientas inspeccionan la página para localizar problemas de accesibilidad y generan una lista de cosas que consideran errores y otras cosas que creen merecedoras de una investigación por parte de seres humanos. Por ejemplo, si Cynthia Says encuentra un elemento `img` con `alt=""`, emitirá una advertencia (no un error) que indicará al usuario que “verifique que esta imagen se utiliza sólo como espaciador o para el diseño y que no tiene ningún significado”. Si el equivalente de texto correcto para esta imagen es una cadena vacía, deberíais pasar al siguiente error o advertencia.

Quizá la mayor ventaja de los comprobadores de accesibilidad es que si elegís uno, como el TAW 3\*, que se puede ejecutar con múltiples URL, podréis encontrar páginas en series muy grandes que probablemente necesiten una atención más precisa.

\* <http://www.tawdis.net/>

### 2.4.2. Inspectores estructurales

Muchas herramientas de inspección están diseñadas para investigar las estructuras del contenido web. Las estructuras, por decirlo de una manera sencilla, definen cuáles son los componentes de un sitio web y cómo se relacionan entre ellos. Por ejemplo, en el modelo de objetos de documento (DOM) de HTML, el texto se puede designar como una etiqueta para un campo de formulario utilizando el elemento `label`. Los navegadores convierten el HTML en un modelo de objetos de documento. El navegador asocia varios comportamientos con componentes concretos. Por ejemplo, si



hacéis clic en la etiqueta de una casilla de selección, ésta normalmente quedará seleccionada.

Los entornos de escritorio y las aplicaciones aceptan la interactividad con lectores de pantalla, software de reconocimiento del habla y otras tecnologías de asistencia ofreciendo una estructura similar que representa el contenido y las funciones disponibles en la presentación visual. En Windows, el sistema estructural principal se conoce como Microsoft Active Accessibility (MSAA), o Vista UI Automation\*. Por ejemplo, un cuadro de diálogo tiene una serie de hijos relacionados, como el título, los campos, los botones y las etiquetas.

\* <http://msdn.microsoft.com/en-us/library/ms788733.aspx>

Las tecnologías de asistencia típicas se preocupan básicamente de la representación que hacen los navegadores y los conectores del contenido de la web con respecto a estos sistemas estructurales en lugar de procesar directamente los modelos de objetos del documento web.

Hay inspectores tanto para las estructuras de nivel de escritorio como para los modelos de objetos de nivel de web. Las herramientas para analizar el modelo de objetos de documento de (X)HTML incluyen los inspectores DOM de los diferentes navegadores y grupos de herramientas de accesibilidad como los que encontraréis en <https://www.w3.org/WAI/ER/tools/>.

Los inspectores DOM muestran el árbol de los elementos, atributos y texto construido a partir de la serialización de (X)HTML, mientras que los inspectores de accesibilidad web extraen componentes o relaciones concretos y los enumeran. Por ejemplo, pueden enumerar todos los campos con sus etiquetas, todos los títulos o todos los enlaces.

Normalmente, el análisis del modelo de accesibilidad no debería ser necesario para (X)HTML, aunque quizá también necesitaréis investigar esta capa si creéis que un navegador representa incorrectamente en una tecnología de asistencia una estructura (X)HTML correcta. Sin embargo, normalmente comprobaréis directamente las estructuras (X)HTML.

No todo el contenido se puede inspeccionar con inspectores DOM o de la web. La inspección de aquello que se muestra en las estructuras de accesibilidad de nivel de escritorio es importante para comprobar qué contenido del conector (reproductores de medios, contenido Flash y *applets* Java) se muestra en la tecnología de asistencia que utiliza aquellos modelos de accesibilidad.

En general, debéis comprobar que todos los controles aparezcan con la función apropiada (por ejemplo, que los cuadros de texto sean cuadros de texto y los botones sean botones) y con las propiedades necesarias.

### 2.4.3. Simulación y uso de tecnologías de asistencia de usuario final

La simulación implica emular las experiencias de las personas con discapacidades durante las pruebas. Esto se puede hacer utilizando la tecnología de asistencia para interactuar con un sitio o intentando restringir de alguna manera las capacidades propias. Por ejemplo:

- Utilizar un puntero de boca para pulsar teclas mientras se comprueba la accesibilidad del teclado.
- Ver una página con el simulador Vischeck, que intenta presentar la página, con las imágenes incluidas, tal como la ven las personas con diferentes tipos de daltonismo.
- Apagar el monitor mientras se utiliza un lector de pantalla junto con un navegador.

La simulación puede ayudar al desarrollador a ser consciente de las necesidades de las personas con discapacidades y puede hacer patente defectos de diseño fundamentales. El uso de tecnologías de asistencia puede aclarar ciertos malentendidos sobre cómo son compatibles e interactúan, o no, con los estándares web. Por ejemplo, los lectores de pantalla más populares no utilizan estilos sugeridos para los tipos de soportes CSS `aural` o `braille`, sino que intentan representar el tipo `screen` presentado por los navegadores visuales con los que interactúan.

El uso de tecnologías de asistencia no es una tarea que se pueda tomar a la ligera, ya que para entender bien cómo utilizan estos sistemas puede ser necesario un nivel muy alto de inmersión y formación. Existe un riesgo muy grande de crear nuevos malentendidos. Los desarrolladores se pueden esforzar mucho para hacer algo con un lector de pantalla y llegar a la conclusión de que no se puede hacer por culpa de un defecto del lector de pantalla, cuando en realidad es una consecuencia de su inexperiencia con la herramienta. También puede ser que intenten utilizar la herramienta de una manera equivocada; por ejemplo, quizá intentan leer una página de manera secuencial cuando un lector de pantalla real iría saltando por la página utilizando los títulos y otros elementos mientras busca puntos de interés. O incluso es posible que no consigan leer adecuadamente la pantalla. La lectura de una página que podéis ver o que conocéis bien con un lector de pantalla es muy diferente de la exploración de un sitio totalmente nuevo que no podéis ver.

El uso de tecnologías de asistencia debe ir acompañado de la experiencia sobre el uso que hacen de estas tecnologías los usuarios habituales y las conclusiones extraídas de este uso se deberían confirmar idealmente con usuarios expertos.

En general, los principiantes deberían dejar el uso de las tecnologías de asistencia a los usuarios verificadores.

#### 2.4.4. Inspección detallada

Una vez que se hayan solucionado todos los problemas reales identificados por la herramienta de comprobación elegida, ya podréis pasar a la comprobación, la verificación, el análisis y la revisión manuales del proyecto.

Las WCAG 2.0 dividen su criterio de mejores prácticas en cuatro principios. El contenido y las funciones deben ser:

- **Perceptibles** (por ejemplo, las imágenes deben tener equivalentes de texto).
- **Operativos** (por ejemplo, debería ser posible interactuar con un sitio web sin un ratón y navegar con un lector de pantalla).
- **Comprensibles** (por ejemplo, el texto no debe ser más complicado de lo necesario y el sitio web debe funcionar de una manera predecible).
- **Robustos** (por ejemplo, los sitios web deben funcionar con interoperatividad con diferentes agentes de usuario y la navegación debe ser lógica).

En este subapartado presentaremos algunos ejemplos sobre cómo los verificadores expertos pueden evaluar hasta qué punto el contenido se adecua a estos principios. Tened en cuenta que esta sección no está pensada para sustituir la revisión de las WCAG y sus técnicas.

#### Perceptibilidad

Un subgrupo de los problemas de perceptibilidad está relacionado con la existencia de soportes alternativos de varios tipos. Podéis comprobar los equivalentes de texto desactivando las imágenes y el contenido multimedia en vuestro navegador y mirando la página. Pero deberéis tener un cuidado especial con los elementos `img` e `input`. Normalmente, deberéis definir atributos `alt` en blanco (`alt=""`) para todas las imágenes puramente decorativas, de manera que el lector de pantalla las ignore. Sin embargo, en los casos siguientes:

- imágenes que son el único contenido de enlaces y
- botones de formulario

cuando estos elementos tienen unos atributos `alt=""`, los lectores de pantalla interpretarán la imagen o el botón como si el atributo `alt=""` no estuviera, e intentan poner uno (por ejemplo, leyendo la URL de la imagen).

Por lo tanto, en estas circunstancias concretas deberéis comprobar que las imágenes dentro de enlaces o botones tengan un atributo `alt` que describa el destino del enlace o la acción del botón, incluso aunque sea un poco repetitivo.

#### Nota

La comprobación de equivalentes sincronizados con multimedia, como por ejemplo títulos y descripciones de audio, se puede llevar a cabo mirando las preferencias de vuestro reproductor de soportes para activar los parámetros de accesibilidad.

Otro grupo de problemas de perceptibilidad es el que hace referencia a los estilos de la página. En este caso, hay tres áreas que hay que investigar:

- ¿La presentación propuesta para la página es razonablemente accesible? Por ejemplo, ¿el contraste del color es suficiente? ¿El texto tiene un tamaño lo suficientemente grande para poder leerlo cómodamente? Aparte de mirar la página por vuestra cuenta, también podéis utilizar una herramienta como Juicy Studio CSS Analyser\* para comprobar las combinaciones de colores de fondo y de primer término con fórmulas que pretenden medir la legibilidad.
- ¿Las sugerencias del editor para la presentación se pueden combinar bien con las preferencias de los usuarios normales que tienen como objetivo hacer que el contenido sea más legible, como un tipo de letra de un tamaño mayor, el zoom y unos colores por defecto diferentes? Intentad aumentar el tamaño del texto entre 2 y 5 veces; no os preocupéis si los resultados no son perfectos en cuanto a pixelación, sino que lo que debéis mirar es que la maquetación no quede muy afectada y haga que el contenido sea difícil de leer. Intentad cambiar las preferencias de color para ver qué sucede. Si el CSS del editor define los colores, debería definir explícitamente los colores de fondo y de primer término juntos para garantizar que la combinación de preferencias poco habituales y estilos del editor no resulte en un texto ilegible o invisible. Los navegadores más populares permiten que los usuarios definan sus preferencias de color y que desactiven las imágenes de fondo del CSS. Cuando lo probéis por vuestra cuenta, quizá descubriréis unas técnicas de colocación de imágenes de CSS mal concebidas que hacen que el texto quede oculto, ya que la imagen no se cargará pero el texto seguirá siendo invisible.
- Si rechazáis las sugerencias de presentación del editor, ¿se conserva toda la información comunicada para estas sugerencias en el contenido de la web para su uso por parte de los estilos por defecto del agente de usuario o por parte de los estilos de usuario? Intentad desactivar el CSS e inspeccionad el modelo de objetos de documento para comprobar que los títulos están

\* <http://juicystudio.com/services/csstest.php>

marcados como títulos y que las tablas se utilizan para los datos tabulados y no para la maquetación.

## Operatividad

La salud y la seguridad son una parte crucial a la hora de conseguir que un sitio web sea operativo, aunque no se tienen en cuenta casi nunca. El contenido intermitente, por ejemplo, es un riesgo que puede provocar ataques a las personas epilépticas fotosensibles. Podéis hacer una captura de pantalla de vuestro sitio web actual y pasarla por la Photosensitive Epilepsy Analysis Tool (PEAT)\* de Trace Center para ver si hay contenido intermitente que pueda representar un peligro para vuestros usuarios. Evidentemente, si creáis un sitio web para compartir vídeos, éste será un aspecto al que deberéis prestar una atención especial. Durante la fase de diseño del producto podéis pensar en la posibilidad de incluir un proceso de filtraje automatizado para las cargas.

\* <http://trace.wisc.edu/peat/>

Además, una buena manera de comprobar la operatividad de los sitios web es sencillamente mirar si podéis acceder a todo el contenido esencial y a todas las funciones con diferentes dispositivos:

- Intentad utilizar vuestro sitio web utilizando sólo el teclado. ¿El punto marcado en cada momento está siempre claramente indicado? ¿Es posible acceder a todas las funciones con el teclado?
- Intentad utilizar vuestro sitio web utilizando sólo una pantalla táctil.
- Intentad navegar por vuestra página web con instrucciones de voz utilizando Opera para Windows y su complemento Voice, o Windows Vista Speech Recognition e Internet Explorer.

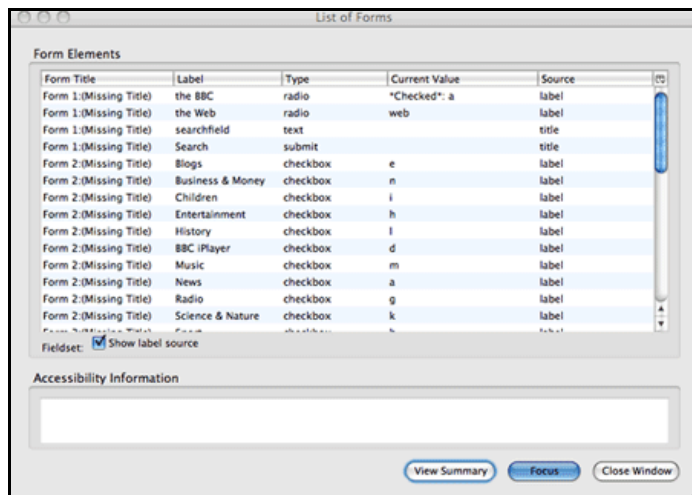
### Nota

Hace poco se ha introducido el reconocimiento del habla comercial con calidad de dictado en el Mac OS X mediante el MacSpeech Dictate, pero actualmente no existe ningún equivalente en las plataformas \*nix gratuitas.

Los lectores de pantalla y otras tecnologías de asistencia pueden utilizar la estructura semántica del (X)HTML para asociar correctamente el contenido y para permitir la navegación por el contenido. Por ejemplo, los lectores de pantalla pueden permitir a los usuarios que pasen hasta el próximo sitio donde hay un título u otro tipo de elemento, o pueden listar todas las ocurrencias de un tipo concreto. El uso correcto de los elementos `label` y `legend` permite que las tecnologías de asistencia asocien las etiquetas con los campos de formulario correctos; y el uso correcto de los elementos `th` y de los atributos `header`, `scope` y `axis` les permite asociar los títulos de tablas con las casillas de datos de las tablas. La estructura semántica se puede evaluar con un inspector

del modelo de objetos del documento (DOM), como por ejemplo el que se encuentra en Opera Dragonfly. Las herramientas de inspección de la accesibilidad, como la Firefox Accessibility Extension, pueden facilitar estas tareas, por ejemplo, enumerando todos los títulos de la página o listando los atributos de los campos de formulario (y mostrar cuáles no tienen ninguna etiqueta asociada). En la figura 1 encontraréis un ejemplo de esto.

Figura 1



Captura de pantalla de la ventana de información de formularios de Firefox Accessibility Extension con la nueva página inicial de la BBC

## Comprensibilidad

Evaluar la comprensibilidad es una tarea aún más subjetiva que comprobar la legibilidad. A no ser que un evaluador sea nuevo en un proyecto o sea un editor profesional, es muy probable que no sea la mejor persona para evaluar si un texto es tan comprensible como sea posible. Sin embargo, podéis utilizar la herramienta Readability Test de Juicy Studio\* para tener una idea aproximada de la sencillez del texto de vuestro sitio web.

\* <http://juicystudio.com/services/readability.php>

De todos modos, hay algunos aspectos que se pueden comprobar de una manera muy objetiva, como por ejemplo si el contenido tiene metadatos de idioma que permitan (por ejemplo) que los lectores de pantalla y los navegadores de voz lean el contenido con la pronunciación correcta. Con HTML podéis utilizar un inspector DOM para comprobar la presencia de un atributo `lang` para el documento y para cada uno de los cambios de idioma.

Estad pendientes de las incoherencias de los sitios web, tanto respecto a las incoherencias internas como a la predictibilidad a partir de las convenciones habituales de la web. Los usuarios de ampliadores de pantalla que sólo pueden ver una parte de la página en cada momento se fían mucho de esta coherencia para saber dónde mirar para encontrar un contenido o una función concreta.

## Robustez

La comprobación de si el contenido es sólido implica comprobar que las tecnologías se utilizan correctamente. A un nivel muy básico, podéis pasar el etiquetado y el código por analizadores como:

- Validador HTML de WDG con las advertencias activadas  
<http://htmlhelp.com/tools/validator/>
- Validador de CSS del W3C  
<http://jigsaw.w3.org/css-validator/>
- Analizador de JavaScript JSLint  
<http://www.jshint.com/>

A continuación, podéis revisar el código con detenimiento para comprobar que las funciones se utilizan correctamente. Por ejemplo, podéis comprobar que se utilicen los controles nativos de HTML en lugar de emular controles con elementos sin significado y JavaScript, y que el JavaScript utilice la detección de funciones en lugar de la detección de navegador\* siempre que sea posible.

\* [http://www.jibbering.com/faq/faq\\_notes/not\\_browser\\_detect.html](http://www.jibbering.com/faq/faq_notes/not_browser_detect.html)

Entonces podéis hacer pruebas en múltiples agentes de usuario y tecnologías de asistencia y comprobar que el sitio sea perceptible, operativo y comprensible con cualquier combinación de CSS, JavaScript y conectores activados o desactivados.

El problema más habitual es probablemente el JavaScript obstructivo, como por ejemplo las anclas y los botones que se encuentran en el etiquetado sin *scripts* de la página pero que dependen del JavaScript para poder hacer realmente algo. Pero hay otros problemas más sutiles que surgen de la vinculación excesivamente estrecha del JavaScript con otras capas de la pila de tecnología. Por ejemplo, el JavaScript puede aplicar `display: none;` de CSS para ocultar el contenido, pero ¿qué sucede cuando no se aplica el CSS del editor?

Otro ejemplo son los controles multimedia. A menudo, cuando se incluye el contenido de un conector, la interfaz de usuario propia del conector se desactiva y en su lugar el conector pasa a estar controlado por *widgets* HTML con *scripts*. Cuando el contenido del conector sólo se añade mediante JavaScript después de la detección del conector basado en JavaScript, no hay ningún problema. Pero en ocasiones el contenido del conector se incluye en la página antes de cargarse los *scripts*. En estos casos, vale la pena comprobar no sólo que se dispone de una alternativa por si no hay disponible un conector adecuado, sino también que la interfaz de usuario propia del conector no esté desactivada a no ser que el JavaScript esté disponible. Si no se da el primer caso, entonces los usuarios verán el contenido alternativo; si no se da el segundo caso, entonces los usuarios verán el conector pero no lo podrán controlar.

## 2.5. Pruebas de usuario

Por más inspección y simulación que haya por parte de los desarrolladores, no hay nada que pueda superar el choque real entre un usuario y un sitio web para saber si está bien hecho. Teniendo en cuenta las dificultades para entender todas las sutiles interacciones entre el contenido de la web y las tecnologías de asistencia y las dificultades para acercarse a la experiencia de los usuarios con discapacidades, esto se aplica sobre todo a los usuarios con discapacidades. A ser posible, deberíais probar vuestros sitios web con usuarios reales con discapacidades. Esto se puede hacer a gran escala y con un presupuesto muy alto, pero no se deben subestimar los beneficios de hacer pruebas aunque sea a una escala muy pequeña.

### 2.5.1. Encontrar verificadores

Podéis encontrar verificadores de la misma manera que podéis encontrar candidatos para las pruebas de usabilidad en general (por ejemplo, por medio de anuncios y de agencias). Las organizaciones de discapacidades locales también os podrían recomendar los foros apropiados para encontrar sujetos para las pruebas.

Las pruebas no dejan de ser un trabajo y, por lo tanto, se deberían remunerar como tal. La tarifa habitual para las pruebas de usuario está en torno a los 70 dólares la hora.

Aun así, también es posible que podáis encontrar gente que realice pruebas de proyectos más pequeños de manera gratuita. Es posible que haya personas con discapacidades entre vuestros amigos, familiares y compañeros de trabajo. Además, hay grupos de discusión en línea centrados en las cuestiones de accesibilidad del software, como por ejemplo:

- Lista de discusión de accesibilidad WebAIM.  
<http://www.webaim.org/discussion/>
- Lista de correo del grupo de interés de la iniciativa de accesibilidad de la web: un foro para la discusión de cuestiones relacionadas con la accesibilidad de la web.  
<http://www.w3.org/WAI/IG/Overview.html>
- [jfw@freelists.org](mailto:jfw@freelists.org): lista de correo para usuarios del lector de pantalla JAWS.  
<http://www.freelists.org/archives/jfw/>
- GW-Info: lista de correo para usuarios del lector de pantalla GW Micro Window-Eyes.  
[http://www.gwmicro.com/Support/Email\\_Lists/](http://www.gwmicro.com/Support/Email_Lists/)
- [macvoiceover@freelists.org](mailto:macvoiceover@freelists.org): usuarios de VoiceOver de Apple.  
<http://www.freelists.org/list/macvoiceover>



- Blinux-list: lista sobre el uso de Linux por parte de personas ciegas y con discapacidades visuales.  
<https://www.redhat.com/mailman/listinfo/blinux-list>
- Usuarios de GNOME Orca.  
<http://mail.gnome.org/mailman/listinfo/orca-list>
- Grupo de Yahoo! deaf-uk-technology: discusión de tecnologías relacionadas con los sordos.  
<http://groups.yahoo.com/group/deaf-uk-technology/>

Estos grupos aceptan normalmente preguntas de desarrolladores web sobre la accesibilidad de sus sitios o sobre técnicas concretas.

### 2.5.2. Consideraciones prácticas

Recordad que el entorno de pruebas en sí también debe ser accesible. Por ejemplo, si preparáis materiales de prueba escritos, deberéis estar preparados para ofrecerlos en formatos alternativos. La logística necesaria para reproducir exactamente el entorno de navegación del usuario en vuestras instalaciones de pruebas habituales es muy complicada, por lo que puede ser más práctico hacer las pruebas en casa del usuario. Si no es posible, también se puede pensar en la posibilidad de hacer pruebas remotas.

Una consideración concreta que probablemente es aún más importante para los usuarios con discapacidades que para otros usuarios es la tecnología con la que están familiarizados. Las tecnologías de asistencia pueden añadir muchas capas de complejidad a su experiencia informática, lo cual crea una gran división entre los usuarios noveles y los experimentados; además, también podemos encontrar usuarios muy expertos con su configuración propia pero que se desorientan totalmente con las tecnologías que no conocen. (¡Sólo debéis pensar en las dificultades con las que se encuentran los usuarios sin discapacidades acostumbrados a los ordenadores por cambiar entre un Mac y un PC!).

Si encontráis a un usuario muy experimentado con el lector de pantalla Window-Eyes, ponedlo ante una máquina que desconozca con el lector de pantalla JAWS instalado y pedidle que pruebe un sitio web; le será muy difícil distinguir sus problemas con el JAWS de sus problemas con el sitio web. Vistas las diferencias tan importantes entre versiones y cómo los usuarios suelen personalizar su configuración, incluso les puede ser difícil aunque les deis el Window-Eyes. Por ello, si no es que queréis probar específicamente la accesibilidad de vuestro sitio web en entornos no familiares (por ejemplo en bibliotecas o en ordenadores de amigos), lo mejor es permitir que los usuarios hagan las pruebas con su configuración propia o con algo lo más similar posible a ésta.

De la misma manera, a no ser que queráis poner a prueba concretamente a usuarios muy inexpertos o muy expertos, deberíais seleccionar a unos usuarios con aproximadamente un año de experiencia con el uso de su configuración actual para acceder a la web. Es muy importante aprender tanto las tecnologías de asistencia como las convenciones de la web en sí. Con los usuarios inexpertos no sabréis si los problemas son culpa de vuestro sitio web o si son intrínsecos del proceso de aprendizaje, y los usuarios expertos pueden tener recursos al alcance que otros no tienen.

### 2.5.3. Elegir las tareas

El simple hecho de observar a un usuario explorando un sitio web puede llegar a ser muy instructivo. Igual que con cualquier otro tipo de pruebas de usuario:

- Intentad encargar algunas tareas concretas a los usuarios.
- Preguntadles qué piensan y escuchad bien lo que os digan.
- Prestad atención a aquello que hagan, porque puede ser diferente de aquello que dicen: las preferencias que puedan expresar son una mala guía para el rendimiento.

Cuando diseñéis un sitio web, os deberéis concentrar en las transacciones que quieren hacer los usuarios más que en los controles concretos que deban utilizar. De la misma manera, cuando probéis la accesibilidad, las tareas que encomendéis deberían reflejar (por lo menos inicialmente) los objetivos de un visitante que utilice el sitio y no se deberían concentrar tanto en sus interacciones con controles concretos. Estas transacciones serán normalmente similares para la gente con y sin discapacidades.

Por ejemplo, si estáis comprobando la accesibilidad de un sitio en el que se comparten vídeos, no empecéis preguntándoles si pueden utilizar controles concretos (“Eso es el volumen. ¿Lo podéis ajustar?”). Lo que deberíais hacer es plantearles escenarios y pedirles que realicen tareas clave para los usuarios. Por ejemplo:

- Echar una ojeada a una serie de vídeos y elegir uno para reproducirlo.
- Buscar un vídeo.
- Cargar un vídeo.
- Hacer una pausa en un vídeo, reproducirlo, quitar el sonido, poner el sonido, rebobinarlo y volver a reproducirlo.
- Calificar un vídeo.
- Compartir un vídeo con un amigo.

De esta manera, es muy probable que descubráis muchos problemas que no habíais previsto. Por ejemplo, un usuario de un lector de pantalla puede no ser capaz de encontrar el cuadro de búsqueda o los controles del vídeo. Pero también puede ser que los usuarios utilicen unas estrategias de navegación para moverse por el sitio web que ni siquiera conocíais.

#### 2.5.4. Interpretar los resultados

En un mundo ideal, podríamos probar todas las combinaciones posibles y saber la opinión de todo el mundo. Pero en la realidad, el tiempo y el dinero pondrán límites a las pruebas de usuario. Una vez asumido esto, las opiniones pueden ser al mismo tiempo una ventaja y un problema. Aunque nos pueden hacer ver muchas cosas, existe el peligro de dar demasiada importancia a la opinión de una persona que quizá no es representativa de la mayoría del público objetivo. Por ejemplo, algunos usuarios de lectores de pantalla suelen buscar una experiencia adaptada para usuarios ciegos, mientras que otros lo que quieren es saberlo todo sobre el sitio que sus amigos videntes ven.

Aquí es donde entran en juego realmente los estándares de accesibilidad como WCAG. Si seguís estas directrices, tendréis más posibilidades de disponer de una base de accesibilidad incluso para usuarios con los que no podéis hacer pruebas.

Cuando observáis un problema, debéis analizar sus causas. Por ejemplo, vuestro sitio web para compartir vídeos incluye una página que muestra los vídeos más populares en una tabla de datos; las columnas de esta tabla incluyen un fotograma del vídeo, un título, la fecha en la que se cargó, la fecha en la que se reprodujo por última vez y su índice de audiencia general, y todos los vídeos aparecen organizados por filas según su categoría. En las pruebas de usuario, un lector de pantalla tiene problemas para utilizar esta tabla de datos. Esto puede ser un reflejo de:

- Un problema con el código del sitio. Por ejemplo, puede ser que los desarrolladores hayan creado una tabla de datos a partir de elementos `div` sin significado en lugar de utilizar el etiquetado adecuado para una tabla de datos. En este caso, lo que habría que hacer es volver a escribir el código de la tabla.
- Inexperiencia por parte del usuario. Por ejemplo, un usuario de JAWS puede no estar familiarizado con las funciones de JAWS para navegar por tablas de datos y leer los datos. En este caso, lo que se podría hacer es ofrecer documentación adicional o consejos para los usuarios no tan expertos. Es posible que los usuarios expertos no sean los sujetos ideales para las pruebas, pero sí que son unos grandes asesores en cuestiones como ésta.
- Un problema con el agente de usuario. Por ejemplo, Safari expone la tabla de datos en el modelo de accesibilidad de Apple como una serie de cuadros de maquetación y no como un conjunto de relaciones de datos. En este caso, se podría informar de este problema al distribuidor o a los desarrolladores del agente de usuario, o intentar encontrar alguna técnica que funcionara con este agente de usuario, o quizá comentar esta limitación en la documentación y sugerir agentes de usuario alternativos que sí que funcionen con vuestro sitio web.

- Un problema con el lector de pantalla. Por ejemplo, los desarrolladores pueden haber acortado las cabeceras de tabla largas con el atributo `abbr`, pero el lector de pantalla no tiene ninguna interfaz de usuario para leer la versión abreviada. En este caso, se podría informar de este problema al distribuidor o a los desarrolladores del lector de pantalla, o tratar de encontrar alguna técnica que funcionara con este lector de pantalla, o quizá comentar esta limitación en la documentación y sugerir una herramienta o una estrategia de navegación alternativa que sí que funcione.

## 2.6. Comunicar los resultados de las pruebas de accesibilidad

A la hora de comunicar los resultados de la evaluación de accesibilidad, debéis documentar de manera precisa lo que se ha evaluado. Si habéis probado la conformidad con algún estándar en concreto, debéis puntualizar claramente dónde se da esta conformidad y dónde no. Siempre que surja un problema, comprobad que lo explicáis en términos sencillos y que explicáis claramente cómo puede afectar a los usuarios. Describid la manera de reproducir el problema y tratad de encontrar una solución. Sugerid técnicas prácticas para conseguir la conformidad o para mejorar la accesibilidad.

Por ejemplo, podéis informar de un problema con el sitio web para compartir vídeos de la siguiente manera:

- **Problema:** el menú desplegable no se puede abrir sin un ratón para situarse sobre las opciones del menú, y la selección del teclado desaparece de la pantalla cuando se pulsa la tecla del tabulador para desplazarse por el menú.
- **Cómo reproducirlo:** abrid la página en vuestro navegador e intentad llegar hasta un subelemento del menú utilizando sólo el teclado.
- **Explicación:** la navegación por la web no debería depender del dispositivo, de manera que los usuarios que utilizaran dispositivos diferentes de un ratón (como los usuarios ciegos o con discapacidades motrices) pudieran acceder al contenido y a las funciones. Actualmente, estos usuarios no pueden acceder a las opciones de los submenús y los usuarios videntes que utilizan el teclado pueden quedar desconcertados cuando desaparece el indicador de selección.
- **Implicaciones para la conformidad:** la operatividad del teclado es una exigencia para la conformidad con el nivel “A” de WCAG 1.0 y WCAG 2.0 (podéis ver la directriz 9 de WCAG 1.0 y la directriz 2.1 de WCAG 2.0).
- **Soluciones propuestas:** cuando no esté disponible JavaScript, utilizad una simple lista de enlaces a subpáginas para cada sublista de navegación. En las subpáginas, presentad la navegación principal seguida de la sublista. Cuando JavaScript esté disponible, eliminad la sublista del DOM y añadid sublistas para cada uno de los elementos de menú a la acción `click`; estos elementos se pueden activar con teclados, ratones, el reconocimiento de la voz y pantallas táctiles.

## Resumen

No todas las páginas pasarán por una evaluación de accesibilidad por parte de expertos y de todo un grupo de sujetos de pruebas pagados. Pero cualquier desarrollador web puede aprender los principios de la accesibilidad, intentar aplicarlos en su código y enviar los resultados de su trabajo a listas de correo

de usuarios para descubrir otros problemas, y de esta manera adquirir nuevos conocimientos que podrá aplicar en desarrollos futuros.

### **Preguntas de repaso**

1. Intentad navegar por un sitio complejo sin utilizar el ratón. ¿Qué dificultades tenéis? ¿Cómo os podrían ayudar a los desarrolladores del sitio?
2. Desactivad el CSS durante todo un día y haced lo que haríais normalmente. ¿Qué problemas tenéis?
3. Desactivad el JavaScript durante todo un día y haced lo que haríais normalmente. ¿Qué problemas tenéis?
4. Elegid uno de vuestros sitios preferidos, diseñad algunos personajes para este sitio y evaluad su conformidad con WCAG 1.0 y la accesibilidad en general como verificador experto. Diseñad un plan de pruebas de usuario para un sitio; incluid los requisitos para la selección de los usuarios y las tareas que se someterán a pruebas. Redactad un informe sobre cómo se podría mejorar la accesibilidad.

